

IBM z Systems Development and Test Environment

Version 11 Release 0 Modification 2

Contents

Chapter 1. IBM z Systems Development and Test Environment. 1

Chapter 2. Product overview 3

What is new?	5
Learning resources for z Systems Development and Test Environment.	6
IBM and accessibility	6
Notices	6
Trademarks.	8
Terms and conditions for product documentation	8

Chapter 3. IBM z Systems Development and Test Environment Tools 11

Introduction	11
Terminology	11
Capabilities	11
Roles	12
Limitations	13
Installing	14
Prerequisites	14
Configuring the Tools server environment	16
Installing IBM z Systems Development and Test Environment Tools	17
Installing IBM z Systems Development and Test Environment Tools z/OS files	19
Starting and stopping IBM z Systems Development and Test Environment Tools server	19
Updating IBM z Systems Development and Test Environment Tools	20
Uninstalling IBM z Systems Development and Test Environment Tools	21
Configuring	21
Configuring Users	22
Configuring Target environments	22
Configuring the FTP Storage.	23
Configuring the IBM Application Discovery Server	23
Configuring the ADCD	24
Configuring the emulator licenses	25
Operating	25
Creating application images	25
Deploying application images	29
Managing	31
Monitoring the creation and deployment	31
Scheduling	32
Troubleshooting	32
Setting up a Windows machine as an FTP server	32
Troubleshooting the configuration of the Application Discovery server	33
Troubleshooting the application components creation from IBM Application Discovery projects	33
Scenarios	34
Provisioning CICS subsystem from a source z System	34

Chapter 4. Installing z System Development and Test Environment . . . 37

z Systems Development and Test Environment prerequisites	37
Hardware provisioning	37
Hardware requirements	39
Software requirements.	41
Configuring the base Linux system	42
Migrating from previous version of z Systems Development and Test Environment	45
Uninstalling a previous version of z Systems Development and Test Environment	46
Installing z Systems Development and Test Environment interactively	48
Installing z Systems Development and Test Environment Personal Edition	48
Installing z Systems Development and Test Environment License Manager	49
Installing IBM z Systems Development and Test Environment silently	50
Installing z Systems Development and Test Environment Personal Edition	50
Installing z Systems Development and Test Environment License Manager	51
Updating IBM z Systems Development and Test Environment	51
Updating z Systems Development and Test Environment interactively	52
Updating z Systems Development and Test Environment silently	52
Uninstalling IBM z Systems Development and Test Environment	53
Uninstalling z Systems Development and Test Environment interactively	53
Uninstalling z Systems Development and Test Environment silently	53
After you install z Systems Development and Test Environment	54
Updating /etc/sysctl.conf	54
Updating \$HOME/.bashrc	54
Checking the installation	55
Starting the license manager.	55
Installing an operating system on z Systems Development and Test Environment	55
Installing z/OS volume images.	55
Installing z/VM	59

Chapter 5. Enabling product operation 61

Product enablement checklist	61
Software-based license.	61
USB hardware device	62
Software-based licensing	63
Software-based licensing terminology.	64
Overview of the activation process	65
The license manager	68

Activating a license manager	71
Activating a license manager with Rational Tokens	73
Returning a software-based license key less than 31 days before expiration	74
Returning a software-based license key more than 30 days before expiration	75
Activating and configuring a license manager client	77
The USB hardware device	78
USB hardware device licensing terminology	79
State of the hardware device.	81
Capacity of the hardware device	81
Serial number of the USB hardware device	82
Expiration dates	82
Capability configuration of the USB hardware device	85
License keys and the Application Developer Controlled Distribution (ADCD)	85
Parallel Sysplex Activations	85
Rational Token Activations and the USB hardware device.	86
Planning for USB hardware devices	87
Obtaining an update file	87
Activating the USB hardware device	93
Quick setup instructions for using and migrating the product license server	97
Next steps	100

Chapter 6. Configuring an instance of z Systems Development and Test Environment. 103

Defining the device map	103
Sample program to create device map	105

Chapter 7. Starting and stopping z Systems Development and Test Environment. 107

Starting the z Systems Development and Test Environment	107
IPLing z Systems Development and Test Environment from a remote emulated terminal for the system console	109
Stopping z Systems Development and Test Environment	110

Chapter 8. Customizing a z/OS distribution 111

Customizing a migrated z/OS system	111
A brief introduction to z/OS system configuration	111
Configuration overview	113
Logging on to TSO	113
Creating new zFS files for /tmp and /u	114
Setting up TCP/IP.	117
Optional tasks	122
Customizing the May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment	124

Defining the device map for z/OS 2.2 ADCD	125
Starting and stopping the z/OS 2.2 ADCD	127
Configuring a working z/OS system	129
z/OS 2.2 ADCD for z Systems Development and Test Environment	143
Hints and tips to manage the z/OS 2.2 ADCD	166

Chapter 9. Deploying z Systems Development and Test Environment for Parallel Sysplex and the coupling facility 183

Installing z Systems Development and Test Environment with Parallel Sysplex and z/OS guests	183
Downloading a z/VM distribution	184
Enabling the coupling facility	184
Enabling the coupling facility in the device map	184
Starting IBM z Systems Development and Test Environment and IPLing z/VM	185
IPLing the z/OS guests	186
Starting a TSO session	186
Configuring a working z/OS 2.2 ADCD	187
Shutting down all systems and stopping IBM z Systems Development and Test Environment.	187
z/VM 6.3 for z Systems Development and Test Environment notes	187
Products contained in this release	188

Chapter 10. Troubleshooting and support 191

Finding known problems	191
IBM Software Support	191
Obtaining the latest editions of Redbooks	191
Troubleshooting tips	192
Installing z Systems Development and Test Environment	192
USB Hardware device activation and licensing	194
Troubleshooting installing, operating, and uninstalling the License Manager.	198
Installing z/OS volume images	198
Defining the z Systems Development and Test Environment machine characteristics (DEVMAP)	199
Starting and stopping z Systems Development and Test Environment	199
z/OS networking issues.	200
Performance.	201
Connecting with Rational License Key Servers	202
Troubleshooting license manager and USB hardware device activation	203

Chapter 11. Reference. 205

Setting up Rational Tokens	205
Installing, configuring, and administering a Rational License Key Server	205
Returning any existing license key for the USB hardware device or license manager.	205
Obtaining and activating a new license key for Rational Token licensing.	206

Updating the device map to point to a Rational License Key Server	206
Troubleshooting connections with Rational License Key Servers	207
zPDT license servers	208
Concepts	210
Client installation and configuration	214
Server installation and configuration	216
Notes	218
Scenarios	220
Renewing licenses	225
Searching servers	226
Allocating multiple licenses	227

Switch from Product License Server to License Manager	227
Accessibility features	228
z Systems Development and Test Environment Keyboard shortcuts for the help system in the product	229




Chapter 12. z Systems Development and Test Environment Glossary . . . 231

Index 237

Chapter 1. IBM z Systems Development and Test Environment



This page provides an entry point to product information about IBM z Systems™ Development and Test Environment. Use the links in the navigation tree on the left to find documentation for specific versions and editions of this product family. On the right, you can find links to various helpful resources. That list is updated frequently. Below, you can find entry points to sites that help you learn more about this product and other IBM® products. It provides links to sources for support and troubleshooting information.

Learn more




-  [IBM z Systems Development and Test Environment product page](#)
-  [Enterprise DevOps for z Systems cloud-based trial](#)
-  [z Systems Development and Test Environment product page on IBM developerWorks](#)

Product overview





IBM z Systems Development and Test Environment (previously named Rational® Development & Test Environment for System z) runs an IBM z Systems z/OS® distribution on a personal computer or workstation Linux environment. It creates an environment for mainframe application demonstration, development, testing, and employee education without the need for z Systems mainframe hardware. It enables z/OS, middleware, and other z/OS software to run on Intel and compatible computers. It emulates z Systems architecture with virtual I/O and devices.

-  [Liberate mainframe development teams with IBM z Systems Development and Test Environment](#)
-  [z Systems Development and Test Environment videos on IBM developerWorks](#)



Connect with experts

-  [developerWorks forum for IBM z Systems Development and Test Environment](#)
-  [IBM z Systems Development and Testing Hub](#)
-  [z Systems Development and Test Environment blogs on IBM developerWorks](#)

Find documentation

-  [IBM z Systems Development and Test Environment Library](#)
-  [Find PDF documentation](#)
-  [zPDT Guide and Reference](#)
-  [zPDT 2016 Sysplex Extensions](#)

Find support

-  [IBM Support](#)
-  [IBM Software Support home page](#)

Chapter 2. Product overview

IBM z Systems Development and Test Environment (previously named Rational® Development & Test Environment for System z) runs an IBM z Systems z/OS distribution on a personal computer or workstation Linux environment. It creates an environment for mainframe application demonstration, development, testing, and employee education without the need for z Systems mainframe hardware. It enables z/OS, middleware, and other z/OS software to run on Intel and compatible computers. It emulates z Systems architecture with virtual I/O and devices.

z Systems Development and Test Environment provides these features:

- An application development and testing environment that can improve development infrastructure availability and flexibility.
- Mixed workload support that can help reduce development costs.
- A high fidelity development and test environment that can streamline the delivery of production-ready applications.
- Current levels of IBM z/OS software, giving access to new runtime capabilities for development and testing purposes.
- An environment for employee education on z Systems that is approachable and portable.

Important: z Systems Development and Test Environment cannot be used for production workloads of any kind, nor robust development workloads, including without limitation, production module builds, preproduction testing, stress testing, or performance testing.

z Systems Development and Test Environment Tools

IBM z Systems Development and Test Environment Tools is a simple utility which enables you to create and monitor instances of z Systems installs, or import existing packages and deploy them to test systems. This allows you to develop and test outside the regular production systems.

IBM z Systems Development and Test Environment Tools includes the following features:

- Creating application images from various sources.
- Provisioning application images for developers and testers in a self-service automated way.
- Monitoring the status and availability of all created assets and target systems.

To Learn more about z Systems Development and Test Environment Tools, see IBM z Systems Development and Test Environment Tools

To enable IBM z Systems Development and Test Environment Tools, you need to install IBM z Systems Development and Test Environment License Manager with valid licenses. For more information, see Chapter 4, “Installing z System Development and Test Environment,” on page 37.

z Systems Development and Test Environment

IBM z Systems Development and Test Environment is a platform for mainframe application demonstration, development, testing and education. It enables IBM z/OS® middleware and other z/OS software to run on Intel-compatible platforms without z Systems mainframe hardware. The personal edition provides two kinds of licensing mechanism.

- **Hardware (USB) Based License**

This licensing method requires a USB hardware device that contains a license key that controls the licensee's access to all or portions of the program. To use USB based license, you can use the following two ways:

- Making a machine as server and connecting clients (Instances) to server to get licenses.
- Installing the z Systems Development and Test Environment (server/client) on same machine.

To learn more about USB based license, see “The USB hardware device” on page 78.

- **Software Based License**

When you use software-based licensing, a license manager (License Server) provides a server for centralized management of license keys for one or more instances of z Systems Development and Test Environment.

The z Systems Development and Test Environment license manager is installed on a PC, and the authenticated license key files are installed on the license manager (License Server). The licensee is not authorized to use z Systems Development and Test Environment or any of its components except when that license manager is activated with a license key file and is accessible by the program, as it provides proof of license entitlement.

The license manager and an instance that runs z Systems Development and Test Environment cannot run on the same platform, but it does need network connectivity to the license manager. You can also enable license using Rational Tokens.

To learn more about Software Based License including Rational tokens, see “Software-based licensing” on page 63.

z/OS Application Developer Controlled Distribution

One or more z/OS distributions are supplied with z Systems Development and Test Environment. Each is a unique distribution that is created specifically for sites that are entitled to z Systems Development and Test Environment, based on the Application Developers Controlled Distribution (ADCD). They are referred to in this IBM Knowledge Center as the z/OS ADCD.

A z/OS ADCD for z Systems Development and Test Environment contains many of the products and subsystems you expect in a z/OS system. Most of these products and subsystems run without any customization, while some need some customization or initialization to make them work. If you migrate your own z/OS distribution, you lessen the need for customizations to your own individual products and subsystems. However, customizations to networking, such as mapping your own network devices to the devices in your Linux environment, might be required.

Important: Any z/OS ADCD for z Systems Development and Test Environment, and any related software, are supplied as a convenience. IBM does not support the

operating system distributions that are made available with z Systems Development and Test Environment. You can apply fixes to any z/OS distribution through normal SMP/E facilities.

To learn more about ADCD, see “Installing an operating system on z Systems Development and Test Environment” on page 55

This IBM Knowledge Center contains instructions for installing and configuring z Systems Development and Test Environment, and z Systems Development and Test Environment Tools. It gives guidance for installing and customizing a z/OS distribution, whether it is a z/OS ADCD for z Systems Development and Test Environment, a migrated customer z/OS system, or a z/OS distribution from a previous release of Rational® Development and Test Environment for z Systems. Therefore, people who read this IBM Knowledge Center need basic z Systems and z/OS knowledge. Readers must understand the concept of device addresses and must be able to use the Interactive System Productivity Facility (ISPF) and SDSF to edit files, submit jobs, and review job output. After you complete the instructions in this IBM Knowledge Center, you might need to do additional customization of z/OS to meet your individual needs.

You also need a basic knowledge of the Linux operating system. Readers must be able to log in and out of their Linux system, use a Linux command line with the bash shell, navigate the Linux file system, edit files, set file permissions, mount and unmount file systems, and switch between regular user and superuser (root) authority. Knowledge of package installation methods for your Linux distribution is required if you need to install missing packages such as x3270, Perl, or other optional tools that might not be installed at your site.

What is new?

Learn what's new in this release.

The following new features and changes were added for IBM z Systems Development and Test Environment Version 11.0.2:

IBM z Systems Development and Test Environment Tools

- **Support for extraction of datasets**

IBM z Systems Development and Test Environment Tools V11.0.2 supports extracting z/OS datasets from an IBM z™ System. For more information, see “Creating application components from IBM z System datasets” on page 26.

- **IBM Application Discovery integration**

IBM Application Discovery is integrated with IBM z Systems™ Development and Test Environment Tools V11.0.2. With the integration, you can deploy the datasets that are identified through an IBM Application Discovery project to the target machine. For more information, see “Creating application components from IBM Application Discovery projects” on page 27.

- **Support for certificate-based login**

SSH key-based authentication for the root account can be used when you deploy an application image to a target environment. For more information, see “Deploying application images” on page 29.

- **Option to auto IPL system**

As a part of deployment, you can choose to issue the z/OS initial program load (IPL) command after the emulator is started. For more information, see “Deploying application images” on page 29.

IBM z Systems Development and Test Environment

- **Support withdraw**

Rational Test Control Panel 9.1.0 e-assembly (CNJ1TML) is not available with z Systems Development and Test Environment e-assembly from version 11.0.2.

Learning resources for z Systems Development and Test Environment

Access web-based resources for learning more about IBM z Systems Development and Test Environment.

- IBM z Systems Development and Test Environment product page
- Enterprise DevOps for z Systems cloud-based trial
- developerWorks®: Rational z Systems Development and Testing Hub
- developerWorks forum for IBM z Systems Development and Test Environment
- zPDT® Guide and Reference
- zPDT 2016 Sysplex Extensions
- Education Assistant

IBM and accessibility

IBM strives to provide products with usable access for everyone, regardless of age or ability.

For Reference information about the accessibility features available in this product, see “Accessibility features” on page 228 and the topics under it. You can review details about navigating the user interface by using the keyboard, listening to the user interface controls with a screen reader, or adjusting the colors and typefaces that are used in the interface. Information about navigating the help system by using the keyboard is in the topic “Keyboard shortcuts for the help system in the product” on page 229.

For more information about the commitment to accessibility that IBM has, see the IBM Human Ability and Accessibility Center.

Note: Recent accessibility news from IBM

Notices

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

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

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-17

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

Intellectual Property Licensing Legal and Intellectual Property Law IBM Japan Ltd. 19-21, Nihonbas

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

This information might 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 might make improvements or changes in the product(s) or the program(s) described in this publication at any time without notice.

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

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

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

IBM Director of Licensing IBM Corporation North Castle Drive, MD-NC119 Armonk, NY 10504-17

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

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

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

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of

performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

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

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

COPYRIGHT LICENSE:

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

Each copy or any portion of these sample programs or any derivative work must include a co

Trademarks

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

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

Chapter 3. IBM z Systems Development and Test Environment Tools

This document describes the usage of IBM z Systems Development and Test Environment Tools, including capabilities, installation, and configuration.

Introduction

IBM z Systems Development and Test Environment Tools is a simple Web utility that enables you to create and monitor instances of z System installs, or import existing packages and deploy them to test systems. This allows you to develop and test outside the regular production systems.

IBM z Systems Development and Test Environment Tools includes the following features:

- Creating application images from various sources.
- Deploying and automatically provisioning application images to z Systems Development and Test Environment instances.
- Monitoring the status and availability of all created assets and target systems.

Terminology

Understand the terms that are used to describe IBM z Systems Development and Test Environment Tools.

Application Component

- Collection of volumes from an IBM z System, intended to be reusable in a mix and match fashion with other components from the same IBM z System.

Application Image

- Collection of application components that originate from the same IBM z System, a z System Development and Test Environment system, or an ADCD package. The application components can be packaged together as a single deliverable for a target system.

Target Environment

- One (or more) x86 Intel (or compatible) systems capable of running IBM z Systems Development and Test Environment that users can deploy an application image to.

Capabilities

Learn about the capabilities of IBM z Systems Development and Test Environment Tools.

IBM z Systems Development and Test Environment Tools includes the following capabilities:

Application image creation

IBM z Systems Development and Test Environment Tools allows you to create application images from various sources:

- Existing IBM z System, including:
 - IBM z System DATASETS
 - IBM z System VOLUMES
 - IBM Application Discovery PROJECTS
- Existing IBM z Systems Development and Test Environment instance.
- Application Developers Controlled Distributions (ADCD). The z/OS Software distribution is bundled with the product.

Self-service provision of application images

After application images are created, IBM z Systems Development and Test Environment Tools allows you to provision application images to Linux systems; z Systems Development and Test Environment instances will be created in an automated way.

Dashboard monitoring

IBM z Systems Development and Test Environment Tools allows you to monitor inventory and check status on all created assets and target environments. You can ensure that the creation and deployment processes are running properly, and also manage access permissions to the images you created.

Roles

Learn about the various roles that can be assigned when configuring users of IBM z Systems Development and Test Environment Tools.

For users who log in to IBM z Systems Development and Test Environment Tools, three types of roles are included:

Administrator

Users who are responsible for maintaining the system to run IBM z Systems Development and Test Environment Tools server and the software. Administrators can use any function of this product.

Builder

Users who are responsible for building the components and application images. Builders are knowledgeable about z/OS and IBM z Systems that might be used by the Tools. However, builders do not have permission to modify any configuration settings within the Tools.

Developer or Tester

Users who only use the Tools to deploy an already configured image to a new instance of IBM z Systems Development and Test Environment on a target environment.

Related information:

“Configuring Users” on page 22

Learn how to configure the Users page.

Limitations

Before you use IBM z Systems Development and Test Environment Tools, make sure that you are aware of the following known limitations.

Application Developer Controlled Distribution (ADCD)

There is only one ADCD license. You can uncompress only one ADCD volume(*.ZPD) at a time.

Creation support

IBM z Systems Development and Test Environment Tools only supports the creation of monoplex-emulated Z environments now.

On-going extracted volumes

When the creation of application images is in progress, if the WebSphere® Liberty server that hosts the Tools is shut down, the permissions to access IBM z System will be lost, which will cause a failure to extract the remaining volumes. Then, you need to rerun the creation process to reextract and build the application images.

IBM Application Discovery (AD) integration

Before you create application images from IBM Application Discovery projects, make sure that you are aware of the following known limitations:

- If the dataset is referred in the SYSIN card, IBM Application Discovery might not be able to identify the datasets, and cannot pick datasets from the JCL as shown in the following example:

```
//DELETE1 EXEC PGM=IDCAMS  
  
//SYSPRINT DD SYSOUT=**  
  
//SYSIN DD *  
  
DELETE IBMUSER.GENAPP.KSDSCUST  
  
IF MAXCC=8 -  
  
THEN SET MAXCC=0  
  
/*
```

- The dataset that is uncatalogued and identified by Application Discovery cannot be transferred.
- All versions of GDG that is identified by Application Discovery will be transferred.
- If you plan to use Application Discovery to provision the CICS environment, you need to put the CICS JCL in the JCL PDS where your project is built. Also, any dependency that is not identified as a part of CICS startup JCL will not be transferred to the target machine. You might have to transfer it separately. For more information, see “Provisioning CICS subsystem from a source z System” on page 34.
- All entities of VSAM file will be transferred. For example, if KSDS has an alternative index, it will be transferred with the primary cluster to the target machine, even though the alternative index is not identified in the Application Discovery project.

- All members of the PDS identified by Application Discovery project will be transferred. For example, if `ibmuser.sysin(xyz)` is referred in the JCL, the complete PDS (`ibmuser.sysin`) will be transferred to the target machine.
- Tapes are not supported.

Installing

Learn how to install IBM z Systems Development and Test Environment Tools.

To install IBM z Systems Development and Test Environment Tools, two parts are required:

1. The web User Interface
2. The z/OS program FEUVIMG

Web User Interface

The web User Interface is available as an RPM Package Manager (RPM) for installation on supported Red Hat Linux distributions, or as a Debian package (DEB) for installation on supported Ubuntu distributions.

z/OS program FEUVIMG

The z/OS program FEUVIMG is available as a System Modification Program/Extended(SMP/E) offering, and can be found in the `/opt/ibm/zDT/zSystem/` directory of the IBM z Systems Development and Test Environment Tools web User Interface install.

Note: For z Systems Development and Test Environment Tools beta users, you need to manually run the following command first:

```
rpm -e ibm-java-x86_64-sdk-8.0-4.7.x86_64
```

Then you can successfully install this version of z Systems Development and Test Environment Tools.

Prerequisites

Learn about the prerequisites for IBM z Systems Development and Test Environment Tools.

IBM z Systems Development and Test Environment Tools requirements

Learn about the hardware and software requirements for the web server that the Tools runs on.

To install and run IBM z Systems Development and Test Environment Tools, a 64-bit x86 Intel system is needed. The additional required resources are as follows:

Disk space

- 2GB of disk space is required.

Memory

- 1GB of RAM for Linux

Processor

- Minimal.

Operating systems and platforms

- Red Hat 6.x/7.x 64-bit or Ubuntu 16.04 64-bit Linux OS.

Ubuntu requirement

Libasound2 must be installed. To install libasound2, use the following command:

```
sudo apt-get update  
sudo apt-get install libasound2
```

Other requirements

- root permissions.

z/OS system requirements

Learn about z/OS system requirements.

If you want to extract volumes from z/OS systems, the following required and optional requirements are needed.

Required

The required requirements include:

- An SSH server must be running and accessible by the system to run the Tools.
- The FTP client must be able to connect to the Tools FTP server.
- Grant READ access to the SAF profile in class DASDVOL for the volumes that will be extracted to the user ID that will run the Tools.

Optional

The optional requirements include:

- Configure zEnterprise Data Compression (zEDC) if it is available. This hardware feature can offload the compression work from your CPs and reduce processing time.
- Grant READ access to the resource FPZ.ACCELERATOR.COMPRESSION in SAF class FACILITY to the user ID that will run the Tools.

FTP storage requirements

Learn about the FTP storage requirements.

To install and run IBM z Systems Development and Test Environment Tools, an FTP server to host the Tools artifacts, such as z system volumes, datasets, tooling metadata, must be set up.

Disk space

- Sufficient space is needed to hold numerous and potentially large files for extracted IBM z Systems volumes.
- 150GB of disk space is needed for ADCD z/OS V2.2 distribution.

Software requirements

- A running FTP server.

Target systems requirements

Learn about the target systems requirements.

To automatically deploy application images to any target environment, the following requirements are needed:

Basic requirements

For the basic hardware and software requirements, see “Hardware requirements” on page 39 and “Software requirements” on page 41.

Additional requirements

The additional requirements are as follows:

- The software repository must be available and accessible by the target environments.
 - A Red Hat software repository for 'yum' must be available and accessible by the target environments.
 - An Ubuntu software repository for 'apt-get' must be available and accessible by the target environments.
- An SSH server must be running on the target environments and accessible by the system to run the Tools.
- The following commands must reside on the target environments: `wget` and `ftp client`.
- The root permission is needed for the users who are responsible for deployment.
- An extra 100 M of disk space is needed for the folder `/root` in the target system, as the `loadparm.txt` that is generated for a script to modify z/OS parameters might cause space problem.

Licensing requirements

Learn about the licensing requirements for z Systems Development and Test Environment Tools.

To install and run z Systems Development and Test Environment Tools, the following licensing requirements are needed:

- At least one License Manager that runs software licensing must be installed to provide licenses to the target systems.
- If your license requires Rational Tokens, then a Rational License Key Server must be installed and run.

Configuring the Tools server environment

Learn how to configure IBM z Systems Development and Test Environment Tools server environment.

Before starting IBM z Systems Development and Test Environment Tools, you must configure the IBM WebSphere Application Server Liberty instance to utilize your company's LDAP system.

To configure the IBM WebSphere Application Server Liberty instance, follow these steps:

1. Navigate to the `/opt/ibm/zDT/Liberty/usr/servers/zDTServer` directory, which is the IBM WebSphere Application Server Liberty instance directory for the Tools.
2. Refer to *Configuring LDAP user registries in Liberty* to learn how to define a LDAP registry.

3. The server for the Tools has been configured to read the LDAP registry information from the file `ldap-conf.xml`, you can find the file in the server instance directory. If this file does not exist, then you need to create it, and define a LDAP registry by referring to Configuring LDAP user registries in Liberty.

Note: The server configuration already contains the appSecurity-2.0 and ldapRegistry-3.0 Liberty features.

Installing IBM z Systems Development and Test Environment Tools

Learn how to install IBM z Systems Development and Test Environment Tools.

`zDT_Install.x86_64` is a bash script that allows you to install, update or uninstall the following z Systems Development and Test Environment products:

- IBM z Systems Development and Test Environment Personal Edition
- IBM z Systems Development and Test Environment License Manager
- IBM z Systems Development and Test Environment Tools

If you have previously purchased the product, you can use your existing Software Licensing or activate the product with your USB hardware device and a remote license server.

For the download information about the installation package, see IBM z Systems Development and Test Environment e-assemblies to download from Passport Advantage. After downloading the package, you need to unzip the package by issuing the command `tar -xzvf filename.tgz`. The unzipped package includes the installer and `readme.txt`. To know more about the installation steps, see `Readme.txt` file.

After copying the installation file, `zDT_Install.x86_64`, to the target machine, you can run this bash script to install, update or uninstall z Systems Development and Test Environment products.

To install IBM z Systems Development and Test Environment Tools by using this script, you can select one of the following methods:

- Interactive installation
- Silent installation

Note: The installation procedure needs to be executed from a root User ID.

Installing IBM z Systems Development and Test Environment Tools interactively

Learn how to install IBM z Systems Development and Test Environment Tools interactively.

To install IBM z Systems Development and Test Environment Tools interactively, follow these steps:

1. Go to the directory where the installation program is present.
2. Execute the following command:

```
./zDT_Install.x86_64
```

Then the following menu is displayed.

The following products can be installed/updated/uninstalled. Please select one:

- 1) IBM z Systems Development and Test Environment Personal Edition
- 2) IBM z Systems Development and Test Environment License Manager
- 3) IBM z Systems Development and Test Environment Tools

==>

3. Type **3** to select IBM z Systems Development and Test Environment Tools. Then the following menu is displayed.

What do you want to do with IBM z Systems Development and Test Environment Tools?

- 1) Install
- 2) Update
- 3) Uninstall

==>

4. Type **1** to install the product.
5. Press ENTER, and read the license agreements carefully. Then at the end of license, enter 'Yes' to accept or 'No' to decline the terms.
If you accept the terms in the license agreements, type 'yes' and then ENTER
If you do not accept the terms in the license agreements, type 'no' and then ENTER
6. Wait for the installation to complete.
7. Run the following commands to see if the installation is successful.

```
rpm -qa | grep war
rpm -qa | grep ibm-java-x86_64-jre
rpm -qa | grep derby
rpm -qa | grep liberty
rpm -qa | grep zdtserverenv
rpm -qa | grep zsystem
rpm -qa | grep zdttoolsdb
```

You should get the following outputs:

```
war-1-11.0.0.x86_64
ibm-java-x86_64-jre-8.0-4.10.x86_64
derby-2-10.13.1.1.x86_64
liberty-2-17.0.0.1.x86_64
zdtserverenv-1-11.0.0.x86_64
zsystem-1-11.0.0.x86_64
zdttoolsdb-1-11.0.0.x86_64
```

Installing IBM z Systems Development and Test Environment Tools silently

Learn how to install IBM z Systems Development and Test Environment Tools silently.

To install z Systems Development and Test Environment Tools silently, follow these steps:

1. Go to the directory where the installation program is present.
2. Execute the following command:

```
./zDT_Install.x86_64 --install --zdttools
```

Note: If you install silently, you will not see the option to review the license. The installation process assumes that you have reviewed the license before installation. Read the license as in the installation media, or use the command line installation option.

3. Wait for the installation to complete.
4. Run the following commands to see if the installation is successful.

```
rpm -qa | grep war
rpm -qa | grep ibm-java-x86_64-jre
rpm -qa | grep derby
```



```

rpm -qa | grep liberty
rpm -qa | grep zdtserverenv
rpm -qa | grep zsystem
rpm -qa | grep zdttoolsdb

```

You should get the following outputs:

```

war-1-11.0.0.x86_64
ibm-java-x86_64-jre-8.0-4.10.x86_64
derby-2-10.13.1.1.x86_64
liberty-2-17.0.0.1.x86_64
zdtserverenv-1-11.0.0.x86_64
zsystem-1-11.0.0.x86_64
zdttoolsdb-1-11.0.0.x86_64

```

Installing IBM z Systems Development and Test Environment Tools z/OS files

Learn how to install IBM z Systems Development and Test Environment Tools z/OS files.

This section describes the installation of the mainframe portion of z Systems Development and Test Environment Tools on the system from which data is extracted for setting up z Systems Development and Test Environment instances.

To install the IBM z Systems Development and Test Environment Tools host files, follow these steps:

1. After `zDT_Install.x86_64` bash script is run on a Linux machine to install the web UI, the directory `/opt/ibm/zDT/zSystem` contains the IBM z System components and the program directory file, `HALMB02.pdf`.
2. For the installation instructions of IBM z System component, refer to the details in `HALMB00.pdf`, specifically in section 6.0 "Installation Instructions".

Note: The data set `hlq.IBM.HALMB02.F2` obtained in step 6.1.3 is a usable program library. If you want to install the Tools without SMP/E, complete the following steps:

1. Choose a value for `dsnprefix`, and rename `hlq.IBM.HALMB02.F2` to `dsnprefix.SFEUAUTH`.
2. Go to step 6.1.11 by skipping steps from 6.1.4 to 6.1.1.

Starting and stopping IBM z Systems Development and Test Environment Tools server

Learn to start and stop IBM z Systems Development and Test Environment Tools server.

To start the IBM z Systems Development and Test Environment Tools server, you must use the root User ID to issue the following command:

```
/opt/ibm/zDT/bin/startServer
```

This script is written to ensure that the server process runs under the **zdt** userid, which is created during the Tools installation.

To stop the server, issue the command

```
/opt/ibm/zDT/bin/stopServer
```

Updating IBM z Systems Development and Test Environment Tools

Learn how to update IBM z Systems Development and Test Environment Tools.

To update IBM z Systems Development and Test Environment Tools, you need to execute the update procedure from a root User ID. When you update z Systems Development and Test Environment Tools, the user data will not be changed. However, when you install z Systems Development and Test Environment Tools, all user data will be deleted.

Updating IBM z Systems Development and Test Environment Tools interactively

Learn how to update IBM z Systems Development and Test Environment Tools interactively.

To update IBM z Systems Development and Test Environment Tools interactively, follow these steps:

1. Go to the directory where the installation program is present.
2. Execute the following command:

```
./zDT_Install.x86_64
```

Then the following menu is displayed.

The following products can be installed/updated/uninstalled. Please select one:

- 1) IBM z Systems Development and Test Environment Personal Edition
- 2) IBM z Systems Development and Test Environment License Manager
- 3) IBM z Systems Development and Test Environment Tools

==>

3. Type **3** to select IBM z Systems Development and Test Environment Tools. Then the following menu is displayed.

What do you want to do with IBM z Systems Development and Test Environment Tools?

- 1) Install
- 2) Update
- 3) Uninstall

==>

4. Type **2** to update the product.
5. Press ENTER, and read the license agreements carefully. Then at the end of license, enter 'Yes' to accept or 'No' to decline the terms.
If you accept the terms in the license agreements, type 'yes' and then ENTER
If you do not accept the terms in the license agreements, type 'no' and then ENTER
6. Wait for the update to complete.

Updating IBM z Systems Development and Test Environment Tools silently

Learn how to update IBM z Systems Development and Test Environment Tools silently.

To update z Systems Development and Test Environment Tools silently, follow these steps:

1. Go to the directory where the installation program is present.
2. Execute the following command:

```
./zDT_Install.x86_64 --update --zdttools
```
3. Wait for the update to complete.

Uninstalling IBM z Systems Development and Test Environment Tools

Learn how to uninstall IBM z Systems Development and Test Environment Tools.

To uninstall IBM z Systems Development and Test Environment Tools, you need to execute the uninstallation procedure from a root User ID.

Uninstalling IBM z Systems Development and Test Environment Tools interactively

Learn how to uninstall IBM z Systems Development and Test Environment Tools interactively.

To uninstall IBM z Systems Development and Test Environment Tools interactively, follow these steps:

1. Go to the directory where the installation program is present.
2. Execute the following command:

```
./zDT_Install.x86_64
```

Then the following menu is displayed.

The following products can be installed/updated/uninstalled. Please select one:

- 1) IBM z Systems Development and Test Environment Personal Edition
- 2) IBM z Systems Development and Test Environment License Manager
- 3) IBM z Systems Development and Test Environment Tools

==>

3. Type 3 to select IBM z Systems Development and Test Environment Tools. Then the following menu is displayed.

What do you want to do with IBM z Systems Development and Test Environment Tools?

- 1) Install
- 2) Update
- 3) Uninstall

==>

4. Type 3 to uninstall the product.
5. Wait for the uninstallation to complete.

Uninstalling IBM z Systems Development and Test Environment Tools silently

Learn how to uninstall IBM z Systems Development and Test Environment Tools silently.

To uninstall z Systems Development and Test Environment Tools silently, follow these steps:

1. Go to the directory where the installation program is present.
2. Execute the following command:

```
./zDT_Install.x86_64 --uninstall --zdttools
```
3. Wait for the uninstallation to complete.

Configuring

Learn how to configure IBM z Systems Development and Test Environment Tools.

After the Tools server environment is configured, you need to start the Tools server and navigate the URL in a web browser, which is specified in the output of the startServer script. For more information about starting the Tools server, see “Starting and stopping IBM z Systems Development and Test Environment Tools server” on page 19.

Configuring Users

Learn how to configure the Users page.

The Users page requires Administrator privileges. To configure the Users page, you need to configure the administrators first, and then configure the users.

Administrator Configuration

To log in to the Tools, use the default userid **zdtadmin** that has a default password of **password**.

To add your LDAP account to the Tools, follow these steps:

1. Click Users on the QUICK START page, or click the left-top button to navigate to **Configure > Users page**.
2. Click Add User.
3. Add your LDAP user ID. Typically, the LDAP user ID is your company email address.
4. Click Administrator role, which has access to all aspects of the Tools.
5. Click Add User.
6. Sign out of The tool.
7. Sign in to the Tool with your LDAP user ID and password to ensure all configuration is proper.
8. Navigate to **Configure > Users** page, and remove the **zdtadmin** account.
9. Add any other LDAP users who will act as Administrators.

Users Configuration

Only the users who are listed on the Users page have access to the Tools.

To configure the users who can access to the Tools, follow these steps:

1. Click **Users** on the **QUICK START** page, or click the left-top button to navigate to the **Configure > Users** page.
2. Add the users who need to have access to the Tools, and then select the corresponding roles for the users.
 - For a user who will create application components or application images, select the Builder role.
 - For a user who will deploy an application image to a target environment, select the Developer/Tester role.

Configuring Target environments

Learn how to configure the target environments.

The target environment page is available for all users with different roles.

You can provide the list of environment that the application images will be deployed to.

To configure the target environment, follow these steps:

1. Click Target environments on the QUICK START page, or click the left-top button to navigate to **Configure > Target environments**.
2. Click Add System.

3. Enter the qualified hostname and SSH (Secure Socket Shell) port number of this environment.
4. Click Add System to save the system to the Tools repository.

Note:

1. When an application image is deployed to a target system, a prompt will be issued for the root credentials.
2. For more information about required configuration and software installed on the systems, see “IBM z Systems Development and Test Environment Tools requirements” on page 14.

Configuring the FTP Storage

Learn how to configure the FTP Storage.

The FTP Storage page also requires Administrator privileges.

The FTP Storage is a crucial function of the Tools. To transfer and store all of the contents on the FTP storage, you need to have adequate storage space. For more information about the requirements of the storage space, see “FTP storage requirements” on page 15.

To configure the FTP Storage, complete the following steps:

1. Click FTP storage on the QUICK START page, or click the left-top button to navigate to the **Configure > FTP storage** page.
2. Enter the fully qualified host name of the system that provides the FTP service, along with the port, directory, user ID and password.
3. Click the Test Connection button to ensure that the Tools system is able to communicate with the FTP service.
4. Click the Save button.

Note: The credentials for the FTP storage are the only credentials that the tools stores in its local database and are encrypted using AES 128 bit encryption.

Configuring the IBM Application Discovery Server

Learn how to configure the IBM Application Discovery Server.

The IBM Application Discovery Server page requires Administrator privileges.

To create the application image from IBM Application Discovery project, you need to setup the connection to IBM Application Discovery server:

HTTP/HTTPS

By default, the Application Discovery web service is available via the HTTPS connection. If your Application Discovery web service is available via the HTTP connection, drag the toggle switch to HTTP.

Host Name

The host name or IP of the system that runs IBM Application Discovery.

Port The port number of the IBM Application Discovery server. Typically, the port number is 8090.

User Name and Password

If your HTTP or HTTPS server requires basic authentication, you must

enter the password; otherwise, an "Invalid credential" error message will occur. If your HTTP or HTTPS server does not require basic authentication, leave it blank.

Note: You must ensure that all above terms are configured correctly; otherwise, error messages will occur.

To configure the IBM Application Discovery server, complete the following steps:

1. Click IBM Application Discovery on the QUICK START page, or click the left-top button to navigate to **Configure > IBM Application Discovery Server**.
2. Enter all required information, including the host name, port number, user id and password.
3. Click Test Connection.
4. Click Save when the connection is successful.

When you configure the server, if errors occur, see "Troubleshooting the configuration of the Application Discovery server" on page 33 for reference.

Related information:

 [IBM AD Build User Guide](#)

Configuring the ADCD

Learn how to configure the ADCD.

The ADCD page requires Administrator privileges.

ADCD is updated approximately twice a year and is provided as a part of the product. To make the ADCD provided with version 11.0.0, or later, of IBM z Systems Development and Test Environment available for image creation and deployment, you need to copy the ADCD files to your FTP storage as follows:

1. Provide the required settings for the FTP storage.
2. In the base directory specified on the FTP storage settings page, create a directory labeled **adcd**, if it does not already exist.
3. In the adcd directory created in step 2, create a directory, using any label, for each ADCD you want to make available in the Tools, for example, **adcd/may2017**.
4. Transfer the ADCDTOOLS.XML file to the directory specified in step 3. The file describes the metadata for the Tools.
5. Create a volumes directory in the directory specified in step 3, for example, **adcd/may2017/volumes**.
6. Transfer all the volume files included with the ADCD to the directory specified in step 5.
7. The Tools will automatically locate any ADCD provided with version 11.0.0 or later of IBM z Systems Development and Test Environment on the **IMAGE from ADCD** page.

Note: The credentials for the FTP storage are the only credentials that the tools stores in its local database and are encrypted using AES 128-bit encryption.

Configuring the emulator licenses

Learn how to configure the emulator licenses.

The emulator licenses page also requires Administrator privileges.

The license servers used by each target z Systems Development and Test Environment instance needs to be configured within the Tools before the Target instances are created.

To configure the emulator licenses, follow these steps:

1. Click Emulator licenses on the QUICK START page, or click the left-top button to navigate to **Configure > Emulator licenses**.
2. In the Primary field, enter the fully qualified host name of the system to run the primary license manager.
3. Optional: In the Secondary field, enter the fully qualified host name of the system to run the secondary license manager.
4. Optional: In the Token Servers table, enter the ports and fully qualified host names of the systems to serve Rational tokens.

Operating

Learn how to use IBM z Systems Development and Test Environment Tools for development and test activities.

Creating application images

Learn how to create application images.

An application image is a collection of application components that originate from the same IBM z Systems and packaged together as a single deliverable for a target system. IBM z Systems Development and Test Environment Tools allows you to create application images from various sources:

- Existing IBM z Systems.
- Application Developers Controlled Distributions (ADCD). The z/OS Software distribution is bundled with the product.
- Existing IBM z Systems Development and Test Environment instances.

Option 1: Creating an application image from an existing IBM z System

Learn how to create an application image from an existing IBM z System.

To create an application image from an existing IBM z System, follow these steps:

1. Create application components from an existing IBM z System.
2. Create an application image by selecting the created components that are needed for an application image.

Creating application components from an existing IBM z System:

Learn how to create an application components from an IBM z System.

An application component is a collection of assets from an IBM z System that is intended to be reusable in a mix and match fashion with other components from the same IBM z System.

You can create application components from three resources:

- IBM z System datasets
- IBM z System volumes
- IBM Application Discovery projects

Creating application components from IBM z System datasets:

Learn how to create an application component from an IBM z System dataset.

IBM z Systems Development and Test Environment Tools supports creating application components from IBM z System datasets. To create a component from IBM z System datasets, you can input a filter string that is passed to z/OS via the REST API. Then, you can get a list of cataloged datasets that match the filter.

The filter string that is used to search dataset list from z/OS follows the same rule of Dsname level in Data Set List Utility panel fields except one difference. ISPF automatically adds `.*` to the end of each filter string, but the Tools needs wildcards at the end of the filter string to indicate that the filter string will match more characters. For example, 'IBMUUSER' in ISPF means 'IBMUUSER.*' in Tools.

For the wildcards definition, see the following table:

Wildcards	Definition
%	Any single character
*	Any number of characters within a single qualifier. Can follow or precede other parts of a qualifier as in a* or *b.
**	Zero or more complete qualifiers

Restriction: You cannot specify only '*' or '**' as the filter string.

- The restriction '**' is set to help you avoid retrieving all datasets unconsciously, which will be slow. But if you really want to retrieve all datasets, you can use '**.*.*'.
- The restriction '*' is set to help you avoid trying to retrieve all datasets but get datasets with only one qualifier. But if you really want to retrieve datasets with only one qualifier, you can use '*.*'.

To create application components from IBM z System datasets, complete the following steps:

1. Click IBM z System DATASETS on the QUICK START page, or click the left-top button to navigate to **Create-Component > Dataset - z System**.
2. Specify the host name of IBM z System you want to work with, and enter the required valid credentials.

Note: Contact your administrator if you do not have this information.

3. Enter a component name.
4. Optional: Enter comments that can help you identify your components.
5. Enter a qualifier, and then select all datasets that are needed for the component.
6. Click Create Component to complete.

When you create an application component, the information about IBM z System and the selected artifacts are saved and used when this component is included as a part of the application image creation process. No extraction of data occurs during this step.

Note: This process needs to be done by a system programmer, or someone that is familiar with IBM z System where the data is being extracted.

Creating application components from IBM z System volumes:

Learn how to create an application components from an IBM z System volumes.

To create application components from IBM z System volumes, follow these steps:

1. Click IBM z System VOLUMES on the QUICK START page, or click the left-top button to navigate to **Create-Component > Volume - z System**.
2. Specify the host name of IBM z System you wish to work with, and enter the required valid credentials.

Note: Contact your administrator if you do not have this information.

3. Enter a component name.
4. Optional: Enter comments that can help you identify your components.
5. Select all volumes you need for the component.
6. Click Create Component to complete.

When you create an application component, the information about IBM z System and the selected artifacts will be saved and used when this component is included as a part of the application image creation process. No extraction of data will occur during this step.

Creating application components from IBM Application Discovery projects:

Learn how to create an application component from an IBM Application Discovery project.

Before creating application components from IBM Application Discovery projects, you must ensure that:

- You are aware of all known limitation for creating application components from IBM Application Discovery projects. For more information, see “Limitations” on page 13.
- The connection is set up and tested through the IBM Application Discovery Configuration page. For more information, see “Configuring the IBM Application Discovery Server” on page 23.

To create application components from IBM Application Discovery projects, complete the following steps:

1. Click the IBM Application Discovery PROJECTS on the QUICK START page, or click the left-top button to navigate to **Create-Component > Projects - IBM AD**.
2. Select an IBM Application Discovery project that you set up on your Application Discovery environment.
3. Enter the host name of z System where all retrieved datasets are present.
4. Enter the port number to connect to z System.

Note: Currently, IBM z Systems Development and Test Environment Tools only uses the SSH port to connect to z Systems. No character is allowed in this number field.

5. Enter a component name.
6. Optional: Enter comments that can help you identify your components.
7. Select the datasets you need to transfer from your z System machine to target z Systems Development and Test Environment machine. Alternatively, you can search the terms you want.

Note: Currently, IBM z Systems Development and Test Environment Tools only supports datasets whose host names are entered. The datasets from IBM Application Discovery whose host name is not entered cannot be selected; an error message will be displayed when you deploy the application images.

8. Click Create Component to complete.

When you create an application component, the information about IBM z System and the selected artifacts will be saved and used when this component is included as a part of the application image creation process. No extraction of data will occur during this step.

After clicking Create Component, and the following two options are displayed, choose any of the options based on your needs:

1. Go to create Image page and create the Image.
2. Create another Application Discovery component.

When you create application components from IBM Application Discovery projects, if errors occur, see “Troubleshooting the application components creation from IBM Application Discovery projects” on page 33 for reference.

Creating an application image from an existing IBM z System:

Learn how to create an application images from an IBM z System.

An application image consists of a set of application components from the same IBM z System. After creating the application components, you need to combine the created application components into a single application image. An application image must contain only one application component that contains your system residence volumes and any number of other components that could represent your application, application data, or any number of subsystems such as CICS® or IMS™.

To create application images, follow these steps:

1. Click IBM z System COMPONENTS on the QUICK START page, or click the left-top button to navigate to **Create > Image - z System**.
2. Specify the host name of IBM z System you want to work with, and enter the required valid credentials. Then enter a new component name and optional comments.

Note: Contact your administrator if you do not have this information.

3. Enter an image name.
4. Optional: Enter comments that can help you identify your images.
5. Select all the components that you want for this image.
6. Click Schedule Image Creation to specify the time you want to extract IBM z system volumes, which is a part of the application image creation process. Or click **Create Image** to extract the volumes now.

Option 2: Creating an application image from ADCD

Learn how to create an application image from ADCD.

IBM z Systems Development and Test Environment Tools allows you to create an application image by selecting existing ADCD (Application Developers Controlled Distribution) components. You can create an application image and select the pre-packaged application components that contain IBM z/OS software, such as CICS 5.2 or 5.3, DB2® v11, and the required z/OS 2.2 components.

To create an application image from ADCD, complete the following steps:

1. Click Application Developers Controlled Distribution (ADCD) on the QUICK START page, or click the left-top button to navigate to **Create > Image -ADCD**.
2. Select an ADCD that is configured.
3. Enter an image name.
4. Optional: Enter comments that can help you identify your images.
5. Select all the components for which you want to create the application image.
6. Click Create Image to complete.

Option 3: Creating an application image from an existing z Systems Development and Test Environment

Learn how to create an application image from an existing z Systems Development and Test Environment.

IBM z Systems Development and Test Environment Tools also allows you to import an application image from an existing z Systems Development and Test Environment.

To import an application image from an existing z Systems Development and Test Environment, follow these steps:

1. Click IMPORT existing content on the QUICK START page, or click the left-top button to navigate to **Create > Import image**.
2. Enter an image name.
3. Optional: Enter comments that can help you identify your images.
4. Provide the hostname and Linux credentials of z Systems Development and Test Environment along with the location of the devmap file. Then, enter your user ID and password.
5. Click Import Image to complete.

Importing existing artifacts will create an application image that contains the devmap file and all volumes defined in that devmap file from the specified system. All artifacts of this application image will be stored on the system that is specified on the FTP storage.

Deploying application images

Learn how to deploy application images to z Systems Development and Test Environment instances in an automated way.

After you create application images, IBM z Systems Development and Test Environment Tools will automatically set up an entire z Systems Development and Test Environment for development and testing when it is needed. The only requirement for the deployment is that a clean Linux OS is installed.

To deploy the application images, complete the following steps:

1. Click **DEPLOY** on the **QUICK START** page, or click the left-top button to navigate to **Deploy**.
2. Select the application image that you want to deploy.
3. Select the target environment that the selected application image is deployed to.
4. Provide the required credentials in one of the following ways:
 - Click **Credentials** and supply the root credentials that can access the target system. Then, click **Next**.
 - Click **Certificate** and supply the root user ID along with your OpenSSH private key certificate file. If your key file requires a passphrase, enter it below. Then, click **Next**.

Note: You need to add your public key to the root `authorized_keys` file in the target system: `/root/.ssh/authorized_keys`.

5. Modify the central processors (cps) and system memory size that are allocated to the z Systems Development and Test Environment instance.
6. Select the automatic IPL option on the additional properties page.
 - If the application image is created from ADCD, the option is selected by default. If you want to issue the IPL command manually, you can clear the option.
 - If the application image is created from an existing IBM z System, the option is not selected by default. If you want the IPL to be done automatically, you can select the option.
 - If the application image is imported from an existing z Systems Development and Test Environment instance, the automatic IPL option is not supported in this release. You need to issue the IPL command manually after z Systems Development and Test Environment is started.
7. Click **Deploy Image**.

Your target systems can be your own on-premises physical machines with your own personal cloud, or the ones from IBM-managed cloud.

Using the deployment options in the Tools enables you to automatically:

1. Silently install the emulator to the target environment.
2. Configure the host Linux machine – iptables, license server and so on.

Note: While the installation and configuration require root credentials, running the emulator processes will be done by using a newly created user called **ibmsys1**. If this ID does not exist, it will be automatically created by the installation process. Also, a password as **sys1** will be generated. After the deployment process completes, it is recommended that you change this password to make it unique. The deployed volumes and devmap file that is named `aprof1` will be located in `/home/ibmsys1/volumes`.

3. Install the application image content.
4. Generate a devmap file for application images that are extracted from an IBM z System.
5. Modify the z/OS parameters for application images that are extracted from an IBM z System so that it is able to IPL in z Systems Development and Test Environment.
6. Start the emulator.

After the above deployment steps automatically completed, if you choose to issue IPL command manually, complete the following step:

1. Click the Monitor page, and expand the application image that is deployed to the system.
2. Find the section that is called Initial Program Load under the application image. This Initial Program Load contains the IPL command to issue.
3. SSH to the target environment, and login with the **ibmsys1** account.

Note: If the **ibmsys1** account was created by the Tools, then the password for this account is **sys1**.

4. Issue the command. For example, issue the command `ipl 0a80 parm 0a82au`.
If the IPL fails, you can issue the following commands in sequence from path `/home/ibmsys1/volumes`:

```
awsstop ---wait for few minutes for zDT to stop
ipl 0a80 parm 0a82CS --- Monitor console for any outstanding message.
awsstart aprof1 --- wait for few minutes for zDT to get ready.
```

After IPL is successful, you can use `ipl 0a80 parm 0a82au` to IPL next time.

Note: This process can be done by any application programmer or tester on-demand whenever they need a new environment.

Managing

Learn how to manage IBM z Systems Development and Test Environment Tools for development and test activities.

Monitoring the creation and deployment

Learn how to manage the Tools to monitor the status of all created assets and target systems.

The monitor page provides a dashboard that allows you to monitor inventory and check the status on all created assets and target systems.

When extracting IBM z System volumes for application image creation and deploying the application images, the process may take a long time to complete. IBM z Systems Development and Test Environment Tools allows you to track the progress of these actions from the monitor page.

The monitor page includes the status of the following parts.

- Application Image
- Target environments

Application Image status

The Application Image status include the following types:

- **Scheduled**
An application image is scheduled for creation at a later time.
- **In progress**
An application image is being created at this time.
- **Available**
An application image is fully constructed and available for use.
- **Needs attention**

An error occurred while creating an application image. When this status is displayed, you need to send the `/opt/ibm/zDT/Liberty/usr/servers/zDTServer/logs/messages.log` to IBM support.

- **Locked**

An application image cannot be used currently.

Target environment status

The Target environment status includes the following types:

- **Available**

A target environment is available for deployment.

- **Deploying**

An application image is being deployed to a target environment.

- **In use**

An application image has been successfully deployed to a target environment.

Note: Only one application image is allowed on a system at a given time, so this system cannot be used again until the application image is removed.

- **Needs attention**

An error occurred while deploying an application image. When this status is displayed, you need to send the `/opt/ibm/zDT/Liberty/usr/servers/zDTServer/logs/messages.log` to IBM support.

- **Offline**

A socket cannot be established to the target systems hostname on its SSH port.

Note: The monitor page is available for all users with different roles.

Scheduling

Learn how to schedule and specify a date and time to create an application image from a z System.

IBM z Systems Development and Test Environment Tools supports scheduling and specifying a date and time to create an application image from a z System. When creating application images from an IBM z System, if you want to schedule the extraction of these artifacts for another date or time, follow these steps:

1. Click **Schedule Image Creation**.
2. Specify the extraction date and time you want.
3. Select the timezone that is applicable for you.
4. Click **Schedule**.

After completing all above steps, the application images will be created whenever you specified.

Troubleshooting

Learn how to troubleshoot IBM z Systems Development and Test Environment Tools for development and test activities.

Setting up a Windows machine as an FTP server

To set up a Windows machine as an FTP server, try the following solutions and workarounds.

When you use z Systems Development and Test Environment Tools, and want to set up a Windows machine as the FTP server, complete the following setting:

1. Set up at least one user to connect to the FTP server.
2. Navigate to the **Configure > FTP storage** page.
3. Specify the base directory based on the shared folder of the specified user. For example, if the shared folder of the specified user is c:\test, then when you specify /sub-test as the base directory, the base directory on the FTP server will be c:\test\sub-test.
4. Set the Read, Write, and Create-dir permissions you want the user to have to the base directory.

Troubleshooting the configuration of the Application Discovery server

Learn how to troubleshoot the application components creation from IBM Application Discovery projects.

Table 1. Troubleshooting the configuration of the Application Discovery server

Symptom	Problem Determination
The credential is not valid.	Verify if the Application Discovery web service is set up as HTTP/HTTPS with basic authentication.
Unexpected end of file from server.	Verify if the Application Discovery web service is set up as HTTP/HTTPS, and select the toggle switch accordingly.
The connection is refused.	Verify if your host name, up and running, and port number are correct.
The connection is closed by remote host during handshake.	Verify if your host name, up and running, and port number are correct. Also, verify if the service is available via HTTP or HTTPS.

Troubleshooting the application components creation from IBM Application Discovery projects

Table 2. Troubleshooting the application components creation from IBM Application Discovery projects

Symptom	Problem Determination
When you enter the component creation page, you are redirected to the IBM Application Discovery configuration page.	Contact your administrator to check if IBM Application Discovery server is configured correctly through IBM Application Discovery configuration page.
You are not authorized to create components.	Contact your administrator to check if IBM Application Discovery server is configured correctly through IBM Application Discovery configuration page.
No project is listed in the IBM Application Discovery Projects drop-down menu.	Contact your administrator to check if the project is set up under the path <http>://<AD Host Name>:<Port>/ws/projects or <https>://<AD Host Name>:<Port>/ws/projects.

Table 2. Troubleshooting the application components creation from IBM Application Discovery projects (continued)

Symptom	Problem Determination
No dataset is listed in the drop-down menu after you select the IBM Application Discovery projects.	Contact your administrator to check if the project is created or updated under the path <http>://<AD Host Name>:<Port>/ws/projects/<Project Name>/datasets or <https>://<AD Host Name>:<Port>/ws/projects/<Project Name>/datasets.
Fewer or more datasets are listed in the IBM Application Discovery Projects drop-down menu.	Contact your administrator to check if the project is created or updated under the path <http>://<AD Host Name>:<Port>/ws/projects/<Project Name>/datasets or <https>://<AD Host Name>:<Port>/ws/projects/<Project Name>/datasets. For more information, see IBM AD Build User Guide to refresh or rebuild the project.
The selected dataset list is required.	Select the dataset list on the right panel of the page.
The port number is not valid.	No character is allowed.

Scenarios

Learn how to troubleshoot IBM z Systems Development and Test Environment Tools for development and test activities.

Provisioning CICS subsystem from a source z System

Learn how to provision a CICS subsystem from a source z System.

To provision a CICS subsystem from a source z System, complete the following steps:

1. Keep the CICS startup JCL in the partitioned data set(PDS), and build the Application Discovery project by using the JCL. For more information about how to build an IBM Application Discovery project, see IBM AD Build User Guide.
2. Configure the connection of the Application Discovery web service. For more information, see “Configuring the IBM Application Discovery Server” on page 23.
3. Create the application component from IBM Application Discovery, and select the project that is built to analyze the CICS startup JCL. For more information, see “Creating application components from IBM Application Discovery projects” on page 27.
4. Create the application image. For more information, see “Creating an application image from an existing IBM z System” on page 28.
5. Deploy the application image to the target machine. For more information, see “Deploying application images” on page 29.

Restriction: You need to manually transfer CICS dependencies that are not identified by the CICS startup JCLs, and the dependency list depends on the complexity of your CICS environment. You can identify some dependencies by reviewing IPL parms, especially BPXPRMCI, PROGCI, LPALSTCI where CI is the parm suffix used in IEASYSxx (xx is the load parm).

- BPXPRMxx has a list of USS (UNIX System Services) datasets that are used by CICS. It is defined in OMVS parameter of IEASYSxx, for example:
OMVS=(00,01,BP,IZ,RZ), SELECT BPXPRMCS
- LPALSTxx has a list of LPA datasets. It is defined in IEASYSxx as LPA parameter. For example:
LPA=00, SELECT LPALST
- PROGxx has a list of other APF that authorizes module load libraries that might not be in the CICS startup JCL. It is defined in IEASYSxx as follow:
PROG=(AB,AM,A0,A2, DYNAMIC APF LIST
SY,LB,LM,L0,L2,LL), LINKLIST LL SUFFIX SHOULD BE LAST

Chapter 4. Installing z System Development and Test Environment

Learn how to install the product.

The installation programs install the z Systems Development and Test Environment system, they do not install a z/OS distribution. A z/OS distribution must either be migrated from a current z/OS system or an existing z/OS system that is running on z Systems Development and Test Environment, or be downloaded from Passport Advantage® and installed separately.

The installation programs must be run from the root user ID.

z Systems Development and Test Environment prerequisites

Learn about hardware and software requirements for IBM z Systems Development and Test Environment.

For the latest and most up-to-date hardware and software requirements, you can go to Software Product Compatibility Reports and generate the reports for hardware and software requirements. Hardware and software requirements are also documented in the zPDT Guide and Reference.

Hardware provisioning

Learn about the hardware systems that can be used for z Systems Development and Test Environment.

PC System

A range of personal computer systems and Linux distributions can be used for z Systems Development and Test Environment. These configurations change over time, due to frequent personal computer hardware advances and new Linux releases. As a general statement, zDT works with any modern Intel compatible processor that is fully supported by the recommended Linux distributions.

The following sections list the hardware and software requirements to install and run z Systems Development and Test Environment. Ensuring your PC meets these requirements is the first step in installing z Systems Development and Test Environment. These requirements are a sufficient guide for that process, assuming your PC is dedicated to Linux. The zPDT Guide and Reference gives additional information that can be valuable. It lists the PC hardware that was used to test zPDT Version 1 Release 6, gives more detailed guidelines on the hardware and software requirements, and gives some considerations if either your PC is not dedicated to Linux or it runs in a virtual environment. For more information, see section 2.4, “Base configurations” and chapter 5, “zPDT installation” in the *zPDT Guide and Reference*.

USB Hardware Device

A USB hardware device is required for z Systems Development and Test Environment Personal Edition and releases before v10.0.

When licensing by using a USB hardware device, z Systems Development and Test Environment requires access to a USB hardware device that is activated with a license key called an update file. The USB hardware device is ordered through Passport Advantage in a media pack that is separate from the electronic media that contains the offering software. It can be connected directly to the USB port on a computer that is hosting the offering or it can be plugged in to the USB port of a separate computer, that is called a Sentinel Hardware Keys (SHK) license server and hosts the offering for distributing authentication to remote instances of the product through a TCP/IP network. Ideally, the USB hardware device is ordered at the same time as z Systems Development and Test Environment, or you are already running a separate computer that is acting as a license server.

Even in its simplest environment, which is a single USB hardware device that is connecting to a single PC that hosts z Systems Development and Test Environment, you must consider several issues. USB hardware devices must be activated. Depending on the type of license, the USB hardware device or a license manager must be activated with the number of CPs needed for all instances that use that device. The license key files have expiration dates, and they can interact with Rational tokens. In more complex environments, license servers can be activated. For an in-depth explanation of software-based licensing and USB hardware device planning, acquisition, and activation, see Chapter 5, “Enabling product operation,” on page 61.

Software Based license - License Manager instance on cloud Machine or VM Machine

Software based License Manager is introduced in IBM z Systems Development and Test Environment v10.0 and enables licenses to be obtained from software files.

When you use software-based licensing, z Systems Development and Test Environment requires access to a license manager that is activated with a software-based license key file to control the licensee's access to all or portions of the program. The license manager, sometimes referred as a software-based license server, manages all software-based license key files. License key files are obtained from the Rational License Key Center, and are specific to a customer account.

Each license key file is generated with a number of entitled emulated central processors which are referred as Central Processors or CPs. A single instance of z Systems Development and Test Environment requires at least 1 CP, and can run with a maximum of eight CPs. The number of CPs needed depends on the number of users, the types and amount of processing required.

The software-based license key file that is generated by the Rational License Key Center is called a license manager update file. It is created by the Rational License Key Center for a specific license manager, and only that license manager can be activated with that license manager update file. A license manager is activated with only one license manager update file. To change the number of entitled emulated CPs on a license manager, or to renew a software-based license key that is currently installed on a license manager, you must first return the license key, and then generate a new one. A single activated license manager can authenticate many instances of z Systems Development and Test Environment. The number of instances that can be authenticated varies based on the number of emulated Central Processors or CPs used in each instance.

When an instance of the offering starts, it seeks an activation from the license manager for the number of CPs specified in the emulator's device mapping file.

When an instance of the emulator is stopped, the CPs authenticated by the license manager become available for another instance. The activated license manager must remain available to the instance while the instance is running.

Note: License manager needs to be a static resource in any infrastructure configuration. In a virtualized or cloud infrastructure, it does not tolerate physical moving. If the license manager is manually or automatically moved inside the infrastructure, it will stop working. If you would like to move the License Manager, you need to return the license before moving, acquire the license after movement, and configure the License manager again.

Hardware requirements

Learn about the hardware platforms that are supported by z Systems Development and Test Environment.

z Systems Development and Test Environment instance

Hardware platforms that are supported:

- 64-bit x86 Intel or compatible servers

Minimum processor rating:

- As a general statement, zPDT works with any modern Intel processor that is fully supported by the recommended Linux distributions

Processor core requirement:

- The hardware or virtual machine system must have at least 1 more PC core than the total number of z Systems CPs for all concurrently running zPDT instances. The number of z Systems CPs allowed for each instance is defined in the processor statement of the device map for each instance, requiring a minimum of 1 CP and a maximum of 8 CPs. For an example of the processor statement, see “Defining the device map” on page 103.
- For standard installations, the following configuration is recommended: a 4-Core system with one core that is devoted to Linux and three cores that are devoted to z/OS.
- For Sysplex installations, use this configuration: an 11-Core system with three cores that are devoted to Linux and eight cores that are devoted to z/VM[®] and its guests.

Memory requirement:

- You need PC memory of *at least* 1 GB larger than the intended size of the emulated z Systems memory for ALL the concurrently running zPDT instances. The size of the z Systems memory to be used for zPDT operation is defined in the memory statement of the device map for each instance. 2 GB is a bare minimum for z Systems memory. 2 GB to 4 GB per emulated CP is a more realistic starting point.
- For the sysplex capability, because extra memory is required for z/VM, the coupling facility, and multiple z/OS guests, a minimum of 16 GB is needed.

Because z Systems Development and Test Environment reserves the full amount of physical memory that is assigned to the virtual z Systems machine, ensure that you have at least as much physical memory as your virtual z Systems machine requires, plus 1 GB for Linux.

Removable media requirements:

- If you are not using a product license server or a license manager, a suitable USB port must be available for the 1091 hardware key. Do not use an unpowered USB port expander when you are using zPDT. In particular, do not install the USB Hardware Device in an unpowered USB port expander. (The product license server, sometimes called a product license server, is described in “zPDT license servers” on page 208, and provides an alternative way to manage the device.)

Hard disk requirements:

50 GB to 100 GB of disk space is required for z/OS. The minimum required available disk space is twice the total required memory that is defined for the server instance. If you are using the sysplex capability, an extra 54 GB of disk space is required to hold the z/VM software distribution. Depending on how much software you load into each z/OS system that is running in a parallel sysplex configuration, you might need up to 220 GB per z/OS system.

Other hardware requirement:

- Disable Hyper-threading (if available) at the BIOS level. Hyper-threading can produce extreme slowdowns when z/OS is running spinloops. If many PC cores are available the slowdowns might be resolved before z/OS console messages are produced, indicating no problem other than reduced performance.

Product license server or license manager

When you activate the product with a USB hardware device, one option for making the license key files available to the program is to set up a remote license server and use network communications to enable independent installations of z Systems Development and Test Environment to be authenticated by a single server. Otherwise, each machine that is hosting z Systems Development and Test Environment requires a hardware device to be installed locally to make it available to the program.

Setting up a product license server to authenticate with a high capacity USB hardware device involves installing z Systems Development and Test Environment on a platform and installing a license key file, which is also called an update file, on a USB hardware device on the license server. The product license server must then be started and the clients must be configured to access it. For more information, see “Quick setup instructions for using and migrating the product license server” on page 97.

With software-based licensing, you can authenticate without a USB hardware device. Setting up a license manager to authenticate software-based licenses involves installing the z Systems Development and Test Environment license manager on a platform, starting the license manager, and installing a license key file on the license manager to activate it. The clients that use the license manager must then be configured to access it, and the license manager client function must be started. For more information, see “Software-based licensing” on page 63.

The systems that host the product license server or the license manager have a different set of requirements.

Hardware platforms that are supported:

- 64-bit x86 Intel or compatible servers

Minimum processor rating:

- 2.0 GHz or higher Intel Core 2 Duo, Generation 1 Intel i3, or equivalent processor required

Memory requirement:

- 2 GB of RAM is required.

Hard disk requirements:

- 12 GB of disk space is required for Linux and workspace.

Removable media requirements:

- If you are using a product license server, a suitable USB port must be available for the 1091 hardware key. Do not use an unpowered USB port expander when you are using zPDT. In particular, do not install the USB Hardware Device in an unpowered USB port expander. (The product license server, sometimes called a product license server, is described in “zPDT license servers” on page 208, and provides an alternative way to manage the device.)

Software requirements

Learn about software requirements for z Systems Development and Test Environment

z Systems Development and Test Environment instance

With z Systems Development and Test Environment, you can install the product directly on the operating system that is running on a supported hardware platform (native installations), or you can install the product on a supported operating system that is running in a supported virtualization environment (virtual installations). These operating systems and platforms are required:

Operating System requirements for native installations:

- Red Hat Enterprise Linux (RHEL) 6.0 or later minor release
- Red Hat Enterprise Linux (RHEL) 7.0 or later minor release
- Ubuntu 16.04.2 LTS or later minor release

Operating System platforms that are supported for virtual installations:

- Red Hat Enterprise Linux (RHEL) 6.0 or later minor release
- Red Hat Enterprise Linux (RHEL) 7.0 or later minor release
- Ubuntu 16.04.2 LTS or later minor release

You can run on later releases within each version. The recommended Linux version is RHEL 7.0 and Ubuntu 16.04.2 LTS. If problems are encountered, support for releases within the supported version is provided.

Virtualization Technologies supported:

- VMWare ESXi 5.1
- Support for KVM kernel modules
- Support for XEN hypervisor.

VMWare ESXi 5.1 must be run on 64-bit Intel compatible hardware.

Directly attached SCSI tape drives are not supported by z Systems Development and Test Environment within any virtualized environment.

When you are running z Systems Development and Test Environment in a virtual environment, you must have sufficient physical memory and processor cores to completely provision each instance of z Systems Development and Test Environment.

- For each instance of z Systems Development and Test Environment, the machine that is running your virtualization software must have one physical core for each CP in the instance, plus one physical core for Linux.
- Physical memory on the machine that is running the virtualization software must be sufficient to contain the total memory of all z Systems Development and Test Environment instances.
- Processors and memory that is used for virtual machines that are running z Systems Development and Test Environment are considered dedicated to z Systems Development and Test Environment. Careful planning must be done if other virtual machines are to run on the same hardware.
- Failure to provide sufficient physical processors and memory, or the over-commitment of memory or processor cores, might lead to severe performance problems and might cause unpredictable errors, or errors that are difficult to diagnose.

Product license server or license manager

Operating System requirements:

- Red Hat Enterprise Linux (RHEL) 6.0 or later minor release
- Red Hat Enterprise Linux (RHEL) 7.0 or later minor release

Virtualization Technologies supported:

- VMWare ESXi 5.1
- Support for KVM kernel modules (packaged with supported levels of RHEL)

Configuring the base Linux system

Install a supported Linux distribution, including the 32-bit runtimes.

To configure Linux to install and run z Systems Development and Test Environment:

- Disk Planning

Create at least three partitions for the hard disk drive:

- One for the root partition, containing all the normal root directories such as /usr, /lib, /home, /etc. Make this partition 10 - 20 GB.
- One for a swap partition for Linux. Make this partition 4 GB or larger.
- A large partition for emulated z Systems volumes and any other user data.

This suggested disk layout usage is not required. It is a starting point solely because it is simple and it isolates emulated z Systems volumes from the normal Linux files. This isolation is useful if you reinstall Linux (without disturbing your emulated volumes) and it might have minor performance benefits because it tends to reduce fragmentation in the disk space that is used for large emulated volumes.

Disk images for z/OS can be on any partition with sufficient space. When you install Linux, a separate partition for user data, including disk images, is needed

so that data is not lost during Linux system maintenance or upgrades. Disk space requirements depend on the number of z/OS volumes you use. If most of your z/OS volumes are 3390-3 images, a minimum amount of disk space that is required by the Development and Test Environment operation can be approximated by the calculation

$(3 \text{ GB} \times (\text{number of volumes})) + (2 \times (\text{memory in your z/OS machine}))$

These examples assume that you are running z Systems Development and Test Environment under the user ID of `ibmsys1`.

Most of the z Systems Development and Test Environment related files that are mentioned in the customization scenario 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 z Systems Development and Test Environment. The hardware clock was set to use Coordinated Universal Time (UTC) because that is required by the USB hardware device. The Linux user ID that was used is `ibmsys1`.

- **Important:** You must include Linux 32-bit support, which is an option during Linux installation. Before you install z Systems Development and Test Environment, you must ensure that the 32-bit runtime libraries in Linux are installed. These runtime libraries can be installed as part of the installation of Linux or can be added later by installing the `libstdc++ 32-bit` package for your Linux distribution. Currently, Red Hat distributions do not install the 32-bit libraries during a default installation.

Use the Linux command `rpm -qa | grep libstdc` to verify the 32-bit runtime library installation. If you see rpms with the name `libstdc` in formats similar to the following, then 32-bit runtime libraries are installed. Formats vary based on operating system.

```
libstdc++xx-32bit
libstdc++xx-*32bit
libstdc++-xx-xx.i686.rpm
libstdc++*.i686.rpm
```

After you finish installing z Systems Development and Test Environment, one of the final steps is to run the `z1090instcheck` command. This command verifies the proper Linux 32-bit runtime libraries are installed.

- If you are using software-based licensing, the license manager and the license manager client function also require a 32-bit version of the Linux `glibc` libraries. Before you install the license manager or start the license manager client on an instance of z Systems Development and Test Environment, you must ensure that the 32-bit versions of the Linux `glibc` libraries are installed.

These `glibc` libraries can be installed as part of the installation of Linux or can be added later by installing the `glibc-32bit` package for your Linux distribution. Currently, Red Hat distributions do not install the 32-bit libraries during a default installation.

Use the Linux command `rpm -qa | grep glibc-32bit` to verify the 32-bit `glibc` library installation. If you see rpms with the name `glibc` in formats similar to the following, then the 32-bit `glibc` library is installed. Formats vary according to operating system and level.

```
Glibc.i686
Glibc-32bit
```

- **System Time**

Set your hardware clock to UTC time to avoid problems when Daylight Saving Time starts and stops. The USB Hardware Device is sensitive to the hardware clock time and does not operate if the time appears to move backward. If the machine is shared with another operating system that expects local time (instead

of UTC time), you might experience a one-hour non-operational time when you shift from Daylight Saving Time to standard time.

- Install a TN3270e client for the MVS™ console

The following TN3270e clients are examples of clients that can be used with the recent z Systems Development and Test Environment offerings:

- x3270 (recent versions)
- Recent Personal Communications releases (running on Windows systems)

The most commonly used TN3270e client is x3270 running on the native Linux host. This client is used in the sample start script for z Systems Development and Test Environment. Frequently, it is not included with Linux distributions. An x3270 package is usually a single rpm, such as:

x3270-3.2.20-467.1.x86_64.rpm

Note: Other x3270 levels can be used or another 3270 emulator can be used for the MVS console. Also, 3270 emulators can be used for the MVS console that are installed remotely, and remove the need to have a 3270 emulator on the native Linux. For more information, see “IPLing z Systems Development and Test Environment from a remote emulated terminal for the system console” on page 109.

- Firewall considerations

You must manage whatever firewall and other security functions that you install with your Linux. Initially disabling any firewall when first working with zPDT simplifies configuration and operation. After you are familiar with zPDT operation, you can reestablish the firewall functions. If you have external TCP/IP connections (for example, local 3270 connections, OSA connections, product license server or license manager connections, or CTC connections) you must provide appropriate port *holes* in any firewall you use.

Paragraph 13.15, “TCP/UDP ports” in the zPDT Guide and Reference, describes the ports that are used for normal zPDT operations. Port 1947 is also used by the license manager, and must be allowed through the firewall. If your firewall is based on iptables (as is common for most current Linux releases), commands such as those shown in the following example might be used. This example is for setting a rule to allow any emulated local 3270 session traffic through the firewall, and then displaying the rules for the filter table afterward

```
$ su (switch to root)
# iptables -I INPUT -p tcp --dport 3270 -j ACCEPT
# iptables -L -n
# exit (leave root)
```

These commands would need to be entered from a *root* user ID each time the server Linux system is started, and you want to pursue some form of automation for entering these commands. Depending on how you connect z/OS to the external network, you might also need to provide appropriate port *holes* in your firewall to allow traffic to access z/OS ports, or you might need to use Network Address Translation functions in your firewall for this traffic.

The customizations described in this Guide eliminate the need to modify your firewall for z/OS ports. For more information, see “Setting up TCP/IP” on page 134.

For more considerations for Linux installation and configuration, see chapter 5.2, “Linux installation” in the zPDT Guide and Reference.

Migrating from previous version of z Systems Development and Test Environment

Learn about how to migrate from previous version of z Systems Development and Test Environment.

If migrating from z Systems Development and Test Environment V10.0, 10.0.1, or migrating from Rational Development and Test Environment version 9.5 or earlier, you must uninstall the previously installed version, and then install z Systems Development and Test Environment version 11.0. Use the following steps:

1. To uninstall the older version of z Systems Development and Test Environment, see “Uninstalling a previous version of z Systems Development and Test Environment” on page 46
2. To install z Systems Development and Test Environment v11.0, see Chapter 4, “Installing z System Development and Test Environment,” on page 37

To migrate from 11.0 to 11.0.1, see “Updating IBM z Systems Development and Test Environment” on page 51.

Note: z Systems Development and Test Environment Tools doesn't support older version of z Systems Development and Test Environment. So, to use z Systems Development and Test Environment Tools, you need to uninstall older version of z Systems Development and Test Environment and install z Systems Development and Test Environment V11.0.

Uninstalling a previous version of z Systems Development and Test Environment

Learn how to uninstall a previous version of z Systems Development and Test Environment.

To uninstall any supported version, use the following steps.

Version 9.0 or above	<ul style="list-style-type: none">• Stop any currently active instance of z Systems Development and Test Environment.<ul style="list-style-type: none">– To stop a base instance of z Systems Development and Test Environment, see Chapter 7, “Starting and stopping z Systems Development and Test Environment,” on page 107.– To stop a license manager, see “Stopping and uninstalling the license manager” on page 69.– If you are running a product license server:<ol style="list-style-type: none">1. Enter the uimserverstop command from the user ID that started the UIM server on the product license server. Do not enter this command as root.2. Enter these two commands to stop the product license server:<pre># cd /opt/safenet_sentinel/common_files/sentinel_k # ./loadserv stop</pre>• Start Installation Manager• Select Uninstall• Select the Installation package for Rational Development and Test Environment for z Systems• Select Uninstall• Uninstall Installation Manager
----------------------	--

Version 8.5	<ul style="list-style-type: none"> • Stop any currently active instance of Rational Development and Test Environment for z Systems • From a Linux console that is running as root, go to the directory that contains Rational Developer for z Systems Unit Test. The default installation directory is /tmp/IBM Rational Development and Test Environment for z Systems V8.5.x where <i>x</i> is the version that is installed. • Go to the subdirectory entitled Uninstall_IBM Rational Development and Test Environment for z Systems V8.5.x • Run the command <code>./'Uninstall IBM Rational Development and Test Environment for z Systems V8.5.x'</code> This command starts an InstallAnywhere uninstall wizard. • When the wizard finishes, you see the message “All items were successfully uninstalled”
Version 8.0	<ul style="list-style-type: none"> • Stop any currently active instance of Rational Development and Test Environment for z Systems • From a Linux console that is running as root, go to the directory that contains Rational Developer for z Unit Test. The default installation directory is /tmp/Rational Developer for z Unit Test V8.0.x where <i>x</i> is the version that is installed. • Go to the subdirectory entitled Uninstall_Rational Developer for z Unit Test V8.0.x • Run the command <code>./"Uninstall_Rational Developer for z Unit Test V8.0.x"</code>. This command starts an InstallAnywhere uninstall wizard • When prompted, select to do a complete uninstall • When the wizard finishes, you see the message “All items were successfully uninstalled”

Note: z Systems Development and Test Environment Tools doesn't support older version of z Systems Development and Test Environment. So, to use z Systems Development and Test Environment Tools, you need to uninstall older version of z Systems Development and Test Environment and install z Systems Development and Test Environment V11.0.

Uninstalling a version of z Systems Development and Test Environment and then reinstalling a newer version does not affect any previously installed z/OS distribution. In other words, any existing z/OS distribution that runs with one level of z Systems Development and Test Environment runs with a newer version, without requiring any form of reinstallation of the z/OS volumes. The z/OS

volumes store all of the customizations and data from the last time the distribution was run. The DEVMAP developed for running with that z/OS distribution on the older version runs unchanged with the newer version. For more information, see “Defining the device map” on page 103.

For information about uninstalling the license manager, see “Stopping and uninstalling the license manager” on page 69.

Installing z Systems Development and Test Environment interactively

Learn how to install the product interactively.

In order to install the product, you need to acquire the z Systems Development and Test Environment image from Passport Advantage. Download the package to your desired directory. Please remember you should have proper accesses to the directory to run the program. If not, change the access level of directory or installation file using the following Linux command:

```
chmod 755 <file name> or <directory name>
```

After downloading the package, you need to unzip the package by issuing the command `tar -xvzf filename.tgz`. The unzipped package includes the installer and `readme.txt`. To know more about the installation steps, see `Readme.txt` file.

Installing z Systems Development and Test Environment Personal Edition

Learn how to install z Systems Development and Test Environment Personal Edition interactively.

To install z Systems Development and Test Environment Personal Edition interactively, follow these steps:

1. Go to the directory where installation program is present.
2. Change the authority of installer as below:

```
chmod 755 <installer name>
```

For example, `chmod 755 zDT_Install.x86_64`.
3. Execute the installer as below:

```
./<Installer Name>
```

For example, `./zDT_Install.x86_64`.
4. Type **1** to select IBM z Systems Development and Test Environment Personal Edition to install.
5. Type **1** to install the product.
6. Press ENTER, and read the license agreements carefully. Then at the end of license, enter 'Yes' to accept or 'No' to decline the terms.

To install the network configurations. Select "Yes" when the following message is displayed. By selecting Yes, your z/OS will start using the IP of your Linux machine. This option is useful, if you want to share your Linux IP address with your z/OS. Without providing a dedicated and static IP address to z/OS, you can interact with your z/OS and applications directly. Also, before selecting "Yes", consider that the network configuration will make changes in your Linux and networking functions, allow Linux firewall to reroute TCP/IP packets to z/OS, and update the TCP/IP specifications in the z/OS disks.

Important: Remember to download the volumes of ADCD May 2017 edition from passport advantage and unzip the *.gz volume on the same machine where you are performing the installation of z Systems Development and Test Environment v11.

7. Wait for the installation to complete.
8. Run following command to see if installation is successful.

```
RHEL
rpm -qa | grep z1091
rpm -qa | grep shk-server
rpm -qa | grep sntl-sud
UBUNTU
dpkg -l | grep z1091
dpkg -l | grep shk-server
dpkg -l | grep sntl-sud
```

You should see respective z Systems Development and Test Environment driver listed.

Note: The above procedure needs to be executed from root User ID.

Installing z Systems Development and Test Environment License Manager

Learn how to install z Systems Development and Test Environment License Manager interactively.

To install z Systems Development and Test Environment License Manager interactively, follow these steps:

1.
 - a.
 - 1)
1. Go to the directory where installation program is present.
2. Execute the following command
./<Installer Name>
For example, ./zD&T_Install.x86_64.
3. Type 2 to select IBM z Systems Development and Test Environment License Manager to install.
4. Type 1 to install the product.
5. Press ENTER, and read the license agreements carefully. Then at the end of license agreements, enter 'Yes' to accept or 'No' to decline the terms.
6. Read the following warning message carefully.

*****Warning*****

License manager needs to be a static resource in any infrastructure configuration. In a virtualized or cloud infrastructure, it does not tolerate physical moving. If the license manager is manually or automatically moved, return the license before moving, acquire the license after movement, and configure the License Manager again.

Do you accept the above terms?(y/n)

7. Wait for the installation to complete.
8. Run following command to see if installation is successful.

```
RHEL
rpm -qa | grep aksusbd
rpm -qa | grep UIM
UBUNTU
dpkg -l | grep aksusbd
dpkg -l | grep UIM
```

You should see respective z Systems Development and Test Environment driver listed.

Note: The above procedure needs to be executed from root User ID.

Installing IBM z Systems Development and Test Environment silently

Learn how to install the product silently.

In order to install the product, you need to acquire the z Systems Development and Test Environment image from Passport Advantage. Download the package to your desired directory. Please remember you should have proper accesses to the directory to run the program. If not, change the access level of directory or installation file using the following Linux command:

```
Chmod 755 <file name> or <Directory Name>
```

After downloading the package, you need to unzip the package by issuing the command `tar -xvzf filename.tgz`. The unzipped package includes the installer and `readme.txt`. To know more about the installation steps, see `Readme.txt` file.

Note: If you install silently, you will not see the option to review the license. Installation process assumes that you have reviewed the license before installation. Read the license as in the installation media or use command line installation option.

Installing z Systems Development and Test Environment Personal Edition

Learn how to install z Systems Development and Test Environment Personal Edition silently.

To install z Systems Development and Test Environment Personal Edition silently, follow these steps:

1. Go to the directory where installation program is present.
2. Execute the following command

```
./zDT_Install.x86_64 --install --zdtpeedition
```

Note: If you install silently, you will not see the option to review the license. Installation process assumes that you have reviewed the license before installation. Read the license as in the installation media or use command line installation option.

3. Execute the following command to install the personal edition with automated network installation.

```
zDT_Installx86_64 --install --zdtpeedition --net
```

Important: Remember to download the volumes of ADCD May 2017 edition from passport advantage and unzip the ***.gz** volume on the same machine where you are performing the installation of z Systems Development and Test Environment v11.

4. Run the following command to see if installation is successful.

```
RHEL
rpm -qa | grep z1091
rpm -qa | grep shk-server
rpm -qa | grep sntl-sud
```



```

UBUNTU
dpkg -l | grep z1091
dpkg -l | grep shk-server
dpkg -l | grep sntl-sud

```

You should get the output respectively as below.

```

z1091-1-7.49.31.01.x86_64
zpdtd-shk-server-1.3.1.5-0.i386
sntl-sud-7.5.5-0.i386

```

Note: The above procedure needs to be executed from root User ID.

Installing z Systems Development and Test Environment License Manager

Learn how to install z Systems Development and Test Environment License Manager silently.

To install z Systems Development and Test Environment License Manager silently, follow these steps:

1. Go to the directory where installation program is present.
2. Execute the following command

```
./zDT_Install.x86_64 --install --zdtlicense
```

Note: If you install silently, you will not see the option to review the license. Installation process assumes that you have reviewed the license before installation. Read the license as in the installation media or use command line installation option.

3. Run following command to see if installation is successful.

```

RHEL
rpm -qa | grep aksusbd
rpm -qa | grep UIM
UBUNTU
dpkg -l | grep aksusbd
dpkg -l | grep UIM

```

You should get the output respectively as below

```

aksusbd-7.40-1.i386
UIM-1-7.49.31.01.x86_64

```

Note: The above procedure needs to be executed from root User ID.

Updating IBM z Systems Development and Test Environment

Learn how to use graphical user interface or silent install method to update the product.

In order to update the product, please remember you should have proper accesses to the directory to run the program. If not, change the access level of directory or installation file using the following Linux command:

```
Chmod 755 <file name> or <Directory Name>
```

After downloading the package, you need to unzip the package by issuing the command `tar -xvzf filename.tgz`. The unzipped package includes the installer and readme.txt. To know more about the installation steps, see Readme.txt file.

Updating z Systems Development and Test Environment interactively

Learn how to update the product interactively.

Before updating z Systems Development and Test Environment, you need to ensure that:

- If you need to update License Manager, make sure that UIMSERVER and AKSUSBD are stopped by issuing `/usr/z1090/bin/UIMSERVERSTOP` or `service aksusbd stop` from root user id.
- If you need to update Personal Edition or instance, make sure that `awsstop` command was issued from non-root userid.

To update the product interactively, follow these steps:

1. Go to the directory where installation program is present.
2. Execute following command.
`./<Installer Name>`
For example, `./zD&T_Install.x86_64`.
3. Type the corresponding numbers (1, 2, 3) to select the product to update.
4. Type **2** to update the selected product in step 3.
5. Press ENTER, and read the license agreements carefully. Then at the end of license, Enter 'Yes' to accept or 'No' to decline the terms.
6. Wait for the process to complete.

Note: This process of uninstall or update is valid only for IBM z Systems Development and Test Environment version 11 or above. For lower IBM z Systems Development and Test Environment or RD&T version, follow respective documentation.

Updating z Systems Development and Test Environment silently

Learn how to update the product silently.

Before updating z Systems Development and Test Environment, you need to ensure that:

- If you need to update License Manager, make sure that UIMSERVER and AKSUSBD are stopped by issuing `/usr/z1090/bin/UIMSERVERSTOP` or `service aksusbd stop` from root user id.
- If you need to update Personal Edition or instance, make sure that `awsstop` command was issued from non-root userid.

To update the product silently, follow these steps:

1. Go to the directory where installation program is present.
2. Use program usage:
`./zDT_Install.x86_64 --update --<product>`

where '`<product>`' can be one of the following:

zdtpedition

IBM z Systems Development and Test Environment Personal Edition

zdtlicense

IBM z Systems Development and Test Environment License Manager

zdttools

IBM z Systems Development and Test Environment Tools

3. Wait for the process to complete.

Note: This process of uninstall or update is valid only for z Systems Development and Test Environment version 11 or above. For lower z Systems Development and Test Environment or RD&T version, follow respective documentation.

Uninstalling IBM z Systems Development and Test Environment

Learn how to use interactive or silent method to uninstall the product.

In order to uninstall the product, please remember you should have proper accesses to the directory to run the program. If not, change the access level of directory or installation file using the following Linux command:

Chmod 755 <file name> or <Directory Name>

Uninstalling z Systems Development and Test Environment interactively

Learn how to uninstall the product interactively.

To uninstall the product interactively, follow these steps:

1. Go to the directory where installation program is present.
2. Execute following command.
`./<Installer Name>`
For example, `./zD&T_Install.x86_64`.
3. Type the corresponding numbers (1, 2, 3) to select the product to uninstall.
4. Type **3** to uninstall the selected product in step 3.
5. Wait for the process to complete.

Note: This process of uninstall or update is valid only for z Systems Development and Test Environment version 11 or above. For lower z Systems Development and Test Environment or RD&T version, follow respective documentation.

Uninstalling z Systems Development and Test Environment silently

Learn how to uninstall the product silently.

To uninstall the product silently, follow these steps:

1. Go to the directory where installation program is present.
2. Use program usage:
`./zDT_Install.x86_64 --uninstall --<product>`

where '<product>' can be one of the following:

zdtpedition

IBM z Systems Development and Test Environment Personal Edition

zdtlicense

IBM z Systems Development and Test Environment License Manager

zdttools

IBM z Systems Development and Test Environment Tools

3. Wait for the process to complete.

Note: This process of uninstall or update is valid only for z Systems Development and Test Environment version 11 or above. For lower z Systems Development and Test Environment or RD&T version, follow respective documentation.

After you install z Systems Development and Test Environment

After you install z Systems Development and Test Environment, you must make minor changes to your kernel configuration and add a few environment variables for the user IDs that are going to run z Systems Development and Test Environment.

After the installation is complete, you must complete these steps before you can enable and start the product.

If you installed the License Manager feature, you must start the UIM Server. For more information, see “Starting the license manager” on page 55.

If you installed the base z Systems Development and Test Environment and not the License Manager feature, you must add a few lines to `/etc/sysctl.conf` and `$HOME/.bashrc` on your Linux system. These updates can be made by using the two zPDT commands **aws_sysctl** and **aws_bashrc**. For more information about these two commands, see the following topics. The zPDT Guide and Reference includes additional information to consider, based on your specific Linux installation.

Updating `/etc/sysctl.conf`

Run the **aws_sysctl** zPDT command to update several of the Linux kernel parameters in `etc/sysctl.conf` that allow for zPDT execution.

Run the **aws_sysctl** zPDT command from root:

```
# /usr/z1090/bin/aws_sysctl
```

This command updates several of the Linux kernel parameters in `etc/sysctl.conf` that allow for zPDT execution, and then runs `/sbin/sysctl` to put them in effect. When prompted, enter `yes` to proceed. These changes are sufficient for most zPDT users. For the details on which parameters are updated, and which parameters might require change for especially large zPDT instances, see the *zPDT Guide and Reference* (SG24-8205).

Note: This step is only applicable for z Systems Development and Test Environment v11.0 and previous version.

Updating `$HOME/.bashrc`

Run the **aws_bashrc** command to update the `.bashrc` file in the home directory of the user IDs that run z Systems Development and Test Environment.

After you update the `/etc/sysctl.conf` file, you must update the `.bashrc` file in the home directory of the user IDs that run z Systems Development and Test Environment. Because it is assumed that you are running as `ibmsys1`, from the `/home/ibmsys1/` directory, run the following zPDT command as follows:

```
$ /usr/z1090/bin/aws_bashrc
```

This command adds the appropriate zPDT PATH statements and updates some user limits in `.bashrc` to allow zPDT execution. When prompted, enter yes to proceed. See the zPDT Guide and Reference for the details on which statements are added or updated.

Double-check the entries in these Linux files. Errors here might be difficult to detect later.

After the `.bashrc` file is updated for a user ID, and a new Linux console is started for that user ID to activate those changes, any zPDT commands that can be validly run by that user ID can be run from any directory. These commands include the ones that are identified in the IBM Knowledge Center, such as the `z1090instcheck` command, and the commands that are identified in the zPDT Guide and Reference.

Note:

1. You must be in the `/usr/z1090/bin` directory to enter the zPDT commands that are run from root.
2. This step is only applicable for z Systems Development and Test Environment v11.0 and previous version.

Checking the installation

After you update `$HOME/.bashrc`, start a new Linux console as user `ibmsys1` to activate the changes and run the `z1090instcheck` command.

If the command result shows errors, fix the errors before proceeding. The zPDT Guide and Reference contains more detailed information on the output that is returned from this command.

Note: The `z1090instcheck` instruction works for both z1090 machine types and z1091 machine types. There is no `z1091instcheck` instruction.

Starting the license manager

If you installed the License Manager feature, the license manager function is automatically started. However, you must start the UIM server, generate and install software-based license key files, and configure the license manager clients to access those keys.

For the remaining steps to configure the license manager and clients, see these topics:

- “Installing and starting the license manager” on page 68
- “Activating a license manager” on page 71
- “Activating and configuring a license manager client” on page 77

Installing an operating system on z Systems Development and Test Environment

Learn how to install supported operating systems on the z Systems Development and Test Environment machine emulator.

Installing z/OS volume images

Learn how to install and configure z/OS volume images for z Systems Development and Test Environment. You can install and customize a z/OS distribution, whether it is a migrated customer z/OS system, a z/OS Application

Developers Controlled Distribution (ADCD) for z Systems Development and Test Environment, or a z/OS distribution from a previous release of z Systems Development and Test Environment.

Note: Any z/OS distribution that is supplied with z Systems Development and Test Environment is a unique distribution that was created specifically for installations that are entitled to z Systems Development and Test Environment, based on the Application Developers Controlled Distribution (ADCD). They are referred to in this IBM Knowledge Center as the z/OS ADCD.

The customizations in this IBM Knowledge Center provide insight into the configuration process and security considerations and result in some additional working subsystems. While you can use any z/OS distribution without implementing these customizations, connectivity to your z/OS system might be limited.

Obtaining a z/OS distribution

You can obtain a z/OS distribution to run with z Systems Development and Test Environment from three sources.

- If you currently have a z/OS license, you can migrate the z/OS volumes from your z/OS system to your z Systems Development and Test Environment instance.
- You can download from Passport Advantage the z/OS ADCD volumes made available with z Systems Development and Test Environment
- You can use a previous z/OS distribution that runs with an older version of Rational Development and Test Environment for z Systems.

z Systems Development and Test Environment requires every z/OS volume in a z/OS distribution be a Linux file in an emulated 3390 format compatible with zPDT. The z/OS volumes in the z/OS ADCD for z Systems Development and Test Environment and in previous releases of z Systems Development and Test Environment are in the Linux file format. If you migrate volumes from your own z/OS distribution, migration tools are provided for the conversion.

Each method of obtaining the z/OS distribution is described. After it is obtained, any z/OS volume image in the emulated 3390 format can then be transmitted between Linux systems by any available file transfer protocol, such as FTP.

Migrating z/OS volumes from a z/OS system:

You can use several methods to migrate z/OS volumes from your z/OS system to your z Systems Development and Test Environment instance.

One method is to use the migration utility that is supplied in the `/usr/z1090/bin` directory after installation of z Systems Development and Test Environment. This utility uses a client/server approach in which the server transmits the requested z/OS volumes from the z/OS system to the Linux image. The client formats each volume to the emulated 3390 format that is used by zPDT. For information on how to use this migration utility to migrate z/OS volumes, see Chapter 15, DASD Volume Migration, in the zPDT Guide and Reference.

If you already have a running z/OS distribution with z Systems Development and Test Environment, another method of moving z/OS volumes between systems is using the ADRDSSU utility. This method is described in section 12.12, "Moving 3390 Volumes." In this scenario, you can move z/OS volumes to a z/OS system

that is running on z Systems Development and Test Environment. After the volumes are moved, they are in the proper emulated format. Chapter 12, "Minor z/OS notes," has other helpful information on how to use a z/OS distribution.

z Systems Development and Test Environment Version offers a new method to migrate z/OS volumes from your z/OS system by using z Systems Development and Test Environment Tools. For more information, see "Option 2: Creating an application image from ADCD" on page 29

Installing May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment:

Learn how to download and install the z/OS 2.2 Application Developers Controlled Distribution for z Systems Development and Test Environment May 2017 edition.

This z/OS 2.2 distribution is a unique distribution that was created specifically for sites that are entitled to z Systems Development and Test Environment, based on the Application Developers Controlled Distribution (ADCD). It is referred to as the z/OS 2.2 ADCD. It contains many of the products and subsystems you expect in a z/OS system. For a list of the products, volumes, and information about the system itself, see "z/OS 2.2 ADCD for z Systems Development and Test Environment" on page 143.

Before you install this z/OS 2.2 ADCD, you must first install z Systems Development and Test Environment. For instructions, see Chapter 4, "Installing z System Development and Test Environment," on page 37. Then do these steps to download and install the z/OS 2.2 ADCD for IBM z Systems Development and Test Environment.

Important: Any z/OS ADCD for IBM z Systems Development and Test Environment, and any related software, are supplied as a convenience. IBM does not support the operating system distributions that are made available with z Systems Development and Test Environment. You can apply fixes to any z/OS distribution through normal SMP/E facilities.

Downloading a z/OS 2.2 ADCD:

For the steps to download, install, and customize your z/OS ADCD for z Systems Development and Test Environment, select the appropriate version tab and follow the instructions in the z Systems Development and Test Environment download information technote.

The download information technote is available at <http://www.ibm.com/support/docview.wss?uid=swg21682519>.

"z/OS 2.2 ADCD for z Systems Development and Test Environment" on page 143 describes what volumes can be downloaded with this distribution and their general contents. To reduce the amount of disk space you use, you can download and install only the volumes that are required and the subsystems you use. In particular, the volume images for CICS, IMS, DB2, and WebSphere Application Server products can be selectively installed as required, choosing which products and which releases are appropriate for your needs. To determine the subsystems that are contained on each volume, see the description of each volume. Download and install the volumes on the instance that runs z Systems Development and Test Environment.

At a minimum, install all the base z/OS products, which are all the volume images that are not associated with the following products:

- CICS 5.3, and 5.2
- IMS versions 14 and 13
- DB2 version 12 and 11
- WebSphere Application Server versions 9.0 and 8.5

This set of volumes includes the base z/OS products:

- SARES1, D2RES1, D2RES2, D2CFG1, D2SYS1, D2USS1, D2USS2, D2PRD1, D2PRD2
- D2PRD3, D2DIS1, D2DIS2, D2PAGA, D2PAGB, D2PAGC, D2KAN1, D2BLZ1

Most of the z Systems Development and Test Environment related files that are mentioned in the customization scenario are stored within the `ibmsys1` home directory. This IBM Knowledge Center assumes that the 2 MVS IPL volume images, with file extension `ZPD`, are downloaded into the directory structure `/home/ibmsys1/z1090`. The remaining virtual 3390 volumes, with file extension `gz`, are downloaded into the directory structure `/home/ibmsys1/z1090/disks`. Any z Systems Development and Test Environment scripts are assumed in `/home/ibmsys1/z`. The directory structure `home/ibmsys1/z1090/disks` was used because that directory structure complies with the structure created when you start the z Systems Development and Test Environment, as described in “Configuring the base Linux system” on page 42.

Installing the z/OS volume images:

The downloaded z/OS volume images are in the `ZPD` and `gz` file extension formats. `ZPD` files require installation by using the **Z1091_AD_CD_install** command, and are stored with a name `volser.ZPD`. All other z/OS volume images are compressed files that are stored with a name `volser.gz`, requiring only the **gunzip** command for installation. In both cases, `volser` is a six character volume serial of the z/OS volume. In z/OS Version 2 Release 2, only 2 IPL volume images require installation by using the **Z1091_AD_CD_install** command: `D2RES1.ZPD` and `SARES1.ZPD`.

The process of installing the z/OS volume images is to use the **Z1091_AD_CD_install** command to install `ZPD` volume images, and use the **gunzip** command to decompress any other z/OS volumes needed. When installed or decompressed, most volumes are in emulated 3390-9 format, which are approximately 8.5 GB. Run all commands as the user that runs z Systems Development and Test Environment, in this example `ibmsys1`.

Perform the following two commands to install the `ZPD` volume images into the `/home/ibmsys1/z1090/disks` directory:

```
$ Z1091_AD_CD_install /home/ibmsys1/z1090/D2RES1.ZPD /home/ibmsys1/z1090/disks/D2RES1
$ Z1091_AD_CD_install /home/ibmsys1/z1090/SARES1.ZPD /home/ibmsys1/z1090/disks/SARES1
```

Perform these three commands to decompress all other z/OS volume images in that directory, and give the correct permissions for all z/OS volume image files.

Note: For some images, the **gunzip** command might produce a message about an unexpected end of file. If the `*.gz` files were correctly downloaded, that message can be safely ignored. Also, the **gunzip** command as shown here deletes the original `.gz` files. To keep the original file, enter **gunzip -k *.gz**. The **-k** parameter is available on later versions of **gunzip**.


```
$ cd /home/ibmsys1/z1090/disks
$ gunzip *.gz
$ chmod 755 *
```

All MVS volume images are in /home/ibmsys1/z1090/disks.

Most of these products and subsystems run without any customization, and some need some customization or initialization to make them work. The customizations to establish TCP/IP connectivity and update the infrastructure to allow z/OS 2.2 to properly run on z Systems Development and Test Environment are described in “Customizing the May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 124.

Using a previous z/OS distribution:

Any existing z/OS distribution that runs with one level of z Systems Development and Test Environment and is a z/OS release that was tested with the level of zPDT contained in the newer version of z Systems Development and Test Environment runs with the newer version of the product.

See “zPDT Releases” in the zPDT Guide and Reference for information on which z/OS releases were tested with all zPDT releases. These z/OS volume images are already in the emulated 3390 format. The data and customizations are stored on those volumes from the last time z Systems Development and Test Environment was run. The DEVMAP developed for running with that z/OS distribution on the older version also runs unchanged with the newer version. For more information about device maps, see “Defining the device map” on page 103.

You can also install or reinstall a previous z/OS ADCD distribution. Use the installation instructions that are supplied with the release of Rational Development and Test Environment that has entitlement to that previous z/OS ADCD. Beginning with the z/OS 2.1 ADCD for Rational Development and Test Environment for z Systems, access to a license key is required to install the system resident volumes of the z/OS ADCD by using the Z1091_ADCD_install command. Therefore, this command requires access to either an activated USB hardware device or an activated license manager.

Making the z/OS volume images available to z Systems Development and Test Environment

After all the z/OS volume images in your z/OS distribution are converted to the emulated 3390 format and decompressed, they must be made available to z Systems Development and Test Environment.

Put all of the images that compose your z/OS distribution in the \$HOME/z1090/disks directory. Then, enter these two commands that are run under the authority of ibmsys1:

```
cd /home/ibmsys1/z1090/disks      #change to destination directory
chmod 755 *
```

Installing z/VM

If you purchased z Systems Development and Test Environment with Parallel Sysplex®, you have entitlement to z/VM 6.3.

For the instructions to download and install z/VM 6.3, and enable the coupling facility, see Chapter 9, “Deploying z Systems Development and Test Environment for Parallel Sysplex and the coupling facility,” on page 183.

After you finish installing z Systems Development and Test Environment, one of the final steps is to run the **z1090instcheck** command. This command verifies the proper Linux 32-bit runtime libraries are installed.

Chapter 5. Enabling product operation

You need an activated license manager or activated USB hardware device to operate the product and authorize the licensee to use z Systems Development and Test Environment or any of its components.

The method of authorization you use depends on the version of the product that you have.

- For Rational Development and Test Environment for z Systems v9.5 and earlier, you must use an activated USB hardware device.
- For z Systems Development and Test Environment v10 and later, the method depends on the type of license entitlement:
 - For z Systems Development and Test Environment Personal Edition v10 (Authorized User Single Install), you must use an activated USB hardware device.
 - For Resource Value Unit (RVU) and Rational Token software-based license entitlements, you must use an activated license manager.

Product enablement checklist

Use these checklists to enable product operation with a license manager or a USB hardware device.

Software-based license

About this task

If you purchased z Systems Development and Test Environment with Resource Value Unit (RVU) or Rational Token entitlement, you must use software-based licensing. Use this checklist to activate the product by using software-based licensing. These steps link to other topics for more information.

Procedure

1. Install the z Systems Development and Test Environment license manager feature on the computer that is to act as the license manager. For step-by-step instructions, see “Installing and starting the license manager” on page 68.
2. Optional: Install the z Systems Development and Test Environment license manager feature on other computers for failover. For more information, see “License manager failover” on page 70.
3. Install z Systems Development and Test Environment on computers that meet the System Requirements and use software-based licensing. For detailed installation and configuration instructions, see Chapter 4, “Installing z System Development and Test Environment,” on page 37.
4. Obtain a license key file. For software-based licenses, the license key file is a license manager update file. Use one of these procedures for your license metric to activate your license manager:
 - “Activating a license manager” on page 71
 - “Activating a license manager with Rational Tokens” on page 73

5. If you purchased entitlement that uses Rational Tokens, you must set up the Rational License Key Server and obtain the corresponding license from the Rational License Key Center. See “Setting up Rational Tokens” on page 205.
6. If the license manager is not on the same subnet, configure each computer that is hosting the offering to use the license manager that you set up in step 1. See “Activating and configuring a license manager client” on page 77.
7. On the computers that host the mainframe applications that run with the emulator, create the device maps used to define your emulated systems:
 - If you purchased entitlement to z Systems Development and Test Environment with Parallel Sysplex, include a line to enable the coupling facility.
 - If you purchased entitlement that uses Rational Tokens, include a line to identify the Rational License Key Server to be used. As an alternative, you can set the RDTSERVER Linux environment variable.
8. To install z/OS on any host computers, follow the instructions in “Installing z/OS volume images” on page 55.

USB hardware device

About this task

If you purchased z Systems Development and Test Environment Authorized User Single Install, or any version before z Systems Development and Test Environment v10.0, use this checklist to activate the product by using one or more USB hardware devices. The process for activating the product is similar for all versions of z Systems Development and Test Environment before v10.0.

Procedure

1. Purchase the z Systems Development and Test Environment offering and license metric that best suits your needs.
2. Obtain one or more USB hardware devices by ordering the appropriate media pack through Passport Advantage.

Note: USB hardware devices are not tied to a particular release of the offering. If you have a USB hardware device from a different release, renew the activation to obtain the update file that is needed to install the z/OS® 2.1 or later Application Developers Controlled Distribution for z Systems Development and Test Environment.

3. Obtain an update file to activate the USB hardware device from the Rational License Key Center. If you purchased entitlement that uses Rational Tokens, you must set up the Rational License Key Server and obtain the corresponding license from the Rational License Key Center. See “Obtaining an update file” on page 87.
4. Optional: Set up a product license server:
 - a. Install z Systems Development and Test Environment on the license server.
 - b. Connect the USB hardware device to the license server.
 - c. Apply the update file to the USB hardware device attached to the license server.
 - d. Start the product license server.

For more information about setting up a product license server, see “Setting up the product license server” on page 97.

5. Install z Systems Development and Test Environment on computers that meet the System Requirements. For detailed installation and configuration instructions, see Chapter 4, “Installing z System Development and Test Environment,” on page 37.
 6. If you are using a product license server, configure each computer that is hosting the offering to use the product license server you set up in step 4. See “Setting up computers to access the product license server” on page 98.
 7. If you are not using a product license server, apply the corresponding update file to the locally attached USB hardware device of each computer that is hosting the offering. See “Activating the USB hardware device” on page 93.
 8. On the computers that host the mainframe applications that run with the emulator, create the device maps used to define your emulated systems:
 - If you purchased entitlement to z Systems Development and Test Environment with Parallel Sysplex, include a line to enable the coupling facility.
 - If you purchased entitlement that uses Rational Tokens, include a line to identify the Rational License Key Server to be used. As an alternative, you can set the RDTSERVER Linux environment variable.
- For more information about creating device maps, see “Defining the device map” on page 103.
9. To install z/OS on any host computers, follow the instructions in “Installing z/OS volume images” on page 55. You can now use the emulator.

Software-based licensing

When you use software-based licensing, a license manager provides a server for centralized management of license keys for one or more instances of z Systems Development and Test Environment. With this method, a USB hardware device is not required.

The z Systems Development and Test Environment license manager is installed on a machine, and the authenticated license key files are installed on the license manager. The licensee is not authorized to use z Systems Development and Test Environment or any of its components except when that license manager is activated with a license key file and is accessible by the program, as it provides proof of license entitlement.

The license manager and an instance that runs z Systems Development and Test Environment cannot run on the same platform, but it does need network connectivity to the license manager.

Note:

1. License manager needs to be a static resource in any infrastructure configuration. In a virtualized or cloud infrastructure, it does not tolerate physical moving. If the license manager is manually or automatically moved inside the infrastructure, it will stop working. If you would like to move the License Manager, you need to return the license before you move it, acquire the license after movement, and configure the License manager again.
2. Do not configure network by using special IP addresses as xxx.xxx.xxx.1 or xxx.xxx.xxx.255 for the License Manager machine; otherwise, communication cannot work properly from z Systems Development and Test Environment client instances to the License Manager instance.

Software-based licensing terminology

Understand the terms that are used to describe the activation process with software-based licensing.

The words *license* and *token* have more than one meaning, which depends on the context in which they are used.

You can encounter at least two different associated meanings for the word *license* as you work with z Systems Development and Test Environment:

- The word *license* can mean the product entitlement that you purchased and the corresponding terms and conditions under which use is allowed.
- The Rational License Key Center uses the word *license* and *licenses* for a quantity of product entitlement parts purchased.

To clarify the intended use of the term, the z Systems Development and Test Environment documentation uses these phrases:

- The phrase *product entitlement* refers to the terms and conditions under which the product can be run by the licensee.
- The phrase *license entitlement* refers to the Rational License Key Center interface and the quantity of entitlement parts purchased.
- The phrase *license key file* refers to a file that represents the user's license entitlement. License key files must be obtained, installed on the license manager, and available to operate z Systems Development and Test Environment.
- The phrase *license manager update file* refers to the specific license key file that is generated in the Rational License Key Center and installed on a license manager by using an **update_license** command to activate it. This file is used for software-based licensing.
- The phrase *license manager activation* refers to the condition of the license manager and its readiness for operation by clients that run the emulator.

You can encounter at least two different associated meanings for the phrase *license server* as you work with z Systems Development and Test Environment:

- The phrase *license server*, *software-based license server*, or *license manager* can mean the License Development Kit–Software License (LDK–SL) server that distributes software-based license keys without the need for a USB hardware device. This server is also referred to as a license manager.
- The same phrase *license server* can mean the Rational License Key Server that is used to share license entitlements among several IBM products.

To clarify these terms, the z Systems Development and Test Environment documentation uses these phrases:

- The phrase *license manager* refers to the remote z Systems Development and Test Environment that distributes authentication by using software-based license keys.
- The phrase *Rational License Key Server* refers to the license server used exclusively to share Rational Token license entitlements among a set of IBM products.

The word *token* is used to refer to a type of IBM Rational product entitlement that allows great flexibility in the deployment and use of associated, purchased products. The z Systems Development and Test Environment documentation uses the phrase *Rational Token* to refer to IBM Rational Token licensing.

The word *instance* refers to a single installation of z Systems Development and Test Environment on a physical or virtual machine.

z Systems Development and Test Environment can be purchased with several different product entitlements. The various types of entitlements authorize use of the product in specific ways.

For details on the product entitlement you purchased, see the documentation that is provided with your copy of the offering or refer to the Product License available at the IBM Software License Agreement website.

The Product License can be found by searching the site for the product name.

Overview of the activation process

When you use software-based licensing, z Systems Development and Test Environment requires access to a license manager that is activated with a software-based license key file to control the licensee's access to all or portions of the program. The license manager, sometimes referred to as a software-based license server, manages all software-based license key files. The licensee is not authorized to use z Systems Development and Test Environment or any of its components except when that license manager is activated with a license key file and is accessible by the program, as it provides proof of license entitlement. License key files are obtained from the Rational License Key Center, and are specific to a customer account.

Each license key file is generated with a number of entitled emulated central processors, referred to as *Central Processors* or *CPs*. A single instance of z Systems Development and Test Environment requires at least 1 CP, and can run with a maximum of eight CPs. The number of CPs needed depends on the number of users and the types and amount of processing required. For recommendations on how many CPs to use with an instance, see section 2.3.4 “Performance” and section 13.12, “CPs, processors, threads, and tokens” in the zPDT Guide and Reference.

Note: z Systems Development and Test Environment Authorized User Single Install, and all releases before v10 that do not include the Parallel Sysplex offering can run with a maximum of 3 CPs.

The software-based license key file that is generated by the Rational License Key Center is called a license manager update file. It is created by the Rational License Key Center for a specific license manager, and only that license manager can be activated with that license manager update file. A license manager is activated with only one license manager update file. To change the number of entitled emulated CPs on a license manager, or to renew a software-based license key that is currently installed on a license manager, you must first return the license key and then generate a new one.

A single activated license manager can authenticate many instances of z Systems Development and Test Environment. The number of instances that can be authenticated varies based on the number of emulated Central Processors or CPs used in each instance.

When an instance of the offering starts, it seeks an activation from the license manager for the number of CPs specified in the emulator's device mapping file. When an instance of the emulator is stopped, the CPs authenticated by the license

manager become available for another instance. The activated license manager must remain available to the instance while the instance is running.

Software-based license key files are generated and license managers are activated with that license key file by using a multi-step process:

1. Create a license manager request file on the license manager on which the license key is to be installed.
2. Obtain from the Rational License Key Center a license manager update file unique to that server to provide proof of license entitlement.
3. Activate the license manager by installing the license manager update file.
4. If the license manager is not on the same subnet as all z Systems Development and Test Environment instances that use that manager for their license entitlement, configure network access to the license manager from any of those instances not on the same subnet. Any instance that uses a license manager is known as a license manager client.

If you purchase entitlement to z Systems Development and Test Environment through Rational tokens and software-based licensing, the offering also communicates with a Rational License Key Server to share license entitlements among several IBM products. For more information about shared license entitlements, see “Rational Token activation” on page 67 and “Activating a license manager with Rational Tokens” on page 73.

Note: License manager needs to be a static resource in any infrastructure configuration. In a virtualized or cloud infrastructure, it does not tolerate physical moving. If the license manager is manually or automatically moved inside the infrastructure, it will stop working. If you would like to move the License Manager, you need to return the license before moving, acquire the license after movement, and configure the License manager again.

License key expiration

All license key files expire after one year from the date they were generated, or after less than a year if entitlement is from a term license that expires in less than a year. If your purchase entitles you to use the product for more than one year, you need to renew the license key each year. Renewal activates the license key for a subsequent year or until the end of the fixed term if less than a year.

When z Systems Development and Test Environment is running, you can check the expiration date of a license key file by using the **token** command. You can run the token command only while the emulator is running, and you must run the command from the same user ID that started the emulator.

License keys and the Application Developer Controlled Distribution (ADCD)

z/OS 1.13 ADCD and earlier distributions can be installed without access to an activated license manager. All volumes are supplied as files in compressed (.gzip) format, and the **gunzip** command makes them usable by z Systems Development and Test Environment.

For software-based entitlement, beginning with the z/OS 2.1 ADCD for Rational Development and Test Environment for z Systems, an activated license manager is required to install the system resident volumes of the z/OS ADCD. The **Z1091_ADCD_install** command is used to install these system resident volumes. This command requires access to an activated license manager with connectivity to

the instance that does the installation. The same activated license manager that entitles the user to run z Systems Development and Test Environment also entitles the user to install a z/OS ADCD.

For instructions on installing the z/OS ADCD, see “Installing May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 57

Rational Token activation

You can purchase entitlement to z Systems Development and Test Environment through Rational Tokens. When you use Rational Tokens, entitlement can be shared among several IBM products, with license entitlements tracked by temporary use of a number of Rational Tokens. For z Systems Development and Test Environment, each emulated central processor within an instance of a z Systems virtual machine requires a particular number of Rational Tokens, based on the token value for z Systems Development and Test Environment. When you use Rational Tokens, each instance of the emulator coordinates with a Rational License Key Server instance and starts or continues to run only if sufficient Rational Tokens are available for the number of configured CPs defined for that instance.

For software-based entitlement, use of Rational Tokens does not replace the requirement for an activated license manager to be available to z Systems Development and Test Environment. Any instance that uses Rational Tokens must have access to an activated license manager, and must have access to the Rational License Key Server. That license key file must indicate that use of Rational Tokens is in effect. When z Systems Development and Test Environment is started, the emulator verifies that an activated license manager is found and enough Rational Tokens are available from the Rational License Key Server to run the emulator. Rational Tokens control product operation while enough CPs are available from the license manager.

For information on setting up and using the Rational License Key Server, see the associated documentation available on the offering's installation e-image.

z Systems Development and Test Environment with Parallel Sysplex

z Systems Development and Test Environment with Parallel Sysplex, in addition to providing the base environment that supports z/OS also includes IBM z Personal Development Tool support for virtual Coupling Facility and the z/VM software distribution to enable virtual coupling of z/OS guests under z/VM. Entitlements with this offering that enable the coupling facility are referred to in the product documentation as Sysplex activations. The Parallel Sysplex offering is available only with Resource Value Unit (RVU) or Rational Token license entitlements.

A single instance of z Systems Development and Test Environment with Parallel Sysplex enabled runs z/VM, the coupling facility, and one or more instances of z/OS. The available CPs defined to the instance are shared among all systems that are running, and are defined in the *processor* setting of the *devmap*. Typically, more CPs are required to run a Parallel Sysplex offering. While in principle you can use one CP on a base PC with one processor, or *core*, to run a z/VM system with several Parallel Sysplex z/OS guests, this configuration is not practical and might result in various z/OS timeouts. Configure at least one CP for each z/OS guest on the z/VM system. A single instance of z Systems Development and Test Environment with Parallel Sysplex can be configured to use a maximum of 8 CPs. Chapter 21, “Sysplex” in the zPDT Guide and Reference and zPDT 2016 Sysplex Extensions give more information about Parallel Sysplex systems.

Although a coupling facility is not related to an individual processor in real z Systems hardware, every CP in a z Systems Development and Test Environment with Parallel Sysplex instance requires entitlement to a Parallel Sysplex CP activation, or RVU, to enable the coupling facility for the instance.

The license manager

Learn how to install, configure, and administer license managers.

Installing and starting the license manager

To use software-based licensing, you must install at least one instance of the z Systems Development and Test Environment license manager that manages all software-based license keys.

About this task

For the hardware requirements, software requirements, and steps to install a license manager, see Chapter 4, “Installing z System Development and Test Environment,” on page 37.

The installation process installs both the license manager function and the UIM server function, and starts the license manager if the appropriate glibc library is installed. The license manager is started as a service named aksusbd.

Procedure

- When you use software-based licensing, the UIM server must run on the license manager, and unlike the license manager daemon is not automatically started at installation. Start the UIM server from a non-root user ID by using this command:

uimserverstart

- **Firewall considerations:** The license manager listens for license key requests on port 1947. You must manage any firewalls that are involved with this server. When you initially install in a test environment, it is easier to diagnose errors if you are able to disable firewalls until you are satisfied with your software-based license operations. This action separates network-related problems from license manager issues. If you operate through firewalls, you must ensure that the relevant port numbers can pass through the firewalls. You can use many techniques to manage firewalls, depending on what product is being used.

Capability configuration and capacity of the license manager

Specific combinations of Parallel Sysplex support and use of Rational Tokens define the capability configuration of the license key and of the license manager. The Rational License Key Center does not combine different types of software-based license orders when you generate a software-based license key file. Any orders that are combined to create one license key file must be of the same license type. However, the number of CPs that you can generate for a software-based license key file is limited only by your number of entitlements. You can generate a license key file with any or all of your entitlements for the same capability configuration.

You can have only one installed license key file on a license manager. That license key file allows operating within only one configuration:

- Parallel Sysplex support is not enabled and you are not required to use Rational Tokens.

- Parallel Sysplex support is enabled and you are not required to use Rational Tokens.
- Parallel Sysplex support is not enabled and Rational Tokens are required.
- Parallel Sysplex support is enabled and Rational Tokens are required.

For other steps required to run if you are using Rational Tokens for your entitlement, see “Setting up Rational Tokens” on page 205.

To ensure that only one installed license key file is generated for a license manager, a license key file must be returned to the Rational License Key Center before it can be renewed or another file can be generated for the same server. For more information, see “Returning a software-based license key less than 31 days before expiration” on page 74.

Host ID and host name of the license manager

Each license key file is uniquely identified in the Rational License Key Center with the host name and host ID of the license manager for which the license was generated. For software-based licenses, the host name is the host name of the server, which can be displayed with the Linux **hostname** command. If the host name has periods, such as if it is an IPv4 address, the Rational License Key Center replaces the periods in the host name with underscore characters.

The host ID, which is required to be unique across all license key files in the RLKC, is a generated unique identifier for each license manager. The host ID is generated when the license key file is generated. If you are trying to find the license key file in the RLKC, by using either **Return Keys**, **View Keys by Host**, or **View Keys by User**, the host ID of the server is the field that uniquely correlates a license key file to the license manager for which it was generated.

Administering the license manager

The license manager for z Systems Development and Test Environment is controlled by line commands from a Linux command line on the server. The commands that are needed to administer the server are described in the topics on obtaining, returning, and renewing software-based license key files.

You can display information about the license key that is currently installed on the license manager from the root userid by entering this command:

```
/opt/IBM/LDK/query_license
```

Stopping and uninstalling the license manager

If you need to uninstall the license manager, either to move license managers or to discontinue use of a license manager, you must first return the activated license key.

Before you begin

For instructions for returning an active license key, see “Returning a software-based license key less than 31 days before expiration” on page 74

Procedure

1. To stop the license manager, enter **systemctl stop aksusbd.service**.
2. From a non-root user ID, enter **uimserverstop**.
3. Go to the directory where installation program is present and switch to root user id to run below program.
4. Execute the command **./zDT_Install.x86_64**.

5. Type **2** to select IBM z Systems Development and Test Environment License Manager.
6. Type **3** to uninstall the selected product.
7. Wait for the process to get finish.
8. To uninstall silently, execute the command `./zDT_Install.x86_64 --uninstall --zdtlicense`.

License manager failover

You can activate multiple license managers of the same capability configuration, and have one or more available for failover. Each instance of z Systems Development and Test Environment that uses software-based licensing can configure a list of managers. When an instance configures access to multiple managers, if both managers are active, entitlement is requested evenly between the managers. If either manager does not respond or has insufficient entitlement, either during initial activation or during product operation, the client instance attempts to authenticate entitlement through the other active managers in the list.

About this task

For more information about the steps to access a license manager, see “Activating and configuring a license manager client” on page 77.

Important: Since license key files are unique to each license manager, you cannot use one license key file on multiple license managers. Each license manager, including the ones that are used for failover, must have its own entitlements. For more information about configuring license managers, see “Planning for license managers.”

Changing license managers

If you need to move your license manager to a different platform, you must first return your license keys from the current license manager, and then generate and install license key files for each new license manager. If the original platform that hosts the license manager will not be used as a license manager in the future, you can optionally uninstall the z Systems Development and Test Environment license manager.

Do these steps:

1. Follow the steps in “Stopping and uninstalling the license manager” on page 69. Uninstalling the license manager is optional. If your current server is unusable, contact the Rational License Key Center to return your license key file.
2. For each new license manager, install and start the license manager by using the steps that are described in “Installing and starting the license manager” on page 68.
3. Generate a new license key file for each new license manager by using the steps that are described in “Activating a license manager” on page 71 or “Activating a license manager with Rational Tokens” on page 73.

Planning for license managers

You can spread your license entitlements across one or more license managers. Also, you can choose to have a license manager always active for failover if a failure on the primary license manager occurs. How many license managers are needed, and how those entitlements are activated on each license manager, is dependent upon many factors, such as

- How many instances you run,
- The access of the different users to the license managers,
- The types of license entitlements you use, and
- Having license managers for failover or not.

Consider the following rules when you are planning how to spread your license entitlements across your license managers, and how to later generate your license manager update files:

- Each license manager can have only one installed license key file at a time. That license key file can be of only one capability configuration. Therefore, you must have at least as many license managers as capability configurations you need. For example, if you have entitlement to z Systems Development and Test Environment and z Systems Development and Test Environment with Parallel Sysplex, you need at least two license managers installed.
- The number of CPs on a software-based license key is only limited by your entitlement. Therefore, if all license manager clients have access to a license manager for the required capability configuration, one license manager and one license manager for failover is sufficient.
- The license key file for all active license managers must have at least as many CPs as the combined number of processors that are needed for all concurrent instances that use the license manager. This number can be calculated by adding the number of processors that are defined in the devmap for each instance that can run concurrently.
- If you require a license manager to be available if failover occurs, have one for each capability configuration that is allowed by your entitlements.
- How many CPs to entitle each license manager when multiple license managers are activated to allow for failover can vary. If each license manager has sufficient CPs for the maximum number of concurrent users, operations will continue in the event of failure. You can spread your required entitlements evenly across license managers if you can limit users during the time one license manager fails. If a failure is not temporary, you can also generate a new license key file with more entitled CPs for one or more of the active license managers.
- You can combine orders (license entitlements) in one update file, but only if they are of the same license type

Note: The hostname of the server that is hosting an active license manager must not change while an active license key is installed. In order to change the hostname of the server, the license key must first be returned to the Rational License Key Center, then the server can be renamed, and, if you want to, you can request a new license key for the modified server.

Activating a license manager

To obtain a software-based license key file, you must create a license manager request file from the license manager where the license key file is to be installed. Then, you must go to the Rational License Key Center and install license key file on the license manager to activate it.

If you already activated a license manager with a license key file, and need to generate a new license key file either because the license key is expiring or you need to change the license entitlements that are applied to the license key, you must first return the existing license entitlement in the Rational License Key Center. For more information about returning a license key, see “Returning a software-based license key less than 31 days before expiration” on page 74 or

“Returning a software-based license key more than 30 days before expiration” on page 75. After you return the license key, the procedures to generate a new one are the same as obtaining an initial license key file.

Procedure

After you install and start the license manager, do these steps to activate the license manager:

1. From the root user ID on the license manager, run this command:
opt/IBM/LDK/request_license. This command creates a request compressed file in the root home directory with a name similar to `Hostname_xxxxxxxx.zip`. This file is unique to the license manager, and is used to generate the software-based license key file.
2. If needed, transmit the `Hostname_xxxxxxxx.zip` file to the system you use to log in to the Rational License Key Center.
3. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
4. On the left side of the screen, select **Get Keys**.
5. Select the product line for z Systems Development and Test Environment. The interface displays a list of license key names that show the z Systems Development and Test Environment licenses that are available.
6. Select the z Systems Development and Test Environment license type that you want to apply to the key: IBM z Systems Development and Test Environment Resource Value Unit or IBM z Systems Development and Test Environment with Parallel Sysplex Resource Value Unit.
7. Select the check box next to all of the orders from which you would like to use licenses and click **Next**. The interface displays a table for you to complete.
8. Provide the license manager request file for the license manager for which the license key file is to be generated and the number of Emulated Central Processors you intend to enable on the manager. The total number of emulated CPs must not exceed your total number of Resource Value Unit (RVU) entitlements. For help with the table, click a column heading.
9. Select **Generate** at the bottom of the page. A window opens, prompting you to download the license manager update file.
10. To save the file, click **Download**. You can download the generated file either during this process or later. To download any files later, use the View Keys by Host link on the left side of the Rational License Key Center page.
11. Transmit the license manager update file to the license manager.
12. From the root user ID on the license manager, run this command:
/opt/IBM/LDK/update_license Hostname_xxxxxxxx_update.zip In this command, you must specify the absolute path to the `Hostname_xxxx_update.zip` file. This command installs the license key file on the server.
13. In the license manager, enter the command `./serverconfig` from path `/opt/IBM/LDK`. Then set up the following options:
 - a. Set **Server Access** to 'Y'. If this field is set to 'N', client or instance can not get the license from license manager.
 - b. Set **Enable Access Log** to 'Y'. If you would like to create log, enter the command `./display_gen2_acclog` in the path `/opt/IBM/LDK` once zOS is ipled. Then the log is displayed.

Alternatively, you can use `serverconfig` command via command line interface by entering command `/opt/IBM/LDK/./serverconfig_cli <parameter>`.

To check the list of possible parameter values, enter command `/serverconfig_cli` without parameter.

14. Optional: If you would like to deny or allow access to some instances or client machines, edit the file `/opt/IBM/LDK/rules.ini`, and execute the command `./serverconfig_cli -u` to load rules. To check all rules currently in use, enter command `./serverconfig_cli -r`.

Note: You cannot generate separate license key files for the same license manager at the same time. Before you generate a license key file you want to change for a particular manager, you must return the license key file that was previously created for that server. Returning license entitlements that are assigned to a manager also makes the corresponding active entitlements available for assignment to another license manager.

Activating a license manager with Rational Tokens

Since software-based license key files are unique to a license manager, to obtain a software-based license key file you must create a license manager request file from the license manager where the license key file is to be installed. Then, you must go to the Rational® License Key Center, use the license manager update file to generate a license key file for the license manager, and separately generate a Rational Token license file. Then, you must install the license key file on the license manager to activate it.

Note: You cannot generate separate license key files for the same license manager at the same time. Before you generate a license key file you want to change for a particular manager, you must return the license key that was previously created for that server. Returning license entitlements that are assigned to a manager also makes the corresponding active entitlements available for assignment to another license manager.

After you install and start the license manager, do these steps to activate it:

1. From the root user ID on the license manager, run this command:
opt/IBM/LDK/request_license. This command creates a request compressed file in the root home directory with a name similar to `Hostname_XXXXXXXXX.zip`. This file is unique to the license manager, and is used to generate the software-based license key file.
2. If needed, transmit the `Hostname_XXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.
3. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
4. On the left side of the screen, select **Get Keys**.
5. Select the product line for IBM Rational Tokens.
6. The Select License Keys page displays one or more boxes for orders. Find the order that includes z Systems Development and Test Environment. Make a note of the part number for the z Systems Development and Test Environment product you want to activate. Select the order that contains the parts you noted and click **Next**.
7. Enter the required information to create the Rational Token license file for your host. Additional information for each field is available by clicking the descriptive link next to the input field. Fields with a red star are required fields.
8. Click **Generate** to generate the Rational Token license file. You are presented with a screen with the generated license information.

9. Click **Download License Keys** to download the Rational Token license file to your computer. This token license file is the file that is applied to your Rational License Key Server. It is named `license.dat` by default. Additional information on setting up the Rational License Key Server and applying this file to the server can be found in the Rational License Key Server documentation that is provided in the Rational License Key Server media.
10. On the Select License Keys page, click **Generate Token License Keys for Software-based License Servers** to generate and download the license manager update file for your license manager.
11. A Required Information page opens and displays a table that you must complete. Provide the license manager request file for the license manager for which the license key file is to be generated. Use the **Catalog Item** list to confirm the part that you want to use to generate an update file. This confirmation is important if you have entitlement to both z Systems Development and Test Environment and z Systems Development and Test Environment with Parallel Sysplex. You cannot combine Token and Sysplex Token activations on a single software-based license key file. From the list, select the entitlement part number that corresponds to the activation you want on the specified license manager update file. For help with the table, click a column heading.
12. Select **Generate** at the bottom of the page. A window opens, prompting you to download the license manager update file.
13. To save the file, click **Download** for each file that was generated. You can download the generated file either during this process or later. To download any files later, use the **View Keys by Host** link on the left side of the Rational License Key Center page.
14. Transmit the license manager update file to the license manager.
15. From the root user ID on the license manager, run this command:
`/opt/IBM/LDK/update_license Hostname_XXXXXXXXXX_update.zip`. In this command, you must specify the absolute path to the `Hostname_XXXX_update.zip` file. This command installs the license key file on the server.
16. Restart the license server daemon to make the license key file active by entering `systemctl restart aksusbd.service`.

Related information:

“Setting up Rational Tokens” on page 205

Rational Token licensing is an entitlement that you can purchase and use to run z Systems Development and Test Environment. With Rational Tokens, z Systems Development and Test Environment maintains a connection to a Rational License Key Server and starts and continues to run only when sufficient Rational Tokens are available.

Returning a software-based license key less than 31 days before expiration

To generate a new license key file for a license manager, you must first return the existing license entitlement in the Rational License Key Center, and then generate a new one. Returning the license entitlements is a process of telling the Rational License Key Center that you are no longer using the license entitlements that you previously assigned to the license manager. You do not need to return the physical license manager update file to the Rational License Key Center.

You can return a license entitlement in the Rational License Key Center in several ways. The easiest method is to use the **View Keys by Host** link. That process is described here. You can also use the **Return Keys** link.

Do these steps to return a software-based license key that has less than 31 days until expiration.

1. From the root user ID on the license manager, run this command:
opt/IBM/LDK/request_license. This command creates a request compressed file in the root home directory with a name similar to `Hostname_XXXXXXXXXX.zip`. This file is unique to the license manager, and is used to generate the software-based license key file.
2. If needed, transmit the `Hostname_XXXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.
3. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
4. On the left side of the screen, select **View Keys by Host**.
5. Select the **Hostname** corresponding to the license key you want to return.
6. A table is displayed with data for the **Hostname** selected. At the far right of the table, click the **Change** link.
7. The interface displays a list of the license keys for this license entitlement. Locate the license key of the license manager you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXXX.zip` file that you just created.
8. Click **Initiate Return**.
9. A message is displayed to confirm that the license entitlements were successfully returned.

Returning a software-based license key more than 30 days before expiration

To generate a new license key file for a license manager, you must first return the existing license entitlement in the Rational License Key Center, and then generate a new one. Returning the license entitlements is a process of telling the Rational License Key Center that you are no longer using the license entitlements that you previously assigned to the license manager. You do not need to return the physical license manager update file to the Rational License Key Center.

To return a software-based license key that has more than 30 days until expiration requires that you invalidate the license on the license manager as part of the return process. This process consists of these general steps:

- Create a license manager request file from the license manager where the license key file is to be installed.
- Obtain a license key file that invalidates the current license key from the Rational® License Key Center.
- Install the license key file that invalidates the current license key on the license manager.
- Use the update file to complete the return process in the Rational License Key Center.

You can return a license entitlement in the Rational License Key Center in several ways. The easiest method is to use the **View Keys by Host** link. That process is described here. You can also use the **Return Keys** link.

Do these steps to return a software-based license key that has more than 30 days until expiration.

1. Create a license manager request file from the license manager where the license key file is to be installed:
 - a. From the root user ID on the license manager, run this command:
opt/IBM/LDK/request_license. This command creates a request compressed file in the root home directory with a name similar to `Hostname_XXXXXXXXX.zip`. This file is unique to the license manager, and is used to generate the software-based license key file.
 - b. If needed, transmit the `Hostname_XXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.
2. Obtain a license key file that invalidates the current license key from the Rational® License Key Center.
 - a. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
 - b. On the left side of the screen, select **View Keys by Host**.
 - c. Select the **Hostname** corresponding to the license key you want to return.
 - d. A table is displayed with data for the **Hostname** selected. At the far right of the table, click the **Change** link.
 - e. The interface displays a list of license keys for this license entitlement. Locate the license key of the license manager you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXX.zip` file that you just created.
 - f. Click **Initiate Return**.
 - g. A window opens, prompting you to download the license manager update file with the file to invalidate the license on the license manager.
 - h. To save the file, click **Download**. You can download the generated file either during this process or later. To download any files later, use the **View Keys by Host** link on the left side of the Rational License Key Center page.
 - i. Transmit the license manager update file that invalidates the license key to the license manager.
3. Install the license key file that invalidates the current license key on the license manager.
 - a. From the root user ID on the license manager, unzip the `Hostname_XXXXXXXXX_update.zip` file, run this command:
/opt/IBM/LDK/update_license Hostname_XXXXXXXXX_update.v2c. This command installs the invalidating license key file on the server.
 - b. Restart the license server daemon to make the license key file active by entering `systemctl restart aksusbd.service`.

The license manager is no longer active, and the licenses cannot be used anymore. However, the license entitlements in the Rational License Key Center are still reserved until the return process is completed.
 - c. From the root user ID on the license manager, run this command:
opt/IBM/LDK/request_license. This command creates a request compressed file in the root home directory with a name similar to `Hostname_XXXXXXXXX.zip`. This file verifies that the license key is invalidated and is used to complete the return of the software-based license key file.
 - d. If needed, transmit the `Hostname_XXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.

4. Use the verification file to complete the return process in the Rational License Key Center.
 - a. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
 - b. On the left side of the screen, select **View Keys by Host**.
 - c. Select the **Hostname** corresponding to the host name of the license manager from which you want to return the license key.
 - d. A table is displayed with data for the **Hostname** selected. At the far right of the table, click the **Change** link.
 - e. The interface displays a list of the license keys for this license entitlement. Locate the license key of the license manager you are returning. Click **Browse**, and browse to and select the `Hostname_xxxxxxxx.zip` file that you just created.
 - f. Click **Complete Return**.
 - g. A message is displayed to confirm that the license entitlements were successfully returned.

Activating and configuring a license manager client

Each instance that uses software-based licensing must activate the license manager client and configure the client to have network access to the license manager. You must have network connectivity to the server.

About this task

If the license manager client is activated on an instance of z Systems Development and Test Environment, the zPDT emulator always first seeks access and entitlement from a license manager. For more information, see “zPDT license servers” on page 208.

Procedure

To configure and activate a license manager client:

1. Working as root, enter this command: `# /usr/z1090/bin/gen2_init`. This setup command needs to be done only once. It performs several actions:
 - a. The license manager client requires a 32-bit version of the Linux glibc library. This command first accesses the configured Linux repositories to obtain the latest version of this library. If your base Linux already has glibc-32bit installed, then you can ignore failures in attempting to fetch these libraries. If glibc-32bit is not already installed on your base Linux system and if you cannot connect to these repositories because of firewalls, for example, then you must obtain and install glibc-32bit in some other way before you start the license manager client.
 - b. The license manager client is started. The license manager client is started automatically any time the client Linux system is restarted.
2. If the client instance is running on a platform that previously authenticated with either a USB hardware device or a different license manager, issue this command from root:


```
uimreset -l
```
3. Enter this command from root:


```
cd /usr/z1090/bin/
./clientconfig
```

A window opens with several fields, then complete these fields:

- a. Set Gen2 Contact Server field to the hostname or IP address of license manager.
 - b. Set Gen2 BackupServer field if you want to set up license manager for failover. Otherwise, leave the field empty. This field is optional.
 - c. Leave Gen1 Contact Server empty. This field is used for product license server.
 - d. Leave Gen1 BackupServer empty. This field is used for failover product license server.
 - e. Set UIM ContactServer if you want to use different UIM server than License manager. Otherwise, leave the field empty. This field is optional.
 - f. Leave UIM Local Serial Random empty. This field is optional.
 - g. Press Enter twice to save these values.
4. Optionally, you can use command line interface to complete client configuration. To invoke the client, enter the following command:

```
cd /usr/z1090/bin/  
./clientconfig_cli <parameter>
```

You can write the client configuration file `sntlconfig.xml` for z1090/z1091. `clientconfig_cli` is usually run by the z1090/z1091 administrator as a superuser root.

The following parameters are accepted.

-[g1s1 | g1s2 | g2s1 | g2s2] servername

Specify Gen1 or Gen2 server. One contact server (s1) and one backup server (s2) can be specified.

-ucs server

Specify UIM contact server. The setup is the same as license server by default.

-usm y/n

Specify UIM serial method.

-l List servers currently configured.

-h Help, and print this message.

Results

The instance can be started. It attempts to authenticate with an activated license manager.

The USB hardware device

For v9.5 or earlier, z Systems Development and Test Environment requires a USB hardware device that contains a license key that controls the licensee's access to all or portions of the program. The USB hardware device is ordered through Passport Advantage in a media pack that is separate from the electronic media that contains the offering software. It can be connected directly to the USB port on a computer that is hosting the offering or it can be plugged into the USB port of a separate computer that hosts the offering for distributing authentication to remote instances of the product through a TCP/IP network. This computer is known as a product license server.

The licensee is not authorized to use z Systems Development and Test Environment or any of its components except when that key is made available to

the program. To make the key available to the program, you must obtain a license key file that provides proof of license entitlement from the Rational License Key Center. That file, called an *update file*, must then be applied to the USB hardware device to activate it and allow the licensee to use z Systems Development and Test Environment. The update file is specific to a customer account. The update file works with a single hardware device, which is identified by its unique serial number. The USB hardware device is installed either on the computer that hosts the offering or on a remote product license server that communicates over the network to one or more instances of the offering.

A single high-capacity USB hardware device can authenticate many instances of z Systems Development and Test Environment. The number of instances that can be authenticated varies based on the number of emulated Central Processors or CPs used in each instance.

When an instance of the offering starts, it seeks an activation for the number of CPs specified in the emulator's device mapping file. When an instance of the emulator is stopped, the CPs authenticated by the USB hardware device become available for another instance.

If you purchase entitlement to z Systems Development and Test Environment by using Rational Tokens, the offering can also communicate with a Rational License Key Server to share license entitlements among several IBM products. The license entitlements are tracked by temporary use of a number of Rational Tokens. For information on setting up and using the Rational License Key Server, see the associated documentation available on the offering's installation e-image.

Note: Both the USB hardware device and the Rational License Key Server are required for the product to operate with Rational Tokens. When Rational Tokens are used for authentication with USB hardware devices, implement a product license server. For more information about shared license entitlements, see Rational Token activation and "Setting up Rational Tokens" on page 205.

USB hardware device licensing terminology

Understand the terms that are used to describe the activation process with a USB hardware device.

The words *license* and *token* have more than one meaning, which depends on the context in which they are used.

You can encounter at least four different associated meanings for the word *license* as you work with z Systems Development and Test Environment:

- The word *license* can mean the product entitlement that you purchased and the corresponding terms and conditions under which use is allowed.
- The same word *license* can also refer to a technical indicator on the USB hardware device that is used to activate the hardware device and allow an emulated z Systems instance to operate. The zPDT Guide and Reference and the z Systems emulator messages use the words *license* and *licenses* to refer to these indicators stored on the hardware device. You can use a *remote license server* with the offering that derives its name from this particular usage of the word.
- The Rational License Key Center uses the word *license* and *licenses* for a quantity of product entitlement parts purchased.

- The file that is generated in the Rational License Key Center and that is applied to the USB hardware device to enable z Systems Development and Test Environment operation is called a generated *license* file in the website user interface.

To clarify the intended use of the term, the z Systems Development and Test Environment documentation uses these phrases:

- The phrase *product entitlement* refers to the terms and conditions under which the product can be run by the licensee.
- The phrase *license entitlement* refers to the Rational License Key Center interface and the quantity of entitlement parts purchased.
- The phrase *license key file* refers to a file that represents the user's license entitlement. License key files must be obtained, installed, and available to operate z Systems Development and Test Environment.
- The phrase *USB hardware device activation* or *activation* refers to the condition of the hardware device and its readiness for operation with the emulator.
- The phrase *update file* refers to the specific license key file that is generated in the Rational License Key Center and applied to the USB hardware device to activate it.

You can encounter at least two different associated meanings for the phrase *license server* as you work with z Systems Development and Test Environment:

- The phrase *license server* or *product license server* can mean the remote server that is set up to distribute authentication from a single high-capacity USB hardware device to remote instances of the product as an alternative to attaching discrete USB hardware devices to each computer that hosts one or more instances of the product.
- The same phrase *license server* can mean the Rational License Key Server that is used to share license entitlements among several IBM products.

To clarify these terms, the z Systems Development and Test Environment documentation uses these phrases:

- The phrase *product license server* refers to the remote z Systems Development and Test Environment license server that distributes authentication from a Sentinel Hardware Key USB hardware device.
- The phrase *Rational License Key Server* refers to the license server used exclusively to share Rational Token license entitlements among a set of IBM products.

You can encounter at least two different associated meanings for the word *token* as you work with z Systems Development and Test Environment:

- The word *token* can mean the USB hardware device that is required by z Systems Development and Test Environment. The zPDT Guide and Reference and z System emulator messages use the words *token* and *tokens* to refer to these hardware devices. In such places, you can find further reference to the 1091 token that is intended for use with the z Systems Development and Test Environment offering.
- The same word *token* can mean a type of IBM Rational product entitlement that allows great flexibility in the deployment and use of associated, purchased products.

To clarify these terms, the z Systems Development and Test Environment documentation uses these phrases

- The phrase *USB hardware device* refers to the hardware part.
- The phrase *Rational Token* refers to IBM Rational Token licensing.

The word *instance* refers to a single installation of z Systems Development and Test Environment on a physical or virtual machine.

z Systems Development and Test Environment can be purchased with several different product entitlements. The various types of entitlements authorize use of the product in specific ways.

For details on the product entitlement you purchased, see the documentation that is provided with your copy of the offering or refer to the Product License available at the IBM Software License Agreement website.

The Product License can be found by searching the site for the product name.

State of the hardware device

The USB hardware device is not activated by default. When activated, the device allows z Systems Development and Test Environment to run one or more emulated central processors, referred to as Central Processors or CPs.

Activations are placed on the USB hardware device by using a command to apply an update file to the hardware device. For instructions on obtaining update files and applying the files to your USB hardware devices, see sections “Obtaining an update file” on page 87 and “Activating the USB hardware device” on page 93.

The number of CPs needed on the update file is dependent upon the number of z Systems Development and Test Environment instances that use the USB hardware device, and how many CPs each instance uses. A single instance of z Systems Development and Test Environment without Parallel Sysplex enabled requires 1 CP, and can run with a maximum of 8 CPs. The configuration file for each instance, called a device map or *devmap*, has a processor setting that defines the number of CPs that are used by that instance. The appropriate number that is needed for one instance depends on the number of cores in the hardware on which the instance is run, and the z/OS configuration that is running on that instance. See “zPDT instances” in the most current edition of the zPDT Guide and Reference for guidance on the number of CPs to define per instance. For the CP requirements for a z Systems Development and Test Environment instance with Parallel Sysplex enabled, see “Parallel Sysplex Activations” on page 85.

Capacity of the hardware device

While every release of the offering included a USB hardware device, the maximum number of CPs that can be activated by a single device varied over time. The two variations that are issued are high capacity devices and low capacity devices.

High capacity devices

High capacity USB hardware devices can activate up to 100 CPs. These parts have a green label. Since an instance is usually 3 CPs, a high capacity device can typically enable up to 33 product instances of the offering. High capacity devices became available for order with Rational Development Unit Test v8.0.3 and subsequent releases. As of Rational Development and Test Environment for z Systems v8.5, only high capacity devices are packaged with the offering.

Low capacity devices

Low Capacity USB hardware devices can activate up to three CPs. These devices have a blue label. As of IBM z Systems Development and Test Environment v8.5, low capacity devices are no longer available for order.

If you are using a low capacity USB hardware device, you can successfully apply an update file that activates more than 3 CPs. However, you can use only three CPs due to the device limit. These hardware devices are intended for direct installation on the computer that hosts the offering. Only one instance of the offering is expected to run on the hosting machine.

Serial number of the USB hardware device

Each USB hardware device has a unique serial number that is required during the activation process. To locate the serial number of a key, turn the key to the side opposite the colored label. You see three rows of numbers. The lowest or bottom row of numbers is the serial number. The serial number is always of the form 03-xxxxx or 02-xxxxx where xxxxx is five hexadecimal digits.

Expiration dates

USB hardware device activations expire after one year from the date they were generated or less if entitlement is from a term license that expires in less than a year. If your purchase entitles you to use the product for more than one year, you need to renew the USB hardware device activation each year. Renewal activates the USB hardware device for a subsequent year or until the end of the fixed term if less than a year as applicable.

You can check the expiration date of an activation by using the token command. You can run the token command only while the emulator is running, and you must run the command from the same user ID that started the emulator.

Expiration notification example

USB hardware device activations last for up to one year. If you are entitled to use z Systems Development and Test Environment beyond that period, the USB hardware device must be reactivated by obtaining and applying a new update file.

The examples in these topics show how you can set up a system to automatically send you an email each day when the expiration date for your current activation is approaching. The example is a bash shell script that is run on a Linux computer that hosts an instance of z Systems Development and Test Environment. Do not run this script on your product license server because it requires the product to run on the same system. You cannot operate a product license server and the product simultaneously on the same system.

This command works only with a single USB hardware device. If your product license server has multiple USB hardware devices that are attached, the results of this example are unpredictable because the example uses only the device activation that is used by the product. It does not search for or examine all attached USB hardware devices.

This example uses the token command to determine the expiration date of the USB hardware device activation. If the expiration date is within a specified number of days, then it formats an email and sends that email to the specified user.

Related information:

“Expiration dates” on page 82

USB hardware device activations expire after one year from the date they were generated or less if entitlement is from a term license that expires in less than a year. If your purchase entitles you to use the product for more than one year, you need to renew the USB hardware device activation each year. Renewal activates the USB hardware device for a subsequent year or until the end of the fixed term if less than a year as applicable.

Sample Expiration Notification Script Setup

1. This example assumes that the mail command on your Linux system already works with your network. Before you try this example, verify that mail can be sent from your Linux system with the commands:

```
echo Testing>/tmp/mymail mail s "a test subject line"
userid@your.email.address
```

2. Copy the “Sample Expiration Notification Script” on page 84 to your computer and make it available to the user ID that runs it. For example, if ibmsys1 is running the emulator, you can copy this script to /home/ibmsys1/expiredTokenCheck.sh

3. Set the executable flag

```
chmod 755 /home/ibmsys1/expiredTokenCheck.sh
```

4. Set the EMAILADDRESS and THRESHOLD variables in the script to contain the email address to be notified and the number of days before expiration that notification is to begin.
5. Ensure PATH and LD_LIBRARY_PATH variable exports are set correctly.

Note: Changing the variable exports is probably not necessary.

6. Ensure that awsstart is running continuously on the computer on which the script runs. The script might be modified to start a small instance temporarily if awsstart is not running, but that would require that a license be available.
7. Create a cron job to run this command each day. For example, under the user ID that is running the emulator and this script, enter the command

```
crontab e
```

Next, add a line to your cron file that points to your script. For example, if this script is named expiredTokenCheck.sh and is to be run by user ibmsys1, then your crontab entry would include:

```
@daily /home/ibmsys1/expiredTokenCheck.sh
```

Sample Expiration Notification Script Output

This example, as shown, would send a note to USERID@YOUR.EMAIL.ADDRESS with a subject line similar to

```
RDT 0xDF0 expiration warning: 27 days (3/20/2014)
```

The body of the note is similar to the information shown in Table 3 on page 84.

Table 3. Output of the sample expiration notification script

```

Hostname ..... : rdtesys1
DNS Name ..... : rdtesys1.example.dnsname

Today's Date: ..... : 02/20/2015 (mm/dd/yyyy)
Expiration date .... : 3/20/2015
Days remaining ..... : 27 Days

**** USB Serial 0xDF0 will expire within 30 days. (27 remaining)
**** A new license file is required by

CPU 0, zPDTA(1091) available and working. Serial=4380(0x111C)
    Lic=3568(0xDF0) EXP=3/20/2015
CPU 1, zPDTA(1091) available and working. Serial=4380(0x111C)
    Lic=3568(0xDF0) EXP=3/20/2015
CPU 2, zPDTA(1091) available and working. Serial=4380(0x111C)
    Lic=3568(0xDF0) EXP=3/20/2015

End of zPDTA Status display

```

Sample Expiration Notification Script

```

#!/bin/bash
#
# Send an e-mail when a RDT USB hardware key activation is about to expire.
# This works with Rational Development and Test Environment only.
# For previous releases (RDzUT), the EXPDATE and SERIAL regular expressions
# would need to be modified because the format of the "token" command changed.
#
# This script should not be run on the product license server.
# This script runs on the machine running the emulator
#
# This is a sample only and may need modification on your system.
#
# If you would prefer notification other than email, you might want to
# - FTP a file to your z/OS system through the tunnel address.
# - Run a z/OS console command through "oprmsg".
# - Take other actions appropriate for your installation.

EMAILADDRESS=USERID@YOUR.EMAIL.ADDRESS # email address to receive emails
THRESHOLD=30 # number of days before expiration to start notification

export PATH=/usr/z1090/bin:$PATH # path to token command
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH # path to libman.so

SERIAL=`token |grep 'EXP='|head -n 1 | sed -e "s/.*Lic=.*(//" -e "s/).*//"`

if [ ! -z $SERIAL ]; then # if serial number was found

    EXPDT=$(token | grep "EXP="|head -n 1 | sed -e 's/.*EXP=([^\ ]*).*/\1/g')
    TOKENDATE=`date -d $EXPDT +%s` # get expire date
    TODAY=`date +%s` # get today date in seconds
    DAYSLEFT=$(( ($TOKENDATE-$TODAY)/(24*3600) )) # get days left

    NOTE=`mktemp`
    echo Hostname ..... : `hostname` >> $NOTE
    echo DNS Name ..... : $(nslookup `hostname` | \
grep -i "Name:"|sed -e 's/Name:\s*//') >> $NOTE
    echo >> $NOTE
    echo Today's Date: ..... : $(date +"%m/%d/%Y") \ (mm/dd/yyyy\ ) >> $NOTE
    echo Expiration date .... : $EXPDT >> $NOTE
    echo Days remaining ..... : $DAYSLEFT Days >> $NOTE

    if [ $THRESHOLD -ge $DAYSLEFT ]; then # if days left <= threshold
        # send mail or other actions.
        # This is where you would send an email or take other actions.
        echo >> $NOTE
        echo "**** USB Serial $SERIAL will expire within $THRESHOLD \
" days. ($DAYSLEFT remaining)" >> $NOTE
    fi
fi

```

```

echo "**** A new license file is required by $EXPDATE" >> $NOTE
echo >> $NOTE ; token >> $NOTE ; echo >> $NOTE
echo Sending $NOTE to $EMAILADDRESS for serial $SERIAL...
mail -s "RDT $SERIAL expiration warning: $DAYSLEFT days ($EXPDT)" \
$EMAILADDRESS < $NOTE
fi
cat $NOTE
rm $NOTE
fi

```

Capability configuration of the USB hardware device

Based on details in the update file that is applied to the USB hardware device, the device is configured to make z Systems Development and Test Environment capable of operating within only one of the following parameters:

1. Parallel Sysplex support is not enabled and you are not required to use Rational Tokens.
2. Parallel Sysplex support is enabled and you are not required to use Rational Tokens.
3. Parallel Sysplex support is not enabled and Rational Tokens are required.
4. Parallel Sysplex support is enabled and Rational Tokens are required.

Although the hardware device itself permits the application of more than one update file, it is best to avoid mixing Sysplex and Non-Sysplex update files on the same USB hardware device. Mixing the update files can make it difficult to manage and track the deployment of license entitlements and potentially separate expiration dates for each update file that is applied to the device. Because of these concerns, the Rational License Key Center does not generate multiple update files for the same USB hardware device.

License keys and the Application Developer Controlled Distribution (ADCD)

z/OS 1.13 ADCD and earlier distributions can be installed without access to an activated license key file. All volumes are supplied as files in compressed (.gzip) format, and the **gunzip** command makes them usable by z Systems Development and Test Environment.

Beginning with the z/OS 2.1 ADCD for Rational Development and Test Environment for z Systems, an activated license key file is required to install the system resident volumes of the z/OS ADCD. The **Z1091_ADCD_install** command is used to install these system resident volumes. This command requires access to an activated license key file, either on a locally attached USB hardware device, or on a product license server with connectivity to the instance that does the installation. The same activated license key file that entitles the user to run z Systems Development and Test Environment also entitles the user to install a z/OS ADCD.

For instructions on installing the z/OS ADCD, see “Installing May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 57

Parallel Sysplex Activations

Some entitlements of z Systems Development and Test Environment enable emulation of the Coupling Facility under z/VM while others do not. Entitlements that enable the coupling facility are referred to in z Systems Development and Test Environment documentation as Sysplex activations.

Although a coupling facility is not related to an individual processor in real z Systems hardware, every CP in a z Systems Development and Test Environment instance requires a corresponding Sysplex CP activation to enable the coupling facility for the instance.

Similarly, instances that do not enable the coupling facility require non-Sysplex CP activations for every CP in the instance. The update file that you receive to activate your USB hardware device is based on the type of license you purchased.

A single instance of z Systems Development and Test Environment with Parallel Sysplex enabled runs z/VM, the coupling facility, and one or more instances of z/OS. The available CPs defined to the instance are shared among all systems that are running, and are defined in the processor setting of the *devmap*. Configure at least one CP for each z/OS guest. A single instance of z Systems Development and Test Environment with Parallel Sysplex can be configured to use a maximum of 8 CPs.

Rational Token Activations and the USB hardware device

USB activations that are specific to Rational Tokens must be used when Rational Token licensing is implemented. The emulator uses the USB hardware device activation that is stored on the USB hardware device to determine whether Rational Tokens are required, then communicates with the Rational License Key Server to determine whether enough tokens are available to activate the number of CPs requested for this instance.

Attention: When you are using multiple USB hardware devices, especially in an SHK license server, you must not mix activations that require Rational Tokens with activations that are not intended for Rational Tokens.

If the emulator receives inconsistent USB activations, your environment might be out of compliance with your licensing terms or you might be unable to access some of the CPs to which you are entitled.

z Systems Development and Test Environment usage can be governed by Rational Tokens. When you use Rational Tokens, the emulator coordinates with a Rational License Key Server instance and starts or continues to run only if sufficient Rational Tokens are available.

Use of Rational Tokens does not replace the USB hardware device. A USB hardware device that is activated with a valid update file is still required.

When z Systems Development and Test Environment is started, the emulator verifies that a valid USB hardware device activation is found and enough Rational Tokens are available from the Rational License Key Server to run the emulator. Rational Tokens control product operation while enough CPs are available from the USB hardware device or devices used. Each high capacity USB hardware device that is activated for use with Rational Tokens has a limit of 100 CPs. If you need to serve more than 100 CPs across all product instances, you can obtain extra high capacity USB hardware devices and activate them for use with Rational Tokens. In the Rational License Key Center, you can generate as many extra update files as needed to activate additional USB hardware devices. Multiple USB hardware devices can be attached to the product license server.

Both the activated USB hardware device and the Rational Tokens must be available to the emulator while the emulator is running. The emulator suspends operation if either becomes unavailable.

Related information:

“Setting up Rational Tokens” on page 205

Rational Token licensing is an entitlement that you can purchase and use to run z Systems Development and Test Environment. With Rational Tokens, z Systems Development and Test Environment maintains a connection to a Rational License Key Server and starts and continues to run only when sufficient Rational Tokens are available.

Planning for USB hardware devices

You can spread your license entitlements across one or more USB hardware devices.

How many USB hardware devices are needed, and how those entitlements are activated on each device, is dependent upon many factors, such as:

- How many instances you run
- The capabilities of the underlying platforms
- The access of the different users to the different platforms
- The types of license entitlements you use

Consider the following rules when you are planning how to spread your license entitlements across your USB hardware devices, and how to later generate your update files:

- Activate a USB hardware device with only one update file.
- You can generate each update file with only one license type. That update file defines the capability configuration for the device.
- You can combine orders (license entitlements) in one update file, but only if they are of the same license type
- You can use multiple USB hardware devices in a single Linux environment, such as on a product license server. However, the following restrictions apply.
 - Do not have multiple USB hardware devices where one device is activated for Rational Tokens and one is not.
 - It is unpredictable which USB hardware device is used in obtaining entitlement to run any particular z Systems Development and Test Environment instance.

Note: Using multiple USB hardware devices in a single Linux environment can particularly affect product license servers in environments that use both Sysplex and non-Sysplex activations. For more information, see USB hardware device considerations when you use the coupling facility.

Obtaining an update file

To activate a USB hardware device and enable the z Systems Development and Test Environment to run, you must first obtain a unique *update file* for each USB hardware device you are using.

You must then apply the update file to the USB hardware device.

Updates to the processes are published on the IBM Development and Test Environment Hub on developerWorks.

Important: The method for obtaining update files for the Rational Development and Test Environment for z Systems USB hardware device is different from the method that is described in the zPDT Guide and Reference.

Most update files are now distributed as part of an archive with a file extension of .zip. These files are obtained through the Rational License Key Center. For version 9.1 and later, the *.zip file is applied to the USB hardware device without being extracted. The *.upw file that used to be delivered alone to update the USB hardware device for all entitlements is now in the compressed file (.zip file) along with other files necessary for version 9.1.0 and later releases of the offering. If you are using earlier releases of the offering and receive a compressed file, you can extract the compressed file and continue to use the *.upw file alone with your USB hardware device just as you have in the past. Offering releases before version 9.1.0 do not require any other content in the compressed file.

To obtain an update file the first time, log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login>, select your account and use the **Get Keys** screen to select the order for which you want to generate and download an update file.

If you want to change the license entitlements for a USB hardware device you are already using, you must first return the update file that was previously created for that device, and then generate a new update file. The Rational License Key Center does not generate a new update file for a USB hardware device unless any previous update file for that same device is returned. To renew an update file, log in to the Rational License Key Center, select your account and use the **Return Keys** link to return the old update file. Then, use the **Get Keys** link to generate a new one. The return function does not require that you provide the old update file. The return function signals the Rational License Key Center that you can regenerate a new update file.

If you have a previously activated USB hardware device, you can use it without change to install IBM z Systems Development and Test Environment v9.1 or later and the z/OS 1.13 ADCD. However, if you want to install IBM z Systems Development and Test Environment v9.1 or later and the z/OS 2.1 ADCD or later with a previously activated USB hardware, the USB hardware device must be activated with an update file in the *.zip format. If the update file is in the *.upw format, return that update file and obtain a replacement in the new .zip file format.

Before you sign in to the Rational License Key Center, locate the serial number of your USB hardware device. You must know this number to generate an update file. To locate the serial number of a key, turn the key to the side opposite the colored label. Three rows of numbers are etched on the device. The lowest or bottom row of numbers is the serial number. The serial number is always of the form 03-xxxxx or 02-xxxxx where xxxxx is five hexadecimal digits.



Figure 1. USB hardware device

In Figure 1, the entire serial number is **02-00222**.

If you are unable to read the etched serial number from the USB hardware device contact IBM support. They ask for a “request file” created by the SecureUpdateUtility or the Z1091_token_update command with the -r switch. From this request file, IBM can determine the serial number of the USB hardware device.

Note: The Rational License Key Center now requires the entire serial number of the USB Hardware Device, not just the last (5) digits. For example, 03-00123.

If you are going to apply an update file that enables the coupling facility, see “Enabling the coupling facility” on page 184.

Getting the update file from the Rational License Key Center

Learn how to use the Rational License Key Center to obtain update files.

If your installation is using Rational Tokens to govern entitlement to z Systems Development and Test Environment, see “Obtaining update files for Rational Tokens” on page 91.

Getting the initial file:

Obtain an update file from the Rational License Key Center.

1. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
2. On the left side of the screen, select **Get Keys**
3. Select the product line for z Systems Development and Test Environment. You see a list of license key names that show the z Systems Development and Test Environment licenses that are available.
4. Select the z Systems Development and Test Environment license type that you want to apply to the key.
5. You see one or more boxes that show the separate orders that you can apply to one or more USB hardware devices.

Important: For some license entitlements, you can combine multiple orders on a single USB hardware device or spread the licenses across multiple USB hardware devices. Regardless of how many z Systems Development and Test Environment instances you deploy for your licensed users, you must make sure that you maintain proper entitlement for the number of users who access each instance of the product. License entitlements based on Resource Value Units (RVUs) do not limit the number for users who access the offering. All other types of License entitlements require at least one product entitlement for each user who accesses the product.

6. Check the box next to all of the orders from which you would like to use licenses and click **Next**.
7. A screen is displayed showing a table that you must complete with the USB hardware device serial numbers you want to activate. Depending on the license entitlement that is selected, you might also be asked for the number of instances, the number of licenses to associate with each USB hardware device, and the number of Emulated Central Processors you intend to use for the specific USB Hardware Device activation.

You can enter information for more than one USB hardware device. A separate update file is generated for each key specified. You can download the generated update files either during this process or from the **View keys by host** screen.

More help is available by clicking the column headings in the table, but briefly stated:

- a. The **Serial Number of USB Hardware Key** field is where you specify the full eight character serial number that is etched on the last line of the USB hardware device. It has a format of 02-xxxxx or 03-xxxxx where xxxxx is five hexadecimal digits. The two-digit prefix and the dash must be included when you specify the serial number.
 - b. The **Number of Server Instances** field refers to the number of separate emulated systems you plan to enable with each USB hardware device. The number of central processors (CPs) that is available is automatically calculated for you. For some older entitlements, this value is forced to one.
 - c. The **Number of Licenses** field is applicable to some offering entitlements. Where applicable, each user who accesses a separate instance of a virtual z Systems machine requires a separate license. Individual users are licensed to a specific instance of a virtualized z Systems. The total number of licenses for all serial numbers must not exceed the number of licenses to which you are entitled. For example, if you are entitled to five Authorized User licenses, you might enable 2 instances, and 5 licenses so that one workstation can be used by two people and another workstation can be used by three people. Designating the number of licenses that are associated with a USB hardware device ensures that you are tracking the number of entitlements used.
 - d. The **Emulated Central Processor** field is applicable only to entitlements metered as Resource Value Units (RVUs). For this entitlement, each emulated central processor within an instance of a virtual z Systems machine requires a separate RVU entitlement. For each serial number, you must enter the total number of emulated central processors that you intend to enable with that specific USB hardware device. The total number of emulated CPs for all serial numbers must not exceed your total number of RVU entitlements.
8. After you indicate how you want to distribute license entitlements across USB hardware devices, select **Generate** at the bottom of the page and you are presented with a screen where you can download the update files.
 9. Click **Download** for each update file that is generated to save the update files. If you need to download the update files in the future, use the **View Keys by Host** link on the left side of the Rational License Key Center web page.

Note: You cannot generate separate activations for the same USB hardware device at the same time. Before you generate update files for a particular USB hardware device that you want to reuse, be sure to return the update file that was previously created for that device. Returning license entitlements that are assigned to the USB

hardware device also makes the corresponding active entitlements available for assignment to a USB hardware device again.

Getting the replacement file:

For perpetual license entitlements, USB hardware device activations are set to expire one year from the date an update file is generated. For this type of entitlement, you can return previously generated update files at any time and generate a new update file.

For term license entitlements, if term expiration occurs in more than one year, USB hardware device activations are set to expire one year from the date an update file is generated. Update files can be returned and generated again before term expiration. If term expiration occurs in less than one year, USB hardware device activations are set to expire at term end. After term end, update files cannot be generated in the Rational License Key Center.

To generate a new update file, you must first return the existing license entitlement in the Rational License Key Center, and then generate a new update file. Returning the license entitlements is a process of telling the Rational License Key Center that you are no longer using the license entitlements that you previously assigned to your USB hardware device. You do *not* need to return the physical update file to the Rational License Key Center.

You can return a license entitlement in the Rational License Key Center in several ways. The easiest method is to use the **View Keys by Host** link. You can also use the **Return Keys** link.

For z Systems Development and Test Environment, the term **host** in the Rational License Key Center refers to the USB hardware device that is uniquely identified by its serial number.

1. Log in to Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
2. On the left side of the screen, select **View Keys by Host**.
3. Select the serial number of the USB hardware device you want to work with. This serial number is in the **Host** column.
4. A table is displayed with data for the USB hardware device selected. At the far right of the table, click the **Change** link.
5. You see a list of devices with license entitlements that are assigned to them from the same Order Line. Locate the serial number of the USB hardware device you are working with and click **Return**. A message is displayed to confirm that the license entitlements were successfully returned.
6. Use the **Get Keys** link on the left side of the page to generate a new license. For detailed instruction on generating new licenses, see “Getting the initial file” on page 89

Obtaining update files for Rational Tokens

Learn how to use the Rational License Key Center to obtain update files if your installation is using Rational Tokens to govern entitlement to z Systems Development and Test Environment.

Use of Rational Token licensing requires purchase of Rational Tokens. After the Rational Tokens are purchased, follow these instructions for obtaining Rational Token license files and update files for your USB hardware devices.

Getting the initial update file and Rational Token license:

Learn how to get an update file for your USB hardware device and a Rational token license file from the Rational License Key Center.

1. Log in to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> and select your account.
2. On the left side of the screen, click **Get Keys**
3. Select a product line that contains your Rational Token orders with z Systems Development and Test Environment.
4. On the "Select License Keys" screen, you see one or more boxes for orders. Find the order that includes z Systems Development and Test Environment. Make a note of the part number for the z Systems Development and Test Environment product you want to enable. Select the order that contains the parts you noted and click **Next**.
5. Enter the required information to create the Rational Token license file for your host. Additional information for each field is available by clicking the descriptive link next to the input field. Fields with a red star are required fields.
6. Click **Generate** to generate the Rational Token license file. You are presented with a screen with the generated license information.
7. Click **Download License Keys** to download the Rational Token license file to your computer. This token license file is the file that is applied to your Rational License Key Server. It is named `license.dat` by default.
8. On the same screen, press **Generate Token License Keys for USB Hardware Device** to generate and download the update file for your USB hardware device.
9. A **Required Information** screen is displayed showing a table that you must complete with the USB hardware device serial numbers and associated z Systems Development and Test Environment product you want to enable with each device.

You can enter information for more than one USB hardware device. A separate update file is generated for each key and can be downloaded either during this process or at some time in the future from the **View keys by host** screen.

The serial number is the full eight character serial number that is etched on the last line of the USB hardware device. It has the format 02-xxxxx or 03-xxxxx where xxxxx is five hexadecimal digits. The two-digit prefix and the dash must be included when you specify the serial number.

Use the **Catalog Item** list to confirm the part that you want to use to generate an update file. This confirmation is important if you have entitlement to both z Systems Development and Test Environment and z Systems Development and Test Environment with Parallel Sysplex. You cannot combine Token and Sysplex Token activations on a single USB hardware device. From the list, select the entitlement part number that corresponds to the activation you want on the specified USB hardware device.

10. After you indicate how you want to distribute licenses across USB hardware devices, click **Generate** at the bottom of the screen and you will be presented with a screen where you can download the update files.
11. Click **Download** for each update file to save the update files. If you need to download the update files at some time in the future, use the **View Keys by Host** link on the left side of the page.

Note: You cannot generate separate activations for the same USB hardware device at the same time. Before you generate update files for a USB hardware device that you want to reuse, be sure to return the update file that was previously created for that device. Returning license entitlements that are assigned to the USB hardware device also makes the corresponding active entitlements available for assignment to a USB hardware device again.

Getting a replacement update file and Rational Token license:

To return and regenerate your update file and existing Rational Token license files, use the **Return Keys** function in the Rational License Key Center. Use this process to renew the USB hardware device activation for another year.

Do these steps:

1. Go to the Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login> to return the existing USB hardware device update file. For more information about returning update files, see “Getting the replacement file” on page 91.
2. After you return the update file, click **Get Keys** on the left side of the screen.
3. Click the product line for IBM Rational Tokens.
4. You see a screen titled **Select license keys**. Keys that include z Systems Development and Test Environment have a link to generate update files for USB hardware devices.

Click the link titled **Generate Token License Keys for USB Hardware Device** and complete the form to generate and download the new update file.

Activating the USB hardware device

Applying the update file that you obtained from the Rational License Key Center to your USB hardware device activates the device. Each time that you apply an update file, it overwrites the previous activation on the USB hardware device. To properly activate, the update file must be generated with the same serial number as the USB hardware device that it applies to. To change the activation of a USB hardware device, you must obtain and apply a new update file that activates the total number of license entitlements you intend to use on that device. Changing the activation of a USB hardware device involves returning and regenerating licenses in the Rational License Key Center.

The process for applying update files to USB hardware devices recently changed. For instructions on applying update files to USB hardware devices for z Systems Development and Test Environment version 9.1 or later, see “Activating a USB hardware device” on page 94

For instructions on applying update files to USB hardware devices for Rational Development and Test Environment for z Systems version 9.0 or earlier, see “Activating a USB hardware device for version 9.0 or earlier” on page 95

Important: In a product license server configuration, do not mix USB hardware devices that require Rational Tokens with USB hardware devices that do not require Rational Tokens. This approach is not supported and can result in unpredictable behavior.

Activating a USB hardware device

Before you insert the USB hardware device into a computer's USB port, verify that the computer's clock is set correctly. If your computer is new or recently repaired, ensure that the clock is correct before you insert the USB hardware device.

Note: Never set the system clock to a future date or time when the USB hardware device is plugged in or you will damage the hardware device.

Note: Verify that the Linux system clock is correct. If the USB hardware device is attached to a computer with a clock set to a future date, the device remembers the future date, will become inoperable when the clock is restored to the current time, and will remain inoperable until the computer's clock again reaches that future date. If z Systems Development and Test Environment is unable to authenticate with the USB hardware device because the clock was set to a future time or date, you receive a message that indicates a "time cheat" state.

Set your system clock to Coordinated Universal Time (UTC) to avoid errors that are caused by semi-annual time changes (for example, Daylight Saving Time).

The update file that you obtain from the Rational License Key Center is an archive file with the extension .zip. With Rational Development and Test Environment for z Systems version 9.1 or later, do not extract the contents of the archive.

In Rational Development and Test Environment for z Systems version 9.1 or later, the Z1091_token_update command replaces the SecureUpdateUtility command that is used in previous releases.

To activate your USB hardware device, connect it to a computer that has Rational Development and Test Environment for z Systems version 9.1 or later installed and run the Z1091_token_update command. The step-by-step procedure to activate your USB hardware device is as follows:

1. Connect the USB hardware device to the computer that is running z Systems Development and Test Environment.

Note: Connect only the USB hardware device that is being activated to your computer during this process, or unpredictable results might occur.

2. Use a browser on your Linux system to download the update file directly from the Rational License Key Center or use whatever method is available to copy your update file to that computer (for example, FTP, SFTP, or an external hard disk).

Note: Be sure to use the binary transfer mode and not a text mode such as ASCII.

Tip: The Z1091_token_update cannot process a path when any directory in the path or the update file name contains spaces. This restriction remains true even if the path and file name are supplied to the Z1091_token_update command in single or double quotation marks. Therefore, ensure that neither the path to the update file nor the update file name itself contains spaces.

3. Log in to the computer and, if you are not running as root, enter su followed by the root password.
4. Change to the /usr/z1090/bin directory

```
cd /usr/z1090/bin
```

5. Run the `./Z1091_token_update -u update-file.zip` command where *update-file.zip* is the full path and file name of the update file. This command produces several messages that indicate that the update was successful.
6. After the update is successfully applied, unplug the USB hardware key. Wait at least 10 seconds and then reconnect the hardware key. It is now ready for routine zPDT operation.
7. To ensure that the system sees the newly applied licenses, from root, restart the local SHK license server: `/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart`
8. Optionally, you can verify that licenses are now available on your USB hardware device by using the command:

```
./Z1091_token_update -status
```

Tip: After z Systems Development and Test Environment is started, you can verify the effective z System serial number, the token number, and the expiration date of the license. Enter the zPDT command **token** with no operands.

Tip: The token command might return different results based on the `cpuopt zvm_couplingfacility` setting in the device map configuration file, called the *devmap*. For more information, see “Enabling the coupling facility” on page 184.

Stand-alone .upw update files:

Before IBM started to distribute compressed file archives containing update files, stand-alone update files with an extension of `.upw` were distributed. If you have an update file with an extension of `.upw`, you can use the `Z1090_token_update` command to apply the update file to your USB hardware device. The activated hardware device still allows the emulator to run. However, you cannot install the z/OS 2.1 ADCD for Rational Development and Test Environment for z Systems with this kind of activation.

USB hardware device and Sysplex activation:

When you are using the `Z1091_token_update` command to apply an update file for Parallel Sysplex operation that is provided in compressed file format, it is no longer necessary to apply the `RDT_SP.NLF` file to prepare the USB hardware device. That step is done automatically by the `Z1091_token_update` command.

For instructions on enabling the coupling facility after you have an activated USB hardware device, see “Enabling the coupling facility” on page 184.

Activating a USB hardware device for version 9.0 or earlier

Before you insert the USB hardware device into a computer's USB port, verify that the computer's clock is set correctly. If your computer is new or recently repaired, ensure that the clock is correct before you insert the USB hardware device.

Note: Never set the system clock to a future date or time when the USB hardware device is plugged in or you will damage the USB hardware device.

Note: If the USB hardware device is attached to a computer with a clock set to a future date, the device remembers that future date, will become inoperable when the clock is restored to the current time, and will remain inoperable until the

computer's clock reaches that future date. If z Systems Development and Test Environment is unable to authenticate with the USB hardware device because the clock was set to a previous time or date, you receive a message that indicates a "time cheat" state.

If the update file you obtain from the Rational License Key Center is an archive file with the extension .zip, you need to extract the contents of the archive to a temporary directory during the process that is described here to get the *.upw file that is needed for z Systems Development and Test Environment version 9.0 or earlier releases.

If you have an older update file that has an extension of .upw, you can use that file with the SecureUpdateUtility as described in step 4, step 5, and step 7.

If you follow this process, your USB hardware device might not be able to install the z/OS 2.1 volumes. This restriction is true if the hardware key was never activated for z Systems Development and Test Environment v9.1.0 or later.

Read the entire set of instructions before you start activating your USB hardware device. In summary, you connect the device to a computer that has z Systems Development and Test Environment installed. Extract the contents of your update compressed file and run the SecureUpdateUtility command against the *.upw file that was extracted from the .zip file. The entire procedure is described here:

1. Connect the USB hardware device to the computer that is running z Systems Development and Test Environment.

Note: Update files are written for specific USB hardware devices, so it is acceptable to have more than one USB hardware device that is attached to the computer during this process.

2. Use a browser on your Linux computer to download the update file directly from the Rational License Key Center or use whatever method is available (for example, FTP, SFTP, or an external hard disk) to copy your update file to that computer.

Note: Be sure to use the binary transfer mode and not a text mode such as ASCII.

3. Log in to the computer, create a new, temporary directory and extract the contents of your update file to that directory. For example:

```
mkdir /tmp/myupdatedir
cd /tmp/myupdatedir
unzip update-file.zip ls
```

You see a list of file names. Note the name of the file that ends in .upw. The upw file has a name similar to RDT-xxxxx-yyyymmddhmm-Lcc.upw or RDT-xxxxx-yyyymmddhmm-Lcc.-Sysplex.upw where:

xxxxx Is your device's serial number

yyyymmddhmm

Is the expiration date of the activation

cc Is the number of licensed CPs in hexadecimal format

4. If you are not running as root, enter su followed by the root password.
5. Change to the /usr/z1090/bin directory

```
cd /usr/z1090/bin
```

6. If you are applying an update file for Parallel Sysplex, run the **SecureUpdateUtility** command with the **-n** switch and the full path of the **adcd.NLF** file. For example, enter this command on one line:

```
./SecureUpdateUtility -n /tmp/myupdatedir/adcd.NLF
```

This command produces several messages but indicates that the update was successful or that the license exists, meaning the update file was already applied. You need to run this command only one time for a USB hardware device.

Note: Running this command each time you apply an update file does not cause problems.

7. Run the **./SecureUpdateUtility -u update-file.upw** command where *update-file.upw* is the full path and file name of the update file. For example, enter the following command on one line:

```
./SecureUpdateUtility -u /tmp/myupdatedir/RDT-00DF0-201502152359-L03-Sysplex.upw
```

This command produces several messages but indicates that the update was successful.

8. To ensure that the system sees newly applied licenses, run the following command to restart the local product license server:

```
/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart
```

9. You can verify the effective z System serial number, the token number, and the expiration date of the license after z Systems Development and Test Environment is started. Issue the **zPDT** token command with no operands.

Tip: The token command can return different results based on the **cpuopt** **zvm_couplingfacility** setting in the device map configuration file, called the *devmap*. For more information, see “Enabling the coupling facility” on page 184.

For answers to common errors related to USB hardware device activation, see Troubleshooting license manager and USB hardware device activation .

Quick setup instructions for using and migrating the product license server

Learn how to set up the product license server, how to set up other computers to access the product license server, and how to migrate the product license servers and client instances that use product license servers.

Setting up the product license server

Learn how to set up the product license server.

Configuring a product license server by using the default ports can be accomplished with the following steps:

1. Obtain the update files for the USB hardware devices you want to use with the SHK license server.
2. Install z Systems Development and Test Environment on the computer you want to function as the SHK license server. Follow the instructions for installing z Systems Development and Test Environment in Chapter 4, “Installing z System Development and Test Environment,” on page 37. You do not have to install Rational Test Control Panel.

Note: This computer can be a virtual machine only if it can access the USB hardware device.

3. Connect the USB hardware device to the product license server computer. If you are running the SHK license server in a virtual machine, be sure that the USB hardware device is routed to that virtual machine.
4. Activate the attached USB hardware device. This process is described in “Activating the USB hardware device” on page 93.
5. Log in to the computer as a non-root user and enter the `uimserverstart` command.
6. If you have a firewall that is enabled, verify that your firewall allows connections to ports 9450 and 9451. If your firewall is based on IP tables, commands like

```
# iptables -I INPUT -p tcp --dport 9450 -j ACCEPT
```

or

```
# iptables -I INPUT -p tcp --dport 9451 -j ACCEPT
```

entered from a root user ID can be used to enable your firewall to allow traffic to these ports:

Note: This procedure is an example. Details about managing your Linux firewall and your external routing controls are Linux beyond the scope of this document.

You might want to alter the server configuration to allow remote access to the browser-based configuration interface, to restrict access to specific hosts to enable logging, or to complete other tasks. For details on configuring product license servers and to familiarize yourself with the product license server, see “zPDT license servers” on page 208.

Setting up computers to access the product license server

The computers that authenticate from a product license server instead of a locally attached USB hardware device can be set up by running the `clientconfig` program as root. This process assumes that your product license server is using all of the default ports.

1. Log in to each computer that runs the emulator with authentication from a product license server.
2. Enter the following commands:

```
su (enter root password when prompted)
cd /usr/z1090/bin
./clientconfig
```
3. A window opens with several fields, then complete these fields.
 - a. Leave Gen2 Contact Server empty. This field is used for software-based license manager.
 - b. Leave Gen2 BackupServer empty. This field is used for software-based license manager.
 - c. Set Gen1 Contact Server field to point to IP address or hostname of product license server.
 - d. Set Gen1 BackupServer if you want to set up the failover product license server. Set the field to point to IP address or hostname of failover product license server.

- e. Set UIM ContactServer if you want to use different UIM server than product license server. Otherwise, leave the field empty. This field is optional.
 - f. Leave UIM Local Serial Random empty. This field is optional.
 - g. Press Enter twice to save these values.
4. Optionally, you can use command line interface to complete client configuration. To invoke the client, enter the following command:
- ```
cd /usr/z1090/bin/
./clientconfig_cli <parameter>
```
- You can write the client configuration file `sntlconfig.xml` for z1090/z1091. `clientconfig_cli` is usually run by the z1090/z1091 administrator as a superuser root.
- The following parameters are accepted.
- [g1s1 | g1s2 | g2s1 | g2s2] servername**  
Specify Gen1 or Gen2 server. One contact server (s1) and one backup server (s2) can be specified.
  - ucs server**  
Specify UIM contact server. The setup is the same as license server by default.
  - usm y/n**  
Specify UIM serial method.
  - l** List servers currently configured.
  - h** Help, and print this message.

For details on configuring clients to the product license server, see the “License Server” chapter in the zPDT Guide and Reference.

## Migrating product license servers and client instances that use product license servers

Learn how to migrate product license servers and client instances that use product license servers.

### Migrating clients that use product license servers

In most cases, you can migrate z Systems Development and Test Environment instances that use product license servers (clients) to newer versions or releases without migrating your product license server to the newer release. The USB hardware device on the product license server generally does not need to be updated, and the client is able to use the product license server as soon as the installation of the client is complete. This capability allows for installing new releases without stopping the product license server, and thus not affecting other users.

For more information about installing the client, see Chapter 4, “Installing z System Development and Test Environment,” on page 37. For more information about configuring the client for access to the product license server, see “Setting up computers to access the product license server” on page 98.

### Migrating product license servers

Upgrade your product license server to at least the highest level of z Systems Development and Test Environment in use by any of its client instances.

The product license server must be upgraded only in the following case:

- Your current product license server is running z Systems Development and Test Environment Version 9 Release 0 or below, or the USB Hardware Device of the product license server was activated with a .upw file instead of a .zip file.
- You are upgrading a client to Version 9 Release 1 or above
- You are installing a z/OS 2.2 ADCD for z Systems Development and Test Environment on the client

**Tip:** Even in this case, you can upgrade a client to z Systems Development and Test Environment Version 9 Release 1 and above without migrating your product license server. See <http://www.ibm.com/support/docview.wss?uid=swg21959847>.

Upgrading your product license server to z Systems Development and Test Environment Version 10 can be accomplished with the following steps:

- Obtain a \*.zip update file for the USB Hardware Device that is used by the SHK license server. For the step-by-step procedures, see “Obtaining an update file” on page 87.
- Install z Systems Development and Test Environment v10. The instructions for installing z Systems Development and Test Environment are in Chapter 4, “Installing z System Development and Test Environment,” on page 37.
- Activate the USB Hardware Device. For the step-by-step procedure, see “Activating the USB hardware device” on page 93.
- Set up the product license server. For the quick setup instructions for setting up the product license server after you finish installing z Systems Development and Test Environment, see “Setting up the product license server” on page 97.

**Note:** Any z Systems Development and Test Environment instances that require authentication from the product license server during the upgrade will not run until the upgrade is complete.

After the upgrade to the product license server is complete, any z Systems Development and Test Environment instance can use the product license server both to install and run any z/OS distribution, without a locally attached USB Hardware Device. Also, assuming the product license server is at the same IP address and port as the previous version, no additional actions are required to existing z Systems Development and Test Environment instances that use that product license server. Down level z Systems Development and Test Environment instances can run accessing a z Systems Development and Test Environment v9.5 product license server without change.

## Next steps

You might need to issue a **UIMRESET -1** command before you use z Systems Development and Test Environment with an activated USB hardware device.

These cases are the most common reasons to issue the command:

- If you were using a single local USB hardware device, and are replacing it with another USB hardware device or want to use multiple USB hardware devices.
- If you change the disk containing the Linux partition or upgrade the Linux kernel.
- If you are using a product license server or license manager and the UIM serial number in the local database conflicts with a serial number in the license server.

For more information about UIM serial numbers, how they are assigned, and when resetting the UIM serial numbers is required, see “zPDT license servers” on page 208.

If you worked through the previous sections of this IBM Knowledge Center, you now have one or more activated license keys that are ready to enable operation of one or more instances of z Systems Development and Test Environment. For the next steps to follow to use the offering, see “Installing an operating system on z Systems Development and Test Environment” on page 55.



---

## Chapter 6. Configuring an instance of z Systems Development and Test Environment

Learn how to set up z Systems Development and Test Environment and customize a z/OS software distribution for development and test purposes.

These instructions are intended to allow a person with very little z/OS systems programming experience to configure the z Systems Development and Test Environment.

---

### Defining the device map

The z Systems Development and Test Environment allows the customization of the z Systems resources available within the virtualized environment. The resources can be defined in a device map or *devmap*.

The sections and syntax of device maps are explained in detail in the "1090 Control Files" section of the *zPDT Guide and Reference*.

The system volumes that represent the z/OS distribution are defined in an *awsckd* stanza, and are mounted at arbitrarily chosen addresses within the range of valid addresses that are defined by the z/OS distribution's IODF. Historically in a z/OS ADCD, the *xxRES1* and *xxSYS1* volumes are mounted at the same addresses as those in the *zPDT Guide and Reference* (0A80 and 0A82), and address 0AA3 is always reserved for volume *xxDBAR*.

If you must alter the default tunnel IP addresses of 10.1.1.1 and 10.1.1.2, refer to the example of the `--tunnel_ip` parameter in the *zPDT Guide and Reference*.

The following examples, when combined, define a working device map with a sample z/OS distribution from a z Systems Development and Test Environment ADCD subscription. If you are using the coupling facility or Rational Tokens, see "Enabling the coupling facility" on page 184 and "Setting up Rational Tokens" on page 205.

**Note:** The *processors* statement specifies the number of z Systems CPs that are used in this instance. The default is one. This number must not be more than the number available on the activated USB hardware device or more than the number of CP activations made available through a license manager.

```
[system]
memory 4096m
processors 3
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 tso1
```

For the network adapter definitions, the following example was added. Your setup might differ, and you are encouraged to consult the sections on connectivity in the *zPDT Guide and Reference*.

```

[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 volumes are in the /home/ibmsys1/z1090/disks/ directory, the DASD definitions might look like the following example:

```

[manager]
name awsckd 0001
device 0A80 3390 3390 /home/ibmsys1/z1090/disks/xxRES1
device 0A81 3390 3390 /home/ibmsys1/z1090/disks/xxRES2
device 0A82 3390 3390 /home/ibmsys1/z1090/disks/xxSYS1
device 0A83 3390 3390 /home/ibmsys1/z1090/disks/xxCFG1
device 0A84 3390 3390 /home/ibmsys1/z1090/disks/xxUSS1
device 0A85 3390 3390 /home/ibmsys1/z1090/disks/xxUSS2
device 0A86 3390 3390 /home/ibmsys1/z1090/disks/xxPRD1
device 0A87 3390 3390 /home/ibmsys1/z1090/disks/xxPRD2
device 0A88 3390 3390 /home/ibmsys1/z1090/disks/xxPRD3
device 0A89 3390 3390 /home/ibmsys1/z1090/disks/xxDIS1
device 0A8A 3390 3390 /home/ibmsys1/z1090/disks/xxDIS2
device 0A8B 3390 3390 /home/ibmsys1/z1090/disks/xxPAGA
device 0A8C 3390 3390 /home/ibmsys1/z1090/disks/xxPAGB
device 0A8D 3390 3390 /home/ibmsys1/z1090/disks/xxPAGC
device 0A8E 3390 3390 /home/ibmsys1/z1090/disks/xxBLZ1
device 0A8F 3390 3390 /home/ibmsys1/z1090/disks/xxDBB1
device 0A90 3390 3390 /home/ibmsys1/z1090/disks/xxDBB2
device 0A91 3390 3390 /home/ibmsys1/z1090/disks/xxDBC1
device 0A92 3390 3390 /home/ibmsys1/z1090/disks/xxDBC2
device 0A93 3390 3390 /home/ibmsys1/z1090/disks/xxDBAR
device 0A94 3390 3390 /home/ibmsys1/z1090/disks/xxC521
device 0A95 3390 3390 /home/ibmsys1/z1090/disks/xxC531
device 0A96 3390 3390 /home/ibmsys1/z1090/disks/xxIMD1
device 0A97 3390 3390 /home/ibmsys1/z1090/disks/xxIME1
device 0A98 3390 3390 /home/ibmsys1/z1090/disks/xxINM1
device 0A99 3390 3390 /home/ibmsys1/z1090/disks/xxW851
device 0A9A 3390 3390 /home/ibmsys1/z1090/disks/xxW852
device 0A9B 3390 3390 /home/ibmsys1/z1090/disks/xxW853
device 0A9C 3390 3390 /home/ibmsys1/z1090/disks/xxW854
device 0A9D 3390 3390 /home/ibmsys1/z1090/disks/xxW855
device 0A9E 3390 3390 /home/ibmsys1/z1090/disks/xxW901
device 0A9F 3390 3390 /home/ibmsys1/z1090/disks/xxKAN1
device 0AA1 3390 3390 /home/ibmsys1/z1090/disks/xxW902
device 0AA2 3390 3390 /home/ibmsys1/z1090/disks/SARES1
device 0AA3 3390 3390 # Available for dynamic mounts
device 0AA4 3390 3390 # Available for dynamic mounts
device 0AA5 3390 3390 # Available for dynamic mounts

```

*Figure 2. Device map example*

You can verify the device map with the `awsckmap` command. All of the disk volume images that are referenced in the device map must exist before you verify the device map. Assume that the device map file has the following name:

```
/home/ibmsys1/z/myDeviceMap
```

With the name in the preceding example, the device map can be verified with the command:

## Sample program to create device map

A sample program that is called `create_devmap.pl` is available in the `ConfigGuideSample` directory where you installed z Systems Development and Test Environment. If Perl is installed on your Linux system, you can use the `create_devmap.pl` program to generate a sample device map that is based on your current decompressed 3390 disk images, memory configuration, and available network parameters. Consider the output of `create_devmap.pl` to be a starting point from which you can create a final device map.

The syntax for the `create_devmap.pl` command is shown here:

```
perl <pathtocommand>/create_devmap.pl pathtodisks > generateddevmap
```

In the preceding example, *pathtocommand* is the location of the `create_devmap.pl` file, and *pathtodisks* is the location of your 3390 disk images. *generateddevmap* is the name of the file to contain the new device map.

If you already have a static IP address that is assigned for your virtual z/OS machine and a z/OS host name that can be resolved to that address by Linux, such as through a connected Domain Name Server or static configuration, you can add the `-h <hostname>` parameters after the *pathtodisks* parameter. Adding the parameter causes the script to attempt to generate comments that contain more accurate z/OS TCP/IP configuration samples based on your network.

```
perl <pathtocommand>/create_devmap.pl pathtodisks -h
 hostname_of_zos > generateddevmap
```

The `create_devmap.pl` program creates a memory line based on existing hardware and configuration of your Linux machine. Verify that the amount of memory that is requested is appropriate for your situation.

The device map that is created by `create_devmap.pl` defines OSA devices based on the first Tun/Tap and Wired CHPIDs found that use the **find\_io** command, and a set of sample z/OS TCP/IP definitions that would correspond to the OSA device definitions in the generated device map. These TCP/IP configuration statements can be used as a starting point for your TCP/IP configuration, but probably requires changes to match your network. Verify that the device addresses and device names in your final VTAM® definitions, TCP/IP profile, and device map all correspond to the correct network adapter types.

The device map that is created by `create_devmap.pl` also contains 3390 device statements for files in the *pathtodisks* directory that are over 800 MB, have six-character names, and are verified to be disk images by the `alckcd` command.





---

## Chapter 7. Starting and stopping z Systems Development and Test Environment

Learn how to start and stop z Systems Development and Test Environment.

---

### Starting the z Systems Development and Test Environment

If you use the same directory structure that is used in these examples, and x3270 is installed on your native Linux system, you can start your z Systems Development and Test Environment system with a script similar to the example in this topic.

The script example that is used in this document is named: runzpdtd. After you create the script, you must ensure that it is executable by entering this command:  
**chmod 755 /home/ibmsys1/z/runzpdtd.**

This script is written to accept two command line switches. The -d switch can be used to specify a device map and the -l switch (the lowercase letter L) can be used to specify the load parameter. The sample runzpdtd script is available in the /opt/ConfigGuideSample directory and is shown here:

```

#!/bin/bash
LOADPARAM=CS
DEVMAP=myDeviceMap

#cd /home/ibmsys1/z # Optional: the directory from which you want to run

while getopts "d:l:" opt ; do
 case $opt in
 d)
 DEVMAP=$OPTARG
 ;;
 l)
 LOADPARAM=$OPTARG
 ;;
 \?)
 echo "Invalid paramater:" $OPTARG
 echo " runzpdtd [-d devmap] [-l loadparm]"
 exit 1
 ;;
 esac
 done

PORT=`egrep "^3270port" $DEVMAP | awk '{print $2}'`

echo "Load parm: $LOADPARAM, Devmap: $DEVMAP, Port: $PORT"

if [! -e $DEVMAP]; then
 echo "Devmap file $DEVMAP does not exist"
 exit 1
fi

echo stopping previous instance
awsstop
killall -u $(id -un) x3270
while ps -U $(id -un) |egrep "emily|aws.{3,5}" >/dev/null; do sleep 1;done

start Rational Development & Test environment. --clean is optional
echo awsstart $DEVMAP # --clean
awsstart $DEVMAP # --clean

egrep "AWS[A-Z]{3}[0-9]{3}[ES]" `ls -tl ~/z1090/logs/log_console_* | head -n 1` 1>/dev/null
if [$? -ne 0]; then
 echo
 echo "Rational Development and Test Environment started."
 echo
 /usr/z1090/bin/token
 # start x3270 for the console and one local user terminal

 nohup x3270 -model 4 mstcon@localhost:$PORT 1>/dev/null 2>/dev/null &
 nohup x3270 -model 4 tso@localhost:$PORT 1>/dev/null 2>/dev/null &

 #Perform the IPL of the system
 echo ipl a80 parm 0a82$LOADPARAM
 ipl a80 parm 0a82$LOADPARAM
else
 echo
 echo "No completed startup message was found."
 echo '*****'
 echo '*** runzpdtd terminated with errors. ***'
 echo '*****'
fi

```

*Figure 3. Startup example*

**Note:** Run this script under ibmsys1 user id or equivalent home by using directory structure /home/ibmsys1.

The *ipl* statement contains three pieces of information. The 0a80 is the device address of the SYSRES volume, which is a bootable z/OS volume. The parameter string 0A82xx specifies the 4-digit device address of the IODF volume (which holds IPL configuration files) and the LOADPARM, the suffix that identifies the LOADxx member that is used to start z/OS. In the script, the LOADPARM is a variable that you can change by using the parameter -l xx when you start the script. (The -l is a lowercase "L".)

The first time you start a new z/OS distribution, start z Systems Development and Test Environment with a loadparm that does a cold start, and does not start additional subsystems. After you verify basic z/OS capabilities, you can then customize the system by using the instructions in "Configuring a working z/OS system" on page 129 and create and start different loadparms as needed. As with any z/OS system, warm starts are less disruptive and preserve the JES job spool. Use warm starts when possible.

---

## IPLing z Systems Development and Test Environment from a remote emulated terminal for the system console

If your Linux image with the installed z Systems Development and Test Environment does not have a user interface, you cannot use an emulated terminal product such as X3270 to run your z/OS console. Therefore, you must install the emulated terminal software on a remote system, and you must be dialed to the Linux system that is running z Systems Development and Test Environment before you enter the **ipl** command.

If the remote emulated terminal product is a 3270 emulator, then the only difference between remote administration and local administration of z Systems Development and Test Environment are the commands that you enter between the *awsstart* and *ipl* commands. For remote administration, rather than starting x3270 or another local 3270 emulator, instead, start the 3270 emulator on the remote system and connect to the Linux IP address and zPDT 3270port (usually 3270). From this point forward, the remote 3270 emulator becomes your system console. Any wanted TSO sessions can be connected from remote systems as well using the defined x3270port.

If you want to use the supplied *runzpdt* script to start z Systems Development and Test Environment, you must modify it to perform remote system console administration. Any changes to *runzpdt* depend on what method of starting z Systems Development and Test Environment and automation you currently use. You can customize these alternative modifications to fit your environment.

- Remove the final stanzas after the *awsstart* that start the local x3270 sessions and perform the **ipl**. The *ipl* command can be run as a line command after the system console is connected.
- Replace the x3270 start commands in *runzpdt* with some form of pause while you start your remote system console (either manually or with automation). These examples illustrate two pause techniques:
  - A sleep statement, such as

```
sleep 1m
```
  - A read statement, such as

```
read -p "Press [Enter] key to ipl after system console is connected."
```

---

## Stopping z Systems Development and Test Environment

If possible, always shut down z/OS cleanly. Typically, shutting down cleanly begins by starting a procedure that shuts down all active subsystems.

Any z/OS ADCD for z Systems Development and Test Environment contains sample shutdown scripts for the systems that are available and active in that distribution. For an example of these shutdown scripts, see “Altering system startup and shutdown scripts” on page 142. This example shows the type of commands you can use in a shutdown script.

```
/*-----*/
/* WARN TSO USERS TO LOGOFF */
/*-----*/
F TSO,USERMAX=0 /* DON'T ALLOW ANYONE ELSE TO LOGON RIGHT NOW! */
SEND 'PLEASE LOGOFF - THE SYSTEM WILL BE IPLED IN 2 MINUTES!!',ALL,NOW
PAUSE 10
/*-----*/
/* ISSUE STOP COMMANDS FOR ALL TASKS THAT WILL TAKE THEM. FOR THOSE */
/* THAT WON'T TAKE THEM, SIMPLY ISSUE CANCEL COMMANDS. */
/*-----*/
/*-----*/
/* ISSUE MODIFY COMMANDS FOR THOSE TASKS THAT USE THEM INSTEAD OF STOP*/
/*-----*/
/*-----*/
/* TRY TO WAIT LONG ENOUGH FOR ALL TO COME COMPLETELY DOWN */
/*-----*/
P TSO
C INETD4
P LLA
SETRRS SHUTDOWN
P VLF
MODIFY DLF,MODE=Q
P DLF
P HZSPROC
P TN3270
P TCPIP
P SDSF
PAUSE 20
Z NET,QUICK
PAUSE 10
F OMVS,SHUTDOWN
```

During shutdown, you might have to respond to z/OS console messages, such as when IMS, TSO, or z/OS UNIX are stopped. You can see what programs are still running by entering the D J,L console command. Ensure VTAM and all subsystems end.

After all systems are stopped, stop your JES system. After JES ends, z Systems operation can be stopped. Enter a **QUIESCE** command from the MVS console to ensure that there is no more activity to the z/OS volume image files. The zPDT system can then be stopped with this command in the Linux window:

```
$ awsstop
```

This command produces several messages. It might be necessary to press **Enter** to obtain the Linux prompt. Any 3270 windows can be closed.

---

## Chapter 8. Customizing a z/OS distribution

Learn how to customize a migrated z/OS system, and how to customize the z/OS 2.2 Application Developers Controlled Distribution (ADCD) for IBM z Systems Development and Test Environment.

Whether you migrate a z/OS system or install the z/OS 2.2 ADCD, certain customizations are necessary or highly beneficial. If you want connectivity from your z/OS system to either the Linux system on which it is installed or other networks, you must change TCP/IP and VTAM to create or modify OSA devices and routes that work with z Systems Development and Test Environment. If your z/OS distribution provides small file systems for the /tmp and /u directories, increase them. Small file systems can cause problems, particularly when you are installing z/OS software by using SMP/E, or when programs create large memory dumps on the z/OS UNIX file system.

These topics guide you through procedures to help you make these customizations:

- For general guidance for customizing a migrated z/OS system, see “Customizing a migrated z/OS system.”
- For specific guidance tailored to customizing a z/OS 2.2 ADCD, see \*\*\*\*  
MISSING FILE \*\*\*\*.

---

### Customizing a migrated z/OS system

Learn how to customize a migrated z/OS system.

#### 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 data sets 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 often see references to LOAD*xx*, IEASYS*xx*, and so on. Configuration files refer to other members by a keyword and suffix number. For example, a member that is called IEASYSRC might identify the main z/OS UNIX configuration file with the line OMVS=RC, which means that z/OS UNIX finds the member that contains its configuration parameters by starting with a predetermined name, BPXPRM, and add the suffix RC (resulting in member name of BPXPRMRC). The keywords in the configuration files *are not* usually the same as the member name prefixes.

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

The z/OS ADCD for z Systems Development and Test Environment defines several sets of alternative libraries. The configurations for the distribution itself are stored in a set of libraries that start with the qualifiers ADCD.\*. This z/OS distribution also provides a set of libraries for you to use, which start with the high-level qualifier USER. This z/OS distribution is already set up to read from most of the USER libraries before the ADCD.\* libraries. For more information on the configuration of the z/OS ADCD for z Systems Development and Test Environment, see the specific

sessions “Customizing the May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 124.

If you are migrating your own z/OS distribution, you probably defined your own alternative libraries to contain any z/OS customizations. Your system administrator knows the library structure that is defined for your z/OS distribution, and what libraries to use to contain any customizations. For purposes of describing customizations, this IBM Knowledge Center uses USER.\* libraries, which tend to be commonly available in any z/OS distribution. When you follow these instructions, make all changes in USER.\* libraries, except for adding LOADPARM members to SYS1.IPLPARM.

**Note:** Subsystem configuration changes, such as CICS definitions, also change original libraries such as the CICS CSD.

### **The IPL process: LOADxx and IEASYSxx members**

When z/OS is started, it looks in data set SYS1.IPLPARM for a member called LOADxx. The xx value is specified in the IPL statement of the start script that was created. A LOADxx member defines various settings to start the system, such as the parmlib concatenation that indicates which data sets 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:** Create a loadparm that starts the system without pointing to any alternative libraries that contain customized parmlibs, proclibs, and so on. By creating a loadparm if you make a mistake in your customizations that causes z/OS to not start, you can try to perform an IPL of the system with the loadparm that does not point to the alternative libraries. This method gives you TSO access, where you can modify the customizations to correct any mistakes.

### **PROCLIB: System procedure library**

Parmlib members contain only configuration information. The procedures that start the various subsystems and servers are found in a different concatenation called PROCLIB. As an example, a z/OS ADCD for z Systems Development and Test Environment contains SYS1.PROCLIB, ADCD.Zxxx.PROCLIB, and USER.PROCLIB in its PROCLIB concatenation. To include any PROCLIB data set in the PROCLIB concatenation, you must modify the MSTJCLxx member of parmlib and your JES procedure.

For more information, see “Enabling use of USER.PROCLIB and IPLing” on page 121.

### **z Systems serial numbers:**

z Systems CECs have unique serial numbers, allowing software to identify the machine and LPAR.

The z Systems serial number is part of its CPU ID, a concatenation of the serial number, LPAR number, and other information. In the simple environment that is described in this IBM Knowledge Center, which is a single USB hardware device that is connected to a single PC hosting z Systems Development and Test Environment, the serial number of the z Systems created by zPDT is the serial number of the USB hardware device. When a product license server or a license manager is used, the serial numbers are randomly generated by the UIM server

that is running as part of the product license server, and assigned to the z Systems Development and Test Environment instance.

**Tip:** From the System console or TSO SDSF, a Display M command returns all assigned machine serial numbers. This information is also available by using the CSRSI callable service, which is documented in *MVS Programming: Callable Services for High-Level Languages Version 2 Release 1*, SA23-1377.

After it is assigned, whether from the local USB Hardware Device or from a license server, the z Systems serial number remains consistent, even across upgrades to new emulator versions, and when new license files are applied to the USB hardware device. This assignment allows for a consistent CPU ID. That first assigned serial number is stored locally on any z Systems Development and Test Environment instance in `/etc/z1090/uim/uimclient.db` and reused, thus allowing for this consistency even when you are switching between using a local USB Hardware Device and a license server.

The serial number can change in a few cases. The most common case is if you enter a `uimreset -l` command, which causes the serial number to be reassigned from either the local USB hardware device or from a license server. It can also change if you somehow delete or corrupt the `uimclient.db` file. For more information about z Systems CEC serial numbers and zPDT, see paragraph 8.1 “Methodology” in the zPDT Guide and Reference.

**Tip:** If consistent serial numbers or CPU IDs are a requirement, back up your `/etc/z1090/uim/uimclient.db` file after your initial z Systems Development and Test Environment activation. If the file is corrupted or changed by a `uimreset -l` command, it can be restored from the backup and thus restore the original serial number.

## Configuration overview

Learn how to configure the z/OS system so that you isolate the customized data in your z/OS distribution volumes and establish TCP/IP communications.

These configuration instructions explain how to:

- Create new z/OS UNIX file systems for `/tmp` and `/u`.
- Customize TCP/IP settings to establish network connectivity.

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

- Create TSO user IDs.
- 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

Place these customizations on a single volume, such as ensuring your `USER.*` libraries are on one volume. Placing these customizations on a single volume helps with future migrations to new versions of z/OS distributions.

## Logging on to TSO

After you IPL, you can use the x3270 emulator on the host Linux to start a TSO session

with the following command:

```
x3270 -port 3270 localhost &
```

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

```
x3270 -model 4 localhost:3270 &
```

You can also use an emulator that is not native to the host Linux. For more information, see “Logging on to TSO from emulators not native to the host Linux” on page 122.

Log on to TSO with a valid user ID in your z/OS distribution. You might want to make a few minor changes to your ISPF 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, enter this command:

```
TSO PROFILE PREFIX(tso user id)
```

## Creating new zFS 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 and alternative mount points might require a more in-depth plan.

Your z/OS distribution might provide fairly small file systems for the /tmp and /u directories. Small file systems can cause problems, particularly when you are installing z/OS software by using SMP/E, or when programs create large memory dumps on the z/OS UNIX file system. Allocate 100 primary cylinders and 20 secondary cylinders for these file systems. If the current file systems for /tmp and /u are not large enough, replace them. To create new file systems for /tmp and /u, create two new zFS files and associate these new files with the /tmp and /u directories. Catalog these two new zFS files on the same volume as your other customizations. By avoiding use of the master catalog during migration to a new z/OS distribution, you must only import the user catalog and replicate the alias definitions so that catalog entries are restored.

To create new zFS files for /tmp and /u:

- Create aliases for the high-level qualifiers of the zFS files
- Create the zFS files
- Modify the appropriate BPXPRMxx parmlib member to mount the new zFS files with the /tmp and /u directories
- Copy the existing /tmp and /u directories to the new /tmp and /u directories that are mounted with the new zFS files
- Optionally create a new load parm to contain these changes
- Shut down the system and re-IPL

Any z/OS ADCD for z Systems Development and Test Environment supplies samples with more specific instructions, including sample JCL. These samples can help you perform this step even if you are migrating your own z/OS distribution. For more information, see “Customizing the May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 124.

In the examples in these topics, the two z/OS UNIX file systems are created with a high-level qualifier of CUST on sample volume S1CUST.



## Creating ALIASes for the high-level qualifiers of the zFS files

Start by creating an alias for CUST and any other qualifiers that are used to create data sets here. Create the alias and the file systems on a volume with your other customizations so that it is easy to migrate to a new z/OS distribution.

When you create new users, also create an alias for the user ID in the same catalog. This sample JCL shows you how.

```
//Q12ALIAS JOB (ACCT),MSGCLASS=H,NOTIFY=&SYSUID.
/*-----
//DEFALIAS EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 DEFINE ALIAS (NAME(CUST) RELATE(USERCAT.Z22C.USER))
/*
```

## Creating the zFS files

Use the JCL in this topic to create two new zFS files on volume xxCFG1 named CUST.ZFS.U and CUST.ZFS.TMP.

Adjust the space allocations as needed.

```
//Q13ZFS JOB 'IBMUSER',CLASS=A,NOTIFY=&SYSUID,MSGCLASS=H
//ZFSALLOC EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 DEFINE CLUSTER(-
 NAME(CUST.ZFS.TMP) -
 VOLUME(xxCFG1) -
 LINEAR -
 CYL(100 20) -
 SHAREOPTIONS(3) -
)
 DEFINE CLUSTER(-
 NAME(CUST.ZFS.U) -
 VOLUME(xxCFG1) -
 LINEAR -
 CYL(100 20) -
 SHAREOPTIONS(3) -
)
/*
//ZFSFORMAT EXEC PGM=IOEAGFMT,REGION=0M,COND=(0,LT,ZFSALLOC),
// PARM='-aggregate CUST.ZFS.TMP -compat'
//SYSPRINT DD SYSOUT=*
//ZFSFORMAT EXEC PGM=IOEAGFMT,REGION=0M,COND=(0,LT,ZFSALLOC),
// PARM='-aggregate CUST.ZFS.U -compat'
//SYSPRINT DD SYSOUT=*
```

## Modifying BPXPRM00 parmlib member

After you create CUST.ZFS.TMP and CUST.ZFS.U, copy ADCD.Z22x.PARMLIB(BPXPRM00) into USER.PARMLIB.

Comment out the existing mount of the zFS for tmp, and replace it with a mount of CUST.ZFS.TMP. For example,

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

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

Make a similar change for the mount of /u:

```
/* MOUNT FILESYSTEM('ZFS.&ADC DLVL..USERS') */
/* TYPE(HFS) */
/* MODE(RDWR) NOAUTOMOVE */
/* MOUNTPOINT('/u') */

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

X : Replace Wild Character with current ADCD version character.

## Copying the existing /tmp and /u directories

Finally, you must copy the existing /tmp and /u directories to contain the new /tmp and /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 these commands. Be sure to verify that each command works correctly before you proceed to the next command.

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

## Creating a loadparm for customizations

You might want to isolate your customizations into a new loadparm. For example, you can create an RC loadparm that uses your new zFS files for /tmp and /u.

- Instead of modifying BPXPRM00, copy it and rename it to BPXPRMRC, and modify BPXPRMRC.
- Copy the IEASYSxx member that currently starts the subsystems you want to start to USER.PARMLIB. Rename it to IEASYSRC to match the two character loadparm of BPXPRMRC. Update the **OMVS=** parameter to replace the existing two character loadparm with RC.
- Make a copy of the existing LOADxx member in SYS1.IPLPARM that currently starts the subsystems you want to start. Rename it to LOADRC, and store it back into SYS1.IPLPARM. Modify the **SYSPARM** value to RC.

## Shutting down the system and re-IPLing

Shut down the system and perform an IPL. Use your loadparm that points to the modified BPXPRMxx, or your newly created load parm.

When the system restarts, you can enter the z/OS UNIX command line and enter the **df -k** command to ensure that your file systems are being used and mounted correctly.

After the shutdown and re-IPL is complete, you have larger zFS files for /tmp and /u.

## Setting up TCP/IP

z/OS running on z Systems Development and Test Environment can communicate with your network by using TCP/IP. You can use standard 3270 terminal emulators, FTP, Developer for z Systems, and other services to move data to and from your z/OS system.

Since mainframes are 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 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 zPDT Guide and Reference.

The subtopics in this section of the IBM Knowledge Center show an example of setting up the method referred to as Scenario 4 in Chapter 7, “LANs” in the *zPDT Guide and Reference*. This method allows the z/OS system and the Linux system to communicate with your network, each with their own IP address and with the ability to share the Ethernet adapter. Because each system has its own IP address, both systems can simultaneously use functions such as FTP with minimal configuration change. It also eliminates the need to create firewall *holes* for your z/OS ports or provide Network Address Translation functions in your firewall.

This IBM Knowledge Center shows the changes that you must make to the TCP/IP data sets in your z/OS distribution to gain that TCP/IP network connectivity. In the examples, it is assumed existing TCP/IP parmlib members and procedures are copied as new members into your USER.\* data sets, and modified. By using this method, the parmlib members and procedures of your original z/OS distribution are unmodified. Alternatively, you can change the existing TCP/IP procedures and parmlib members without creating new members.

TCP/IP and LAN configurations are site-dependent. The exact steps that are outlined here might not work at your site because of local network configuration, firewalls, Linux dependencies, or hardware restrictions. You might need the services of a network administrator to configure z/OS within your network.

### Setting up Linux routing

Before you configure TCP/IP, 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 if both the z/OS and Linux address are in the same subnet. The z/OS environment is 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 that is using the address 10.1.1.1. The connection between z/OS and Linux is called a tunnel.

**Note:** You can use the `--tunnel_ip` parameter in the `awsosa` stanza to define a local address other than 10.1.1.1. For example, if you define the local address as 10.1.2.1, then define the z/OS environment as 10.1.2.2.

If you use the `create_devmap.pl` program that is described in “Sample program to create device map” on page 105, and you have a z/OS host name that can be resolved to your static z/OS IP address by Linux, such as through a connected Domain Name Server or static configuration, you can add the `-h hostname` parameter to the invocation of the program. This parameter causes

create\_devmap.pl to generate file contents that are tailored to your network. It generates the TCPIP.DATA and TCPIP.PROFILE statements by using the z/OS IP address that corresponds to the entered hostname, and dynamically finds the name server addresses based on the Linux ethernet adapter that is found by using find\_io. It also generates sample VTAMLST members to define two OSAs; a tunnel OSA and an OSA to be used with one of the Linux ethernet adapters. The contents are shown as comments in the generated device map and are examples only. Because of the variability and complexity of network configuration, the samples that are shown in the generated device map might need modification to work in your environment.

This example assumes that you have the following OSA definitions in your devmap to define the tunnel OSA and one ethernet OSA to be used by TCP/IP:

```
[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
```

## Modifying TCPPARMS files

Learn how to modify two TCPPARMS files to establish TCP/IP connectivity by using an IP address that is defined for z/OS: PROFILE.TCPIP and TCPIP.DATA.

### PROFILE.TCPIP:

Copy your existing TCP/IP profile (TCPIP.PROFILE) to USER.TCPPARMS and modify the DEVICE, LINK, HOME, and route definitions to use the OSAs defined in your devmap, with the 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:** Comments and other statements were removed for brevity. Also, the IP address of the Linux system is not needed here. The gateway address usually ends in .1 or .0, but that might be different on your network.

This example also includes the definitions for the tunnel, the 10.1.1.2 address, which is used 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
 (((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.Z112S.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 port names are PORTA and PORTB. Also, verify that your devmap (see "Defining the device map" on page 103) 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.Z112S.VTAMLST(ATCCON00) to USER.VTAMLST(ATCCON00) and change the word OSATRL1 to OSATRL2.

### TCPIP.DATA:

Copy your existing TCPIP.DATA data set to USER.TCPPARMS and set the HOSTNAME, DOMAINORIGIN, and NSINTERADDR values

, as in this example:

```

TCPIPJOBNAME TCPIP
HOSTNAME RDT900
DOMAINORIGIN RTP.IBM.COM
DATASETPREFIX TCPIP
NSINTERADDR 9.0.0.1
NSINTERADDR 9.0.0.11
RESOLVEVIA UDP
LOOKUP LOCAL DNS
RESOLVERTIMEOUT 10
RESOLVERUDPREDRIES 1
ALWAYSUTO NO

```

If you choose a HOSTNAME or DOMAINORIGIN arbitrarily, be sure that the DOMAINORIGIN is not a real domain name and 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, can cause problems that are difficult to diagnose, such as timeouts, pauses, and connection failures in many areas, including 3270 connections and z Systems Development and Test Environment. Some systems, including components of z Systems Development and Test Environment, require that z/OS can locate itself by name.

Ensure that the NSINTERADDR parameter correctly identifies a DNS server. Incorrect name server specifications can also cause long delays and errors that are difficult to diagnose. Do not specify NSINTERADDR values for stand-alone systems.

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 *TCP/IP implementation volume 1: Base functions* (SG24-7798).

## Modifying TCP/IP Procedures to point to USER.TCPPARMS

An easy way to find Procedures that reference the TCPPARMS data sets by using ISPF is to display a member list of each proclib dataset in your active MSTJCLxx parmlib member (typically MSTJCL00), and then type these commands:

```
SRCHFOR TCPPARMS
SORT PROMPT
```

Copy any existing procedures that point to TCPIP.DATA data sets to USER.PROCLIB. Procedures that typically point to TCPIP.DATA include, but are not limited to,

```
FTPD
PORTMAP
TCPIP
TN3270
RESOLVER
```

Modify each member to change references from the existing TCPPARMS data set to USER.TCPPARMS for any members that you duplicated in USER.TCPPARMS. Do not change the member names except in the TCP/IP procedure as noted. For example, change the line in FTPD from //SYSTCPD DD DISP=SHR,DSN=CUST.TCPPARMS(TCPDATA) to:

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

Remember a modified TCP/IP profile member name was placed in USER.TCPPARMS, so the PROFILE DD statement is similar to this example:

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

If you prefer to set up system-wide use of a common set of TCP/IP settings, set up the RESOLVER procedure to point to your TCP/IP definitions. Create member USER.TCPPARMS(RESOLVER) to contain the following statements.

```
GLOBALTCPDATA('USER.TCPPARMS(TCPDATA)')
GLOBALIPNODES('USER.TCPPARMS(IPNODES)')
COMMONSEARCH
```

Create member USER.TCPPARMS(IPNODES) to contain the following statements similar to the following but customized with your network names and address.

The format of each entry in the IPNODES file is IP address followed by one or more host names. The order of the entries is not significant.

```
This first entry is the ip address of the z/OS host with its associated hostnames
In this example, two hostnames are configured; a shortened alias and a fully
 qualified hostname.
Only one hostname is required, typically the fully qualified name
#
9.12.200.0 ZDT ZDT.IBM.COM
#
This entry is the ip address of the OSA tunnel to the Linux host.
The hostname should be the hostname of the Linux operating system.
Multiple hostnames may be configured if needed
#
10.1.1.1 LINUX
#
Typically, 127.0.0.1 is configured as hostname LOCALHOST
#
127.0.0.1 LOCALHOST
```

Copy your existing RESOLVER procedure to USER.PROCLIB(RESOLVER), and change the SETUP DD statement from:

```
//SETUP DD DISP=SHR,DSN=CUST.TCPPARMS(yourresolverparms),FREE=CLOSE
```

To

```
//SETUP DD DISP=SHR,DSN=USER.TCPPARMS(RESOLVER),FREE=CLOSE
```

Add the following line to the BPXPRMxx PARMLIB members you are using with z Systems Development and Test Environment. This line ensures the RESOLVER procedure that you created is the one that is started.

```
RESOLVER_PROC(RESOLVER)
```

### Modifying VTAMLST members:

Create a VTAMLST member to define a TRL major node that matches the OSAs you are using. Member OSATRL2 in ADCD.\*.VTAMLST matches the sample devmap. If you create a new member to match your OSAs, add that member to USER.VTAMLST. To activate this major node when VTAM starts, add the VTAMLST member name to the VTAM configuration list (member ATCCONxx in VTAMLST) with which you start VTAM.

```
OSATRL1 VBUILD TYPE=TRL
OSATRL1E TRLE LNCTL=MPC,READ=(400),WRITE=(401),DATAPATH=(402), X
 PORTNAME=PORTA,MPCLEVEL=QDIO
OSATRL2E TRLE LNCTL=MPC,READ=(404),WRITE=(405),DATAPATH=(406), X
 PORTNAME=PORTB,MPCLEVEL=QDIO
```

### Enabling use of USER.PROCLIB and IPLing:

Ensure USER.PROCLIB (or whichever data set contains your customized proclibs) is in the concatenation of the IEFPSI DD statement in your active MSTJCLxx parmlib member.

Copy your existing active MSTJCLxx parmlib member to USER.PARMLIB, and make changes similar to those described in this sample:

```
//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=SYS1.PROCLIB,DISP=SHR
//SYSUADS DD DSN=SYS1.UADS,DISP=SHR
//SYSLBC DD DSN=SYS1.BROADCAST,DISP=SHR
```

Ensure USER.PARMLIB is in the PARMLIB statement concatenation of your active LOADxx member.

Also ensure USER.PROCLIB is in the concatenation of the proclibs in your procedure that starts JES. For JES2, that is the PROC00 DD statement. For JES3, that is the IATPLBST DD statement. Copy your active JES procedure into USER.PROCLIB, and make these changes.

```
//JES2 PROC MEMBER=JES2PARM,ALTMEM=JES2BACK
//IEFPROC EXEC PGM=HASJES20,DPRTY=(15,15),TIME=1440,PERFORM=9
//ALTPARM DD DSN=CUST.PARMLIB(&ALTMEM),DISP=SHR
//HASPPARM DD DSN=CUST.&SYSVER..PARMLIB(&MEMBER),DISP=SHR
//PROC00 DD DSN=USER.PROCLIB,DISP=SHR << MODIFIED
// DD DSN=CEE.SCEEPROC,DISP=SHR
// DD DSN=CSQ710.SCSQPROC,DISP=SHR
// DD DSN=IOE.SIOEPROC,DISP=SHR
// DD DSN=EOY.SEOYPROC,DISP=SHR
// DD DSN=HLA.SASMSAM1,DISP=SHR
// DD DSN=CBC.SCCNPRC,DISP=SHR
// DD DSN=SYS1.PROCLIB,DISP=SHR
//HASPLIST DD DDNAME=IEFRDER
```

Perform an IPL of the system and use the loadparm that you want or your newly created loadparm, whichever is appropriate, to verify that your changes are working.

## Optional tasks

Learn about optional configuration tasks for z Systems Development and Test Environment.

### Logging on to TSO from emulators not native to the host Linux

After TCP/IP connectivity is established, you can use your favorite 3270 emulator software, such as IBM Personal Communications Manager (PCOMM), to connect to the non-SNA (coax) 3270 device emulator that is provided by z Systems Development and Test Environment. When you connect from outside the Linux system that is hosting z Systems Development and Test Environment, use the Linux TCPIP address and 3270 as port. (The actual port number is defined in the device map.)

### Creating TSO user IDs

TSO user IDs are created through a series of commands. Typically, you want to use user IDs that have OMVS segments and an associated z/OS UNIX user directory when you are running Developer for z Systems.

The example commands that are shown here create a user ID in the RDZUSERS group with an OMVS segment. Replace *#userid*, *#name*, and *#password* with appropriate values, and do not remove the quotation marks in the commands.

From a CLIST, REXX exec, or TSO command line, enter the following commands. The commands create the user ID, provide an OMVS segment, and assign an account number, default log-on procedure, and region size. They also protect data sets with a high-level qualifier that belongs to the user from being accessed by other users. Finally, the commands create an alias in the master catalog to indicate



that the users data sets are cataloged in the user catalog on volume S1CUST. Be sure that the substitution in the HOME( ) and PROGRAM() parameters are in lowercase.

```
ADDGROUP RDZUSERS OMVS(AUTOGID)
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) COMMAND('ISPF'))
ADDSO '#userid.*' UACC(NONE)
PERMIT ACCT# CLASS(ACCTNUM) ID(#userid) ACCESS(READ)
DEFINE ALIAS (NAME('#userid') RELATE('USERCAT.S1CUST'))
```

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 lowercase.

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

## Defining a new logon procedure

A common customization in z/OS systems is to alter the logon procedure that TSO users use. Do 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 your 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(S1CFG1)'
```

To allow all users to use the TSOLOGON procedure, enter the following TSO commands.

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

## Altering system startup and shutdown scripts

Most z/OS distributions have a shutdown script that contains a series of commands to shut down all active subsystems. Your system administrator knows the name and location of these scripts.

You might want to make these changes to these scripts.

The shutdown scripts stop OMVS automatically. Explicitly stopping ZFS can expedite the shutdown process, but requires a response to a prompt on the operator console. If you want to stop ZFS before you shut down OMVS, add

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

Before

```
F OMVS,SHUTDOWN
```

Some startup and shutdown scripts have PAUSE commands that you might find are too long or too short for your system. Many can be safely changed to pause for a shorter amount of time.

You can change startup scripts the same way. Other changes might include *not* starting particular subsystems, changing pause times, and so forth.

## Defining JES NJE connectivity

Your z/OS distribution can be a stand-alone 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 use the z Systems Development and Test Environment 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, which are run on the z Systems Development and Test Environment system, name the local system RUT0 and define a connection to M168.

```
$TNODE(Z21S),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
```

Give similar operator commands on the M168 system to complete the setup. Since this system is an existing system, the command to define the local node name was skipped. The following commands also assume that 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=RDz8500.RTP.IBM.COM
$SN,SOCKET=REMOTE
```

The z Systems Development and Test Environment system can use the existing NJE definitions on the M168 system to connect to other NJE nodes in your network. Enter the following operator commands on the z Systems Development and Test Environment 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
```

---

## Customizing the May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment

Learn how to customize the z/OS 2.2 Application Developers Controlled Distribution for IBM z Systems Development and Test Environment May 2017 edition.

This z/OS 2.2 distribution is a unique distribution that was created specifically for sites that are entitled to IBM z Systems Development and Test Environment, based on the Application Developers Controlled Distribution (ADCD). It is referred to in this IBM Knowledge Center as the z/OS 2.2 ADCD. It contains many of the products and subsystems you expect in a z/OS system. Most of these products and subsystems run without any customization, and some need some customization or initialization to make them work. These customizations are centered on changes that are needed to establish TCP/IP connectivity and update the infrastructure to allow z/OS 2.2 to properly run on IBM z Systems Development and Test Environment.

**Restriction:** Any z/OS ADCD for z Systems Development and Test Environment, and any related software, are supplied as a convenience. IBM does not support the operating system distributions that are made available with z Systems Development and Test Environment. You can apply fixes to any z/OS distribution through normal SMP/E facilities.

Before customizing this z/OS 2.2 ADCD, you must first install IBM z Systems Development and Test Environment, and download and install the z/OS 2.2 ADCD. For more information about how to download ADCD images and their checksum, see IBM z Systems Development and Test Environment e-assemblies to download from Passport Advantage. Then, to customize the z/OS 2.2 ADCD distribution, do the steps that are described in the next three topics. The last two topics contain reference information about the z/OS 2.2 ADCD distribution.

## Defining the device map for z/OS 2.2 ADCD

The IBM z Systems Development and Test Environment allows the customization of the z Systems resources available within the virtualized environment. The resources can be defined in a device map or *devmap*.

The sections and syntax of device maps are explained in detail in the "1090 Control Files" section of the zPDT Guide and Reference.

The system volumes that represent the z/OS distribution are defined in an *awsckd* stanza, and are mounted at arbitrarily chosen addresses within the range of valid addresses that are defined by the z/OS distribution's IODF. Historically in a z/OS ADCD, the *xxRES1* and *xxSYS1* volumes are mounted at the same addresses that are documented in the zPDT Guide and Reference, and address 0AA3 is always reserved for volume *xxDBAR*.

The following examples, when combined, define a working device map with this z/OS ADCD for IBM z Systems Development and Test Environment. If you are using the coupling facility or Rational Tokens, see "Enabling the coupling facility" on page 184 and "Setting up Rational Tokens" on page 205.

```
[system]
memory 4096m
processors 3
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 tsol
```

For the network adapter definitions, the following example was added. One OSA is defined for connectivity to the host Linux system, called a "tunnel" OSA, and one is defined for establishing remote connectivity by using an ethernet adapter on the Linux host computer. Your setup might differ. For more information about configuring OSAs, see "Configuring a working z/OS system" on page 129 and the sections on connectivity in the zPDT Guide and Reference.

```
[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 volumes are in the `/home/ibmsys1/z1090/disks/` directory, the DASD definitions might look like the following example:

```

[manager]
name awsckd 0002
device 0A80 3390 3390 /home/ibmsys1/z1090/disks/xxRES1
device 0A81 3390 3390 /home/ibmsys1/z1090/disks/xxRES2
device 0A82 3390 3390 /home/ibmsys1/z1090/disks/xxSYS1
device 0A83 3390 3390 /home/ibmsys1/z1090/disks/xxCFG1
device 0A84 3390 3390 /home/ibmsys1/z1090/disks/xxUSS1
device 0A85 3390 3390 /home/ibmsys1/z1090/disks/xxUSS2
device 0A86 3390 3390 /home/ibmsys1/z1090/disks/xxPRD1
device 0A87 3390 3390 /home/ibmsys1/z1090/disks/xxPRD2
device 0A88 3390 3390 /home/ibmsys1/z1090/disks/xxPRD3
device 0A89 3390 3390 /home/ibmsys1/z1090/disks/xxDIS1
device 0A8A 3390 3390 /home/ibmsys1/z1090/disks/xxDIS2
device 0A8B 3390 3390 /home/ibmsys1/z1090/disks/xxPAGA
device 0A8C 3390 3390 /home/ibmsys1/z1090/disks/xxPAGB
device 0A8D 3390 3390 /home/ibmsys1/z1090/disks/xxPAGC
device 0A8E 3390 3390 /home/ibmsys1/z1090/disks/xxBLZ1
device 0A8F 3390 3390 /home/ibmsys1/z1090/disks/xxDBB1
device 0A90 3390 3390 /home/ibmsys1/z1090/disks/xxDBB2
device 0A91 3390 3390 /home/ibmsys1/z1090/disks/xxDBC1
device 0A92 3390 3390 /home/ibmsys1/z1090/disks/xxDBC2
device 0A93 3390 3390 /home/ibmsys1/z1090/disks/xxDBAR
device 0A94 3390 3390 /home/ibmsys1/z1090/disks/xxC521
device 0A95 3390 3390 /home/ibmsys1/z1090/disks/xxC531
device 0A96 3390 3390 /home/ibmsys1/z1090/disks/xxIMD1
device 0A97 3390 3390 /home/ibmsys1/z1090/disks/xxIME1
device 0A98 3390 3390 /home/ibmsys1/z1090/disks/xxINM1
device 0A99 3390 3390 /home/ibmsys1/z1090/disks/xxW851
device 0A9A 3390 3390 /home/ibmsys1/z1090/disks/xxW852
device 0A9B 3390 3390 /home/ibmsys1/z1090/disks/xxW853
device 0A9C 3390 3390 /home/ibmsys1/z1090/disks/xxW854
device 0A9D 3390 3390 /home/ibmsys1/z1090/disks/xxW855
device 0A9E 3390 3390 /home/ibmsys1/z1090/disks/xxW901
device 0A9F 3390 3390 /home/ibmsys1/z1090/disks/xxKAN1
device 0AA1 3390 3390 /home/ibmsys1/z1090/disks/xxW902
device 0AA2 3390 3390 /home/ibmsys1/z1090/disks/SARES1
device 0AA3 3390 3390 # Available for dynamic mounts
device 0AA4 3390 3390 # Available for dynamic mounts
device 0AA5 3390 3390 # Available for dynamic mounts

```

*Figure 4. Device map example. Device map example*

You can verify the device map with the `awsckmap` command. All of the disk volume images that are referenced in the device map must exist before you verify the device map. Assume that the device map file has the following name:

```
/home/ibmsys1/z/myDeviceMap
```

With the name in the preceding example, the device map can be verified with the command:

```
awsckmap /home/ibmsys1/z/myDeviceMap
```

### Sample program to create device map

A sample program that is called `create_devmap.pl` is available in the `ConfigGuideSample` directory where you installed z Systems Development and Test Environment. If Perl is installed on your Linux system, you can use the

create\_devmap.pl program to generate a sample device map that is based on your current decompressed 3390 disk images, memory configuration, and available network parameters.

Consider the output of create\_devmap.pl to be a starting point from which you can create a final device map. The syntax for the create\_devmap.pl command is shown here:

```
perl <pathtocommand>/create_devmap.pl pathtodisks > generateddevmap
```

In the preceding example, *pathtocommand* is the location of the create\_devmap.pl file, and *pathtodisks* is the location of your 3390 disk images. *generateddevmap* is the name of the file to contain the new device map.

If you already have a static IP address that is assigned for your virtual z/OS machine and a z/OS host name that can be resolved to that address by Linux, such as through a connected Domain Name Server or static configuration, you can add the -h <hostname> parameters after the pathtodisks parameter. Adding the parameter causes the script to attempt to generate comments that contain more accurate TCP/IP configuration samples based on your network.

```
perl <pathtocommand>/create_devmap.pl pathtodisks -h
hostname_of_zos > generateddevmap
```

The create\_devmap.pl program creates a memory line based on existing hardware and configuration of your Linux machine. Verify that the amount of memory that is requested is appropriate for your situation.

The device map that is created by create\_devmap.pl defines OSA devices based on the first Tun/Tap and Wired CHPIDs found that use the **find\_io** command, and a set of sample z/OS TCP/IP definitions that would correspond to the OSA device definitions in the generated device map. These TCP/IP configuration statements can be used as a starting point for your TCP/IP configuration, but probably require changes to match your network. Verify that the device addresses and device names in your final VTAM definitions, TCP/IP profile, and device map all correspond to the correct network adapter types.

The device map that is created by create\_devmap.pl also contains 3390 device statements for files in the *pathtodisks* directory that are over 800 MB, have six-character names, and are verified to be disk images by the alckd command.

## Starting and stopping the z/OS 2.2 AD CD

Learn how to start and stop the z/OS 2.2 AD CD.

z Systems Development and Test Environment is started and stopped, and the z/OS system is IPLed, by using commands that are entered on a Linux console that is running under the user ID that is configured to run z Systems Development and Test Environment.

### Start the z Systems Development and Test Environment

The sample runzpdtd script available in the ConfigGuideSample directory can be used to start z Systems Development and Test Environment with any z/OS distribution.

This script is described in “Starting the z Systems Development and Test Environment” on page 107.

When you run this z/OS 2.2 ADCD, the load parameter that is specified with the -l switch (the -l is a lowercase L) must correspond to a valid load parameter defined in the distribution. For information about the load parameters available with this distribution, and which major subsystems they start, see “z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 143.

**Note:** Each loadparm in this z/OS ADCD has an associated parmlib member that is named COMMNDxx, where xx corresponds to a LOADxx loadparm. That parmlib member defines all of the subsystems that are started when that loadparm is IPLed. The COMMNDxx parmlib member runs an SVTAMxx procedure, which uses a VTAMxx parmlib member to start another set of subsystems. You can see all of the subsystems that are started for all loadparms by looking at the COMMNDxx and VTAMxx parmlib members in ADCD.\*.PARMLIB.

The first time this z/OS distribution is started, start z Systems Development and Test Environment with the CS loadparm. This loadparm initiates a cold start, and does not start other subsystems. After you verify basic z/OS capabilities, you can then customize the system by using the instructions in “Configuring a working z/OS system” on page 129 and create and start different loadparms as needed. As with any z/OS system, warm starts are less disruptive and preserve the JES job spool. Initiate a warm start when possible.

When you perform an IPL of the supplied z/OS distribution system for the first time, you might see some errors during the IPL process. For example, SMF files might be full. If you are prompted with the IXC420D REPLY I TO INITIALIZE SYSPLEX ADCDPL, . . . message, respond with R 0,I, that is, the letter R, a blank, a zero, a comma, and the letter I.

## Stopping the z Systems Development and Test Environment

If possible, always shut down z/OS cleanly. Typically, shutting down cleanly begins by logging off all TSO users, and starting a procedure that shuts down all active subsystems.

This z/OS 2.2 ADCD contains shutdown scripts for most systems available and active in the distribution. They are named SHUTxx, where xx corresponds to the VTAMxx parmlib member that was used to start the subsystems for a particular IPL.

**Note:** Use SHUT00 when the z/OS 2.2 ADCD was started with loadparm CS or WS, and SHUTALL when the z/OS 2.2 ADCD was started with loadparm AL. When an **S SHUTxx** command is issued from the MVS master console, these scripts shut down all systems that were automatically started by the corresponding VTAMxx parmlib member. These shutdown scripts are contained in ADCD.\*.PROCLIB, where \* varies according to the release started.

While the shutdown scripts attempt to stop all active subsystems started by that loadparm, some subsystems might require further input to be stopped. You might have to respond to z/OS console messages, such as when IMS, TSO, or z/OS UNIX are stopped. You can see what programs are still running by entering the D J,L console command. Ensure VTAM and all subsystems are stopped.

After you run the appropriate SHUTxx script and all systems are stopped, stop your JES system. The loadparms in your z/OS 2.2 ADCD distribution always start JES2. A **\$pjes2** command that is entered from the MVS console stops JES2.

After JES ends, z Systems operation can be stopped by entering the **QUIESCE** command. After this command completes, stop the zPDT system by entering this command in the Linux window:

```
$ awsstop
```

This command produces several messages. It might be necessary to press **Enter** to obtain the Linux prompt. Any 3270 windows can be closed.

## Configuring a working z/OS system

Learn how to configure a working z/OS 2.2 ADCD system for z Systems Development and Test Environment.

These topics guide you through procedures to help you make customizations to a z/OS 2.2 ADCD. These customizations include establishing TCP/IP connectivity to your z/OS system, increasing your /tmp and /u directories, and optional tasks such as creating new TSO user IDs.

### 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 data sets 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 often see references to LOAD*xx*, IEASYS*xx*, and so on. Configuration files refer to other members by a keyword and suffix number. For example, a member that is called IEASYSRC might identify the main z/OS UNIX configuration file with the line OMVS=RC, which means that z/OS UNIX finds the member that contains its configuration parameters by starting with a predetermined name, BPXPRM, and add the suffix RC (resulting in member name of BPXPRMRC). The keywords in the configuration files *are not* usually the same as the member name prefixes.

**Important:** Do not directly update data sets such as SYS1.PARMLIB and SYS1.PROCLIB. Some installation-specific libraries are searched before the SYS1 data sets, leaving the SYS1 libraries with IBM supplied defaults.

This z/OS 2.2 ADCD defines several sets of alternative libraries. The configurations for the distribution itself are stored in a set of libraries that start with the qualifiers ADCD.\*. The set of libraries that are used to store z Systems Development and Test Environment configurations start with the qualifiers FEU.\*. This z/OS distribution also provides a set of libraries for you to use, which start with the high-level qualifier USER. This z/OS distribution is already set up to read from most of the USER libraries before the FEU.\* libraries and the ADCD.\* libraries. The USER.\* libraries are initially empty, and you can use them for your own changes. They are on a separate volume, D2CFG1. Avoid updating the ADCD.\* and SYS1.\* libraries whenever possible. If you choose to place data in the USER data sets, or create other user data sets or customizations in xxCFG1, these data sets can be easily migrated to a new system. Since volume names change with each release of a z/OS ADCD for z Systems Development and Test Environment, add the xxCFG1 volume to the devmap for the new system, and then copy any customized files back to the newer xxCFG1 volume.

The customizations described here use USER.\* libraries, which tend to be commonly available in any z/OS distribution. While you follow these instructions, all changes are made in USER.\* libraries, except for adding LOADPARM members to SYS1.IPLPARM.

**Note:** Subsystem configuration changes, such as CICS definitions, also change original libraries such as the CICS CSD.

#### **The IPL process: LOADxx and IEASYSxx members:**

When z/OS is started, it looks in data set SYS1.IPLPARM for a member called LOADxx. The xx value is specified in the IPL statement of the start script that was created. A LOADxx member defines various settings to start the system, such as the parmlib concatenation that indicates which data sets 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:** Create a loadparm that starts the system without pointing to any alternative libraries that contain customized parmlibs, proclibs, and so on. By creating a loadparm if you make a mistake in your customizations that causes z/OS to not start, you can try to perform an IPL of the system with the loadparm that does not point to the alternative libraries. This method gives you TSO access, where you can modify the customizations to correct any mistakes.

If you make a mistake that causes z/OS not to start, you can try to perform an IPL of the system with CS or WS as the last two digits of the LOADPARM. These suffixes start the system with a simpler configuration. CS initiates a cold start, which clears the JES spool, and WS initiates a warm start. For example, to use a LOADPARM of CS, change the IPL command in your startup script to **ipl 0a80 parm 0a82cs**.

In an emergency, you can perform an IPL of the stand-alone system, which provides a basic system, but enables you to change your configuration data sets, because it does not share anything with the normal z/OS setup. The stand-alone system does not contain TCP/IP or Unix system service support. The system cannot be used to install products or apply maintenance through SMP/E. When you use the stand-alone system, you must specify the volume name of data sets you want to edit, since all volumes other than the residence volume for the stand-alone system are not cataloged. Do not edit the stand-alone system configuration. You can perform an IPL of the stand-alone system by mounting volume SARES1 and replacing the IPL statement with **ipl 0A98 parm 0A98SA**. Replace **0A98** with the address of volume SARES1. Use **S SHUTSA** to shut down the SARES1 system.

#### **PROCLIB: System procedure library:**

Parmlib members contain only configuration information. The procedures that start the various subsystems and servers are found in a different concatenation called PROCLIB. As an example, this z/OS 2.2 ADCD contains SYS1.PROCLIB, ADCD.\*.PROCLIB, FEU.\*.PROCLIB, and USER.PROCLIB in its PROCLIB concatenation. To include any PROCLIB data set in the PROCLIB concatenation, you must modify the MSTJCLxx member of parmlib and your JES procedure.



For more information, see “Modifying TCP/IP procedures to point to USER.TCPPARMS” on page 138 and “Enabling use of USER.PROCLIB and IPLing” on page 121.

## Configuration overview

Learn how to configure the z/OS system so that you isolate the customized data in your z/OS distribution volumes and establish TCP/IP communications.

The following steps are described:

- Create new z/OS UNIX file systems for /tmp and /u.
- Customize TCP/IP settings to establish network connectivity.

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

- Create TSO user IDs.
- 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

Place these customizations on the single volume D2CFG1 volume that already contains your USER.\* libraries. Placing these customizations on a single volume helps with future migrations to new versions of z/OS ADCD distributions.

## Logging on to TSO

After you IPL, you can use the x3270 emulator on the host Linux to start a TSO session.

Enter this command:

```
x3270 -port 3270 localhost &
```

This example shows an alternative format of the x3270 command, which produces a larger screen size:

```
x3270 -model 4 localhost:3270 &
```

You can also use an emulator that is not native to the host Linux. For more information, see “Logging on to TSO from emulators not native to the host Linux” on page 141.

Log on to TSO with a valid user ID in your z/OS distribution. For this distribution, log on with the IBMUSER account, with initial password IBMUSER. You are prompted to change the password. This account has the administrative RACF® privileges that you need for certain customizations. You might want to make a few minor changes to your ISPF 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, enter the command

```
TSO PROFILE PREFIX(tso user id)
```

## Creating new zFS 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 and alternative mount points might require a more in-depth plan.

This z/OS ADCD provides fairly small file systems for the /tmp and /u directories. Small file systems can cause problems, particularly when you are installing z/OS

software by using SMP/E, or when programs create large memory dumps on the z/OS UNIX file system. Allocate 100 primary cylinders and 20 secondary cylinders for these file systems. If the current file systems for /tmp and /u are not large enough, replace them. To create new file systems for /tmp and /u, create two new zFS files and associate these new files with the /tmp and /u directories. Catalog these two new zFS files on the xxCFG1 volume with your other customization. Since a user catalog named USERCAT.Z22x.USER is on the xxCFG1 volume of the z/OS 2.2 ADCD as shipped, it can be used for cataloging the two new zFS files. By avoiding use of the master catalog, during migration to a new z/OS ADCD, you then need only to import the user catalog and replicate the alias definitions so that catalog entries are restored. Use this procedure for creating new zFS files for /tmp and /u:

- Create aliases for the high-level qualifiers of the zFS files
- Create the zFS files
- Modify the appropriate BPXPRMxx parmlib member to mount the new zFS files with the /tmp and /u directories
- Copy the existing /tmp and /u directories to the new /tmp and /u directories that are mounted with the new zFS files
- Optionally create a new load parm to contain these changes
- Shut down the system and reIPL

The next topics describe the procedures for each step, including sample JCL where appropriate. In these procedures, the two z/OS UNIX file systems are created with a high-level qualifier of CUST on volume xxCFG1.

### Creating ALIASes for the high-level qualifiers of the zFS files:

Start by creating an alias for CUST and any other qualifiers that are used to create data sets here. Create the alias and the file systems on a volume with your other customizations so that it is easy to migrate to a new z/OS distribution.

When you create new users, also create an alias for the user ID in the same catalog. This sample JCL shows you how.

```
//Q12ALIAS JOB (ACCT),MSGCLASS=H,NOTIFY=&SYSUID.
/*-----
//DEFALIAS EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 DEFINE ALIAS (NAME(CUST) RELATE(USERCAT.Z22C.USER))
/*
```

### Creating the zFS files:

Use the JCL in this topic to create two new zFS files on volume xxCFG1 named CUST.ZFS.U and CUST.ZFS.TMP.

Adjust the space allocations as needed.

```
//Q13ZFS JOB 'IBMUSER',CLASS=A,NOTIFY=&SYSUID,MSGCLASS=H
//ZFSALLOC EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
 DEFINE CLUSTER(-
 NAME(CUST.ZFS.TMP) -
 VOLUME(xxCFG1) -
 LINEAR -
 CYL(100 20) -
 SHAREOPTIONS(3) -
```

```

)
 DEFINE CLUSTER(-
 NAME(CUST.ZFS.U) -
 VOLUME(xxCFG1) -
 LINEAR -
 CYL(100 20) -
 SHAREOPTIONS(3) -
)
/*
//ZFSFORMT EXEC PGM=IOEAGFMT,REGION=0M,COND=(0,LT,ZFSALLOC),
// PARM='-aggregate CUST.ZFS.TMP -compat'
//SYSPRINT DD SYSOUT=*
//ZFSFORMT EXEC PGM=IOEAGFMT,REGION=0M,COND=(0,LT,ZFSALLOC),
// PARM='-aggregate CUST.ZFS.U -compat'
//SYSPRINT DD SYSOUT=*

```

### Modifying BPXPRM00 parmlib member:

After you create CUST.ZFS.TMP and CUST.ZFS.U, copy ADCD.Z22x.PARMLIB(BPXPRM00) into USER.PARMLIB.

Comment out the existing mount of the zFS for tmp, and replace it with a mount of CUST.ZFS.TMP. For example,

```

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

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

```

Make a similar change for the mount of /u:

```

/* MOUNT FILESYSTEM('ZFS.&ADCDLVL..USERS') */
/* TYPE(HFS) */
/* MODE(RDWR) NOAUTOMOVE */
/* MOUNTPOINT('/u') */

MOUNT FILESYSTEM('CUST.ZFS.U')
 TYPE(ZFS)
 MODE(RDWR) NOAUTOMOVE
 MOUNTPOINT('/u')

```

X : Replace Wild Character with current ADCD version character.

### Copying the existing /tmp and /u directories:

Finally, you must copy the existing /tmp and /u directories to contain the new /tmp and /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 these commands. Be sure to verify that each command works correctly before you proceed to the next command.

```

cd /
mkdir /tempmnt
/usr/sbin/mount -f cust.zfs.tmp /tempmnt
cp -r /tmp/ /tempmnt
ls /tempmnt
chmod 777 /tempmnt

```

```

/usr/sbin/unmount /tempmnt
/usr/sbin/mount -f cust.zfs.u /tempmnt
cp -r /u/ /tempmnt
ls /tempmnt # to verify that the copy worked
chmod 777 /tempmnt
/usr/sbin/unmount /tempmnt
rm -r /tempmnt

```

### Creating a loadparm for customizations:

You might want to isolate your customizations into a new loadparm. For example, you can create an RC loadparm that uses your new zFS files for /tmp and /u.

- Instead of modifying BPXPRM00, copy it and rename it to BPXPRMRC, and modify BPXPRMRC.
- Copy the IEASYSxx member that currently starts the subsystems you want to start to USER.PARMLIB. Rename it to IEASYSRC to match the two character loadparm of BPXPRMRC. Update the **OMVS=** parameter to replace the existing two character loadparm with RC.
- Make a copy of the existing LOADxx member in SYS1.IPLPARM that currently starts the subsystems you want to start. Rename it to LOADRC, and store it back into SYS1.IPLPARM. Modify the **SYSPARM** value to RC.

### Shutting down the system and re-IPLing:

Shut down the system and perform an IPL. Use your loadparm that points to the modified BPXPRMxx, or your newly created load parm.

When the system restarts, you can enter the z/OS UNIX command line and enter the **df -k** command to ensure that your file systems are being used and mounted correctly.

After the shutdown and re-IPL is complete, you have larger zFS files for /tmp and /u.

### Setting up TCP/IP

z/OS running on z Systems Development and Test Environment can communicate with your network by using TCP/IP. You can use standard 3270 terminal emulators, FTP, Developer for z Systems, and other services to move data to and from your z/OS system.

Since mainframes are 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 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 zPDT Guide and Reference.

This example uses the method referred to as Scenario 4 in Chapter 7, “LANs” in the zPDT Guide and Reference. This method allows the z/OS system and the Linux system to communicate with your network, each with their own IP address and with the ability to share the ethernet adapter. By each having their own IP address, both systems can simultaneously use functions such as FTP with minimal configuration change. It also eliminates the need to create firewall *holes* for your z/OS ports or provide Network Address Translation functions in your firewall.

The examples show the changes that you must make to the TCP/IP data sets in your z/OS distribution to gain that TCP/IP network connectivity. The examples assume that existing TCP/IP parmlib members and procedures are copied as new

members into your USER.\* data sets, and modified. By using this method, the parmlib members and procedures of your original z/OS distribution are unmodified. Alternatively, you can change the existing TCP/IP procedures and parmlib members without creating new members.

TCP/IP and LAN configuration is site-dependent. The exact steps that are outlined here might not work at your site because of local network configuration, firewalls, Linux dependencies, or hardware restrictions. You might need the services of a network administrator to configure z/OS within your network.

### **z/OS 2.2 ADCD network configuration:**

Learn how to change configuration work in your network by either defining OSA as a tunnel or defining OSA to share a Linux Ethernet adapter.

The supplied z/OS 2.2 ADCD by default activates member PROF1 in ADCD.Z22D.TCPPARMS as the TCPIP.PROFILE. It activates one OSA with device name ADM1ETP, an IP address of 192.168.0.61, and a default route to 192.168.0.1. You need to change this configuration work in your network by either defining this OSA as a tunnel or defining this OSA to share a Linux Ethernet adapter, are explained in the "LANs" chapter of zPDT Guide and Reference. In either case, with only one OSA activated, you have connectivity restrictions in this configuration. If you define an external OSA without a tunnel OSA, you cannot communicate to the Linux host. If you define a tunnel OSA without an external OSA, all traffic must come through the Linux host, and is subject to Linux routing restrictions, firewalls, and considerations when both Linux and z/OS use the same ports.

### **Setting up Linux routing:**

Before you configure TCP/IP, 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 if both the z/OS and Linux address are in the same subnet. This z/OS environment is 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 that is using the address 10.1.1.1. The connection between z/OS and Linux is called a tunnel.

**Note:** You can use the `--tunnel_ip` parameter in the `awsosa` stanza to define a local address other than 10.1.1.1. In the customizations below, it is assumed the `--tunnel_ip` parameter is not defined and defaults to 10.1.1.1.

If you use the `create_devmap.pl` program that is described in "Sample program to create device map" on page 126, and you have a z/OS host name that can be resolved to your static z/OS IP address by Linux, such as through a connected Domain Name Server or static configuration, you can add the `-h hostname` parameter to the invocation of the program. This parameter causes `create_devmap.pl` to generate sample contents of z/OS host configuration files that are tailored to your network. It generates the `TCPIP.DATA` and `TCPIP.PROFILE` statements by using the z/OS IP address that corresponds to the entered hostname, and dynamically finds the name server addresses based on the Linux ethernet adapter that is found by using `find_io`. It also generates sample `VTAMLST` members to define two OSAs; a tunnel OSA and an OSA to be used with one of

the Linux ethernet adapters. The contents are shown as comments at the end of the generated device map and are examples only. Because of the variability and complexity of network configuration, the samples that are shown in the generated device map might need modification to work in your environment.

This example assumes that you have the following OSA definitions in your devmap to define the tunnel OSA and one ethernet OSA to be used by TCP/IP:

```
[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
```

### Modifying TCPPARMS files:

Learn how to modify two TCPPARMS files to establish TCP/IP connectivity by using an IP address that is defined for z/OS: PROFILE.TCPIP and TCPIP.DATA.

#### PROFILE.TCPIP:

Copy ADCD.Z22D.TCPPARMS(PROF2) to USER.TCPPARMS(PROFILE) and modify the DEVICE, LINK, HOME, and route definitions to use the OSAs defined in your devmap, with the 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:** Comments and other statements were removed for brevity. Also, the IP address of the Linux system is not needed here. The gateway address usually ends in .1 or .0, but that might be different on your network.

This example also includes the definitions for the tunnel, the 10.1.1.2 address, which is used 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
 (((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.Z112S.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 port names are PORTA and PORTB. Also, verify that your devmap correctly defines the device addresses in the READ, WRITE, and DATAPATH statements of USER.VTAMLST(OSATRL2). For more information about the device map, see “Defining the device map for z/OS 2.2 ADCD” on page 125.

```

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.Z112S.VTAMLST(ATCCON00) to USER.VTAMLST(ATCCON00) and change the word OSATRL1 to OSATRL2.

*TCPIP.DATA:*

Copy ADCD.Z22D.TCPPARMS(TCPDATA) to USER.TCPPARMS and set the HOSTNAME, DOMAINORIGIN, and NSINTERADDR values.

```

TCPIPJOBNAME TCPIP
 HOSTNAME RDT900
 DOMAINORIGIN RTP.IBM.COM
 DATASETPREFIX TCPIP
 NSINTERADDR 9.0.0.1
 NSINTERADDR 9.0.0.11
 RESOLVEVIA UDP
 LOOKUP LOCAL DNS
 RESOLVERTIMEOUT 10
 RESOLVERUDPRETRIES 1
 ALWAYSWHO NO

```

The z/OS 2.2 ADCD ships with a HOSTNAME of S0W1 and a DOMAINORIGIN of DAL-EBIS.IHOST.COM. These work with the shipped WebSphere Application Server configuration. If you chose to define a different HOSTNAME and DOMAINORIGIN, and used the create\_devmap.pl program with the -h option and your chosen HOSTNAME to create your devmap, the output devmap will contain the customized TCPIP.DATA parameters for HOSTNAME and DOMAINORIGIN.

If you choose a HOSTNAME or DOMAINORIGIN arbitrarily, be sure that the DOMAINORIGIN is not a real domain name and 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, can cause problems that are difficult to diagnose, such as timeouts, pauses, and connection failures in many areas, including 3270 connections and z Systems Development and Test Environment. Some systems, including components of z Systems Development and Test Environment, require that z/OS can locate itself by name.

Ensure that the NSINTERADDR parameter correctly identifies a DNS server. Incorrect name server specifications can also cause long delays and errors that are difficult to diagnose. Stand-alone systems should not specify NSINTERADDR values.

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 *TCP/IP implementation volume 1: Base functions* (SG24-7798).

### Modifying TCP/IP procedures to point to USER.TCPPARMS:

An easy way to use ISPF to find Procedures that reference the TCPPARMS data sets is to display a member list of each proclib data set in your active MSTJCLxx parmlib member (typically MSTJCL00), and then type these commands:

```
SRCHFOR TCPPARMS
SORT PROMPT
```

Copy the following members from ADCD.Z22D.PROCLIB to USER.PROCLIB. Be careful not to replace any members you already changed in USER.PROCLIB.

```
FTPD
PORTMAP
TCPIP
RESOLVER
NFSC
NFSS
```

Modify each member to change references from the existing TCPPARMS data set to USER.TCPPARMS for any members that you duplicated in USER.TCPPARMS. Do not change the member names except as noted in the TCP/IP procedure.

For example, locate this line in FTPD:

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

Change it to

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

Remember a modified TCP/IP profile member name was placed in USER.TCPPARMS, so the PROFILE DD statement is similar to this example:

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



If you prefer to set up system-wide use of a common set of TCP/IP settings, set up the RESOLVER procedure to point to your TCP/IP definitions. Create member USER.TCPPARMS(RESOLVER) to contain the following statements.

```
GLOBALTCPDATA('USER.TCPPARMS(TCPDATA)')
GLOBALIPNODES('USER.TCPPARMS(IPNODES)')
COMMONSEARCH
```

Create member USER.TCPPARMS(IPNODES) to contain the following statements similar to the following but customized with your network names and address. If you used the create\_devmap.pl program with the **-h** option to create your devmap, the output devmap contains the customized IPNODES file. The format of each entry in the IPNODES file is IP address followed by one or more host names. The order of the entries is not significant.

```
This first entry is the ip address of the z/OS host with its associated hostnames
In this example, two hostnames are configured; a shortened alias and a fully
qualified hostname.
Only one hostname is required, typically the fully qualified name
#
9.12.200.0 RDT900 RDT90.IBM.COM
#
This entry is the ip address of the OSA tunnel to the Linux host.
The hostname should be the hostname of the Linux operating system.
Multiple hostnames may be configured if needed
#
10.1.1.1 LINUX
#
Typically, 127.0.0.1 is configured as hostname LOCALHOST
#
127.0.0.1 LOCALHOST
```

Copy ADCD.Z22D.PROCLIB(RESOLVER) to USER.PROCLIB(RESOLVER), and change the SETUP DD statement from:

```
//SETUP DD DISP=SHR,DSN=ADCD.Z22D.TCPPARMS(GBLRESOL),FREE=CLOSE
```

To

```
//SETUP DD DISP=SHR,DSN=USER.TCPPARMS(RESOLVER),FREE=CLOSE
```

*Modifying VTAMLST members:*

Create a VTAMLST member to define a TRL major node that matches the OSAs you are using. Member OSATRL2 in ADCD.\*.VTAMLST matches the sample devmap. If you create a new member to match your OSAs, add that member to USER.VTAMLST. To activate this major node when VTAM starts, add the VTAMLST member name to the VTAM configuration list (member ATCCONxx in VTAMLST) with which you start VTAM.

```
OSATRL1 VBUILD TYPE=TRL
OSATRL1E TRLE LNCTL=MPC,READ=(400),WRITE=(401),DATAPATH=(402), X
 PORTNAME=PORTA,MPCLEVEL=QDIO
OSATRL2E TRLE LNCTL=MPC,READ=(404),WRITE=(405),DATAPATH=(406), X
 PORTNAME=PORTB,MPCLEVEL=QDIO
```

## Optional tasks

Learn about optional configuration tasks for z Systems Development and Test Environment.

### Creating a user volume:

Some users might want to create one or more user volumes to isolate configuration data for some of their products. For example, you might want to isolate all of your

configuration files for IBM z Systems Development and Test Environment or you might want a new volume for a particular application or a newer version of an existing product. To create a new volume, do these following steps. This example creates a volume that is named USER00, placing it in the same directory as the other z/OS 2.2 ADCD disks:

- Create a volume in emulated 3390 format.

```
alckd /home/ibmsys1/z1090/disks/USER00 -d3390-9
```

**Note:** You can use -d3390-3 or other volume types and sizes if wanted. See alckd command in the section 4.1.2 of IBM zPDT Guide and Reference Redbook.

- Add the new volume to your device map

```
device 0CCC 3390 3390 /home/ibmsys1/z1090/disks/USER00
```

- After you start IBM z Systems Development and Test Environment, initialize the new volume

- Vary the volume offline

```
V 0CCC,OFFLINE
```

- Create and submit a member of a partitioned data set with the following JCL

**Note:** Customize the job card and VTOC parameters as needed. The VTOC parameters for a 3390-3 and a 3390-9 are shown.

```
//J01INIT JOB (ACCT),MSGCLASS=H,NOTIFY=&SYSUID.,REGION=0M
//* 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(CCC) NOVALIDATE NVFY VOLID(USER00) PURGE -
 VTOC(0,1,2939) INDEX(196,0,150)
/*
```

- If prompted on the console, reply U to the console command (R xx,U) to allow MVS to alter the contents of the volume. This job ends with return code 0.

- Vary the volume online:

```
V 0CCC,ONLINE
```

- Create a user catalog for the new volume by creating and submitting a member of a partitioned data set with the following JCL. Modify the job card as needed.

```
//J02CATLG 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))
/*
```

**Note:** If no user catalog is used, all cataloged data sets are 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 aliases for high-level qualifiers for data sets that you want to catalog in the USER00 user catalog. During migration, you need only to import the user catalog and replicate the alias definitions so that catalog entries are restored.

**Ensure that all new data sets are written to a new volume**

You can increase storage for all new permanent data sets by adding one or more newly created volumes to the volume attribute list (VATLSTxx member of the parmlib). The suffix *xx* of your volume attribute list parmlib member is the value found for the VAL= parameter in the IEASYSxx parmlib member corresponding to your loadparm. Copy the appropriate VATLSTxx member of the parmlib from ADCD.\*.PARMLIB to USER.PARMLIB and add the volume to any existing definitions as shown. In this example, the new USER00 volume is added with a permanently resident mount attribute, a use attribute of storage, and mount messages to be entered. The contents of this member must be in specific columns, so be sure to use the copied contents as a model. See the VATLSTxx section of the z/OS 2.2.0 MVS Initialization and Tuning Reference for the format and explanation of all fields and parameters in the volume attribute list.

```
VATDEF IPLUSE(PRIVATE),SYSUSE(PRIVATE)
B2SYS1,0,0,3390 ,Y
USER00,0,0,3390 ,Y
```

Changes to VATLST00 will not be effective until you perform the next IPL.

### Logging on to TSO from emulators not native to the host Linux:

After TCP/IP connectivity is established, you can use your favorite 3270 emulator software, such as IBM Personal Communications Manager (PCOMM), to connect to the non-SNA (coax) 3270 device emulator that is provided by z Systems Development and Test Environment. When you connect from outside the Linux system that is hosting z Systems Development and Test Environment, use the Linux TCPIP address and 3270 as port. (The actual port number is defined in the device map.)

### Creating TSO user IDs:

Learn how to create TSO user IDs for running z Systems Development and Test Environment.

A z/OS ADCD for IBM z Systems Development and Test Environment provides a number of predefined user IDs. “z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 143 contains the list of all predefined user IDs available to you.

If more TSO user IDs are required, they can be created through a series of commands. Typically, you want to use user IDs that have OMVS segments and an associated z/OS UNIX user directory when you are running Developer for z Systems. The example commands that are shown here create a user ID in the RDZUSERS group with an OMVS segment. Replace #userid, #name, and #password with appropriate values, and do not remove the quotation marks in the commands.

From a CLIST, REXX exec, or TSO command line, enter the following commands. The commands create the user ID, provide an OMVS segment, and assign an account number, default logon procedure, and region size. They also protect data sets with a high-level qualifier that belongs to the user from being accessed by other users. Finally, the commands create an alias in the master catalog to indicate that the users data sets are cataloged in the user catalog on volume S1CUST. Be sure that the substitutions in the HOME( ) and PROGRAM() parameters are in lowercase.

```
ADDGROUP RDZUSERS OMVS(AUTOGID)
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) COMMAND('ISPF'))
```

```

ADDSD '#userid.*' UACC(NONE)
PERMIT ACCT# CLASS(ACCTNUM) ID(#userid) ACCESS(READ)
DEFINE ALIAS (NAME('#userid') RELATE('USERCAT.Z22C.USER'))

```

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 lowercase.

```

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

```

### Defining a new logon procedure:

A common customization in z/OS systems is to alter the logon procedure that TSO users use. Do 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 ADCD.Z22D.PROCLIB(ISPFPROC), and make your 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(C2CFG1)'

```

To allow all users to use the TSOLOGON procedure, enter the following TSO commands.

```

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

```

### Altering system startup and shutdown scripts:

This z/OS 2.2 ADCD has shutdown scripts that contain a series of commands to shut down all active subsystems.

You might want to make these changes to the scripts.

The shutdown scripts stop OMVS automatically. Explicitly stopping zFS can expedite the shutdown process, but requires a response to a prompt on the operator console. If you want to stop zFS before you shut down OMVS, add this change:

```

F OMVS,STOPPFS=ZFS
F OMVS,SHUTDOWN

```

Some startup and shutdown scripts have PAUSE commands that you might find are too long or too short for your system. Many can be safely changed to pause for a shorter amount of time.

You can change startup scripts the same way. Other changes might include not starting particular subsystems, changing pause times, and so forth.

### Defining JES NJE connectivity:

Your z/OS distribution is a stand-alone 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 use the z Systems Development and Test Environment system.

Beginning with 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 run on the z Systems Development and Test Environment system, name the local system RUT0 and define a connection to M168.

```
$TNODE(Z21S),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
```

Give similar operator commands on the M168 system to complete the setup. Since this system is an existing system, the command to define the local node name was skipped. The following commands also assume that 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=RDz9100.RTP.IBM.COM
 $SN,SOCKET=REMOTE
```

The z Systems Development and Test Environment system can use the existing NJE definitions on the M168 system to connect to other NJE nodes in your network. Enter the following operator commands on the z Systems Development and Test Environment 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
```

For more information, see the “Network Job Entry” section of the *z/OS JES2 Initialization and Tuning Guide*.

## **z/OS 2.2 ADCD for z Systems Development and Test Environment**

Learn what's new with this distribution from past z/OS distributions, the products that are included with this distribution, the volumes that can be downloaded and their general contents, the migration guidelines, including the parmlib and proclib structure of the distribution, the loadparms available with this distribution and which subsystems they start, the user IDs and passwords available with the system, and the maintenance service levels of the products.

### **Customization help and location of product PGMDIRs**

Although the products in this ADCD were customized, you might find some products 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* is the product high-level qualifier and *l* is the library name. The purpose of these files is to provide PGMDIRs, readme file, and installation instructions. This ADCD is built from a driver system that uses different volume and library names. Thus, the names that are referenced in the SVSC files are different from the ADCD. You might need to make translations on names to implement in the ADCD. Included

on DVD1 on the RES1 volume are data sets with an HLQ of MVS.Z0SRxx that document MVS. The MVS PGMDIR is contained in this grouping.

## What's new in this release

Learn which modifications and enhancements are incorporated into this z/OS 2.2 ADCD for z Systems Development and Test Environment May 2017 edition.

- z/OS 2.2 and related products were upgraded to PUT1703/RSU1703.
- The following products were removed:
  - IBM Ported Tools for z/OS V1.3
  - DB2 V10
  - IMS Enterprise Suite 3.2.1
  - CICS 5.1
  - Websphere Application Server for zOS 8.0
- The following products are upgraded with the latest available release/mod level:
  - IBM Rational Team Concert™ V6.0.3
  - Urban Code Deploy 6.2.4
  - DB2 V12
  - DB2 Accessories Suite 4.01
  - IBM MQ 9.0.2
  - IBM Developer for z Systems 14.0
- The following products are added to the ADCD z/OS 2.2 May 2017 Edition:
  - IBM Batchpipes for OS/390® V2.1.0
  - IBM Installation Manager 1.4

## Products contained in this release

Lists the products contained in this release.

### z/OS 2.2 May 2017 Edition:

Lists the products available in the z/OS 2.2 May 2017 Edition.

| Product Number | Description                                   | Fmid    | Dependent FMID | HLQ  |
|----------------|-----------------------------------------------|---------|----------------|------|
| 5650-ZOS       | ICKDSF - Device Support Facilities, Base      | EDU1H01 | EDU1H01        | SYS1 |
| 5650-ZOS       | ICKDSF - Device Support Facilities, ISMF/MODS | EDU1H01 | FDU1H07        | SYS1 |
| 5650-ZOS       | ICKDSF - Device Support Facilities, ISMF/ENU  | EDU1H01 | FDU1H08        | SYS1 |
| 5650-ZOS       | Environmental Record Editing and Printing     | EER3500 | EER3500        | SYS1 |
| 5650-ZOS       | MICR/OCR                                      | EMI2220 | EMI2220        | SYS1 |
| 5650-ZOS       | TIOC                                          | ETI1106 | ETI1106        | SYS1 |
| 5650-ZOS       | OSA/SF                                        | H0GI400 | H0GI400        | SYS1 |

| Product Number | Description                                        | Fmid    | Dependent FMID | HLQ  |
|----------------|----------------------------------------------------|---------|----------------|------|
| 5650-ZOS       | Distributed File Service Base                      | H0H2410 | H0H2410        | SYS1 |
| 5650-ZOS       | C/C++ HOST PERFORMANCE ANALYZER                    | H24P111 | H24P111        | SYS1 |
| 5650-ZOS       | BCP Base                                           | HBB77A0 | HBB77A0        | SYS1 |
| 5650-ZOS       | SMP/E Planning and Migration Assistant Software In | HBCNC00 | HBCNC00        | SYS1 |
| 5650-ZOS       | SMP/E Planning and Migration Assistant             | HBCND0B | HBCND0B        | SYS1 |
| 5650-ZOS       | BULK DATA TRANSFER SNA NJE                         | HBD6602 | HBD6202        | SYS1 |
| 5650-ZOS       | BULK DATA TRANSFER BASE                            | HBD6602 | HBD6201        | SYS1 |
| 5650-ZOS       | BDT                                                | HBD6602 | HBD6602        | SYS1 |
| 5650-ZOS       | BOOKMANAGER READ/MVS - ENU                         | HBKM300 | HBKM310        | SYS1 |
| 5650-ZOS       | BOOKMANAGER READ/MVS - ENU                         | HBKM300 | HBKM300        | SYS1 |
| 5650-ZOS       | Library Server                                     | HBKQ400 | HBKQ400        | SYS1 |
| 5650-ZOS       | Hardware Configuration Manager                     | HCM1F10 | HCM1F10        | SYS1 |
| 5650-ZOS       | Cryptographic Services - System SSL Base           | HCPT420 | HCPT420        | SYS1 |
| 5650-ZOS       | z/OS Security Level 3 - System SSL Security Level  | HCPT420 | JCPT421        | SYS1 |
| 5650-ZOS       | z/OS Security Level 3 - OCSF Security Level 3      | HCRY740 | JCRY741        | SYS1 |
| 5650-ZOS       | z/OS Security Level 3 - IBM TDS for z/OS Security  | HRSL420 | JRSL421        | SYS1 |
| 5650-ZOS       | Enhanced Cryptographic Support - ICSF              | HCR77B0 | HCR77B0        | SYS1 |
| 5650-ZOS       | Cryptographic Services - OCSF Base                 | HCRY740 | HCRY740        | SYS1 |

| Product Number | Description                                        | Fmid    | Dependent FMID | HLQ  |
|----------------|----------------------------------------------------|---------|----------------|------|
| 5650-ZOS       | Hardware Configuration Definition Base             | HCS77A0 | HCS77A0        | SYS1 |
| 5650-ZOS       | Data Facility System Managed Storage Base & ENU    | HDZ2220 | HDZ2220        | SYS1 |
| 5650-ZOS       | Network File System Server and Client              | HDZ222N | HDZ222N        | SYS1 |
| 5650-ZOS       | z/OS Font Collection                               | HFNT110 | HFNT110        | SYS1 |
| 5650-ZOS       | z/OS Font Collection - Chinese, Japanese, Korean   | HFNT11J | HFNT11J        | SYS1 |
| 5650-ZOS       | FFST™                                              | HFST101 | HFST101        | SYS1 |
| 5650-ZOS       | 3270 PC File Transfer Program                      | HFX1112 | HFX1112        | SYS1 |
| 5650-ZOS       | GDDM NLS ENU                                       | HGD3200 | JGD3219        | SYS1 |
| 5650-ZOS       | GDDM NLS ENU                                       | HGD3200 | HGD3200        | SYS1 |
| 5650-ZOS       | GDDM-PGF                                           | HGD3201 | HGD3201        | SYS1 |
| 5650-ZOS       | ISPF Base                                          | HIF7P02 | HIF7P02        | SYS1 |
| 5650-ZOS       | IBM HTTP Server                                    | HHAP90P | HHAP90P        | SYS1 |
| 5650-ZOS       | IOCP                                               | HIO1104 | HIO1104        | SYS1 |
| 5650-ZOS       | Communications Server IP                           | HIP6220 | HIP6220        | SYS1 |
| 5650-ZOS       | Communications Server X11R4 XWindows               | HIP6220 | JIP622X        | SYS1 |
| 5650-ZOS       | Communications Server Security Level 3             | HIP6220 | JIP622K        | SYS1 |
| 5650-ZOS       | Integrated Security Services - Enterprise Identity | HIT7750 | HIT7750        | SYS1 |
| 5650-ZOS       | JES2 Base                                          | HJE77A0 | HJE77A0        | SYS1 |
| 5650-ZOS       | JES3                                               | HJS77A0 | HJS77A0        | SYS1 |
| 5650-ZOS       | IBM Knowledge Center for z/OS                      | HKCZ100 | HKCZ100        | SYS1 |
| 5650-ZOS       | Cryptographic Services - PKI Services              | HKY77A0 | HKY77A0        | SYS1 |



| Product Number | Description                                        | Fmid    | Dependent FMID | HLQ  |
|----------------|----------------------------------------------------|---------|----------------|------|
| 5650-ZOS       | XL C/C++ Base                                      | HLB77A0 | HLB77A0        | SYS1 |
| 5650-ZOS       | Language Environment® Base                         | HLE77A0 | HLE77A0        | SYS1 |
| 5650-ZOS       | Infoprint Server - IP PrintWay™ Base               | HMOS705 | HMOS705        | SYS1 |
| 5650-ZOS       | SMP/E Base                                         | HMP1J00 | HMP1J00        | SYS1 |
| 5650-ZOS       | High Level Assembler Toolkit                       | HMQ4160 | JMQ416A        | SYS1 |
| 5650-ZOS       | High Level Assembler Base                          | HMQ4160 | HMQ4160        | SYS1 |
| 5650-ZOS       | Infoprint Server - NetSpool Base                   | HNET7A0 | HNET7A0        | SYS1 |
| 5650-ZOS       | Infoprint Server - Print Interface Base            | HOP17A0 | HOP17A0        | SYS1 |
| 5650-ZOS       | OpenSSH for z/OS                                   | HOS2220 | HOS2220        | SYS1 |
| 5650-ZOS       | z/OS UNIX System Services Application Services Bas | HOT77A0 | HOT77A0        | SYS1 |
| 5650-ZOS       | Common Information Model                           | HPG77A0 | HPG77A0        | SYS1 |
| 5650-ZOS       | BCP - Program Management Binder                    | HPM77A0 | HPM77A0        | SYS1 |
| 5650-ZOS       | BCP - Capacity Provisioning                        | HPV77A0 | HPV77A0        | SYS1 |
| 5650-ZOS       | SDSF Base                                          | HQX77A0 | HQX77A0        | SYS1 |
| 5650-ZOS       | Security Server - RACF Base                        | HRF77A0 | HRF77A0        | SYS1 |
| 5650-ZOS       | RMF™ Base                                          | HRM77A0 | HRM77A0        | SYS1 |
| 5650-ZOS       | Integrated Security Services - Open Crypto Enhance | HRO7740 | HRO7740        | SYS1 |
| 5650-ZOS       | IBM Tivoli® Directory Server for z/OS Base         | HRSL420 | HRSL420        | SYS1 |
| 5650-ZOS       | Metal C Runtime Library                            | HSD7780 | HSD7780        | SYS1 |
| 5650-ZOS       | DFSORT Base                                        | HSM1M00 | HSM1M00        | SYS1 |
| 5650-ZOS       | z/OSMF Core Functions                              | HSMA220 | HSMA220        | SYS1 |

| Product Number | Description                                          | Fmid    | Dependent FMID | HLQ  |
|----------------|------------------------------------------------------|---------|----------------|------|
| 5650-ZOS       | z/OSMF ISPF                                          | HSMA221 | HSMA221        | SYS1 |
| 5650-ZOS       | IBM z/OS Management Facility - RM                    | HSMA222 | HSMA222        | SYS1 |
| 5650-ZOS       | z/OSMF WLM                                           | HSMA223 | HSMA223        | SYS1 |
| 5650-ZOS       | z/OSMF Software Deployment                           | HSMA224 | HSMA224        | SYS1 |
| 5650-ZOS       | z/OSMF Incident Log                                  | HSMA225 | HSMA225        | SYS1 |
| 5650-ZOS       | IBM z/OS Management Facility - Capacity Provisioning | HSMA226 | HSMA226        | SYS1 |
| 5650-ZOS       | z/OSMF Workflow                                      | HSMA227 | HSMA227        | SYS1 |
| 5650-ZOS       | z/OSMF Plug-in A                                     | HSMA228 | HSMA228        | SYS1 |
| 5650-ZOS       | z/OSMF Plug-in B                                     | HSMA229 | HSMA229        | SYS1 |
| 5650-ZOS       | z/OSMF Configuration Assistant                       | HSMA22A | HSMA22A        | SYS1 |
| 5650-ZOS       | z/OSMF Plug-in C                                     | HSMA22B | HSMA22B        | SYS1 |
| 5650-ZOS       | ESCON Director                                       | HSWF100 | HSWF100        | SYS1 |
| 5650-ZOS       | Integrated Security Services- Network Authentication | HSWK420 | HSWK420        | SYS1 |
| 5650-ZOS       | z/OS Security Level 3 - Network Authentication Srv   | HSWK420 | JSWK421        | SYS1 |
| 5650-ZOS       | TSO/E Base                                           | HTE77A0 | HTE77A0        | SYS1 |
| 5650-ZOS       | TSO/E ENU                                            | HTE77A0 | JTE77AE        | SYS1 |
| 5650-ZOS       | Runtime Library Extensions Base                      | HTV77A0 | HTV77A0        | SYS1 |
| 5650-ZOS       | BCP - Support for Unicode Base                       | HUN77A0 | HUN77A0        | SYS1 |
| 5650-ZOS       | Communications Server SNA                            | HVT6220 | HVT6220        | SYS1 |
| 5650-ZOS       | IBM Alternate Library for REXX on zSeries            | HWJ9143 | HWJ9143        | SYS1 |

| Product Number | Description                                      | Fmid    | Dependent FMID | HLQ    |
|----------------|--------------------------------------------------|---------|----------------|--------|
| 5650-ZOS       | z/OS File System Base                            | HZFS420 | HZFS420        | SYS1   |
| 5655-P97       | Encryption Facility DFSMSdss Encryption          | HCF773D | HCF773D        | SYS1   |
| 5655-P97       | Encryption Facility Encrypt Ser                  | HCF7740 | HCF7740        | SYS1   |
| 5655-103       | DITTO/ESA base+1:94                              | H0GB310 | H0GB310        | DIT130 |
| 5655-J51       | XML Toolkit for z/OS                             | HXML1A0 | HXML1A0        | IXM110 |
| 5655-W43       | IBM 31-Bit SDK for z/OS Java™ Technology Edition | HJVA710 | HJVA710        | JVA710 |
| 5655-DGG       | IBM 31-Bit SDK for z/OS Java Technology Edition  | HJVA800 | HJVA800        | JVA800 |
| 5655-W44       | IBM 64-Bit SDK for z/OS Java Technology Edition  | HJVB710 | HJVB710        | JVB710 |
| 5655-DGH       | IBM 64-Bit SDK for z/OS Java Technology Edition  | HJVB800 | HJVB800        | JVB800 |
| 5655-W97       | IBM WebSphere MQ for z/OS USS                    | HMS8000 | JMS8008        | CSQ800 |
| 5655-W97       | IBM WebSphere MQ for z/OS Base                   | HMS8000 | HMS8000        | CSQ800 |
| 5655-W97       | IBM WebSphere MQ for z/OS US English             | HMS8000 | JMS8001        | CSQ800 |
| 5655-W97       | IBM WebSphere MQ for z/OS Uppercase English ENP  | HMS8000 | JMS8004        | CSQ800 |
| 5655-MQ9       | IBM WebSphere MQ for z/OS Base                   | HMS9010 | HMS9010        | CSQ901 |
| 5655-MQ9       | IBM WebSphere MQ for z/OS US English             | HMS9010 | JMS9011        | CSQ901 |

| Product Number | Description                                     | Fmid    | Dependent FMID | HLQ    |
|----------------|-------------------------------------------------|---------|----------------|--------|
| 5655-MQ9       | IBM WebSphere MQ for z/OS Uppercase English ENP | HMS9010 | JMS9014        | CSQ901 |
| 5655-MQ9       | IBM WebSphere MQ for z/OS USS Web Component     | HMS9010 | JMS9016        | CSQ901 |
| 5655-MQ9       | IBM WebSphere MQ for z/OS USS Components        | HMS9010 | JMS9018        | CSQ901 |
|                |                                                 |         |                |        |

| Product Number | Description                           | Fmid    | Dependent FMID | HLQ    |
|----------------|---------------------------------------|---------|----------------|--------|
| 5655-M32       | Code Sample                           | H0HL140 | H0HL140        | APS450 |
| 5655-M32       | Code Page                             | H0HL141 | J0HL143        | APS450 |
| 5655-M32       | Code Page                             | H0HL141 | H0HL141        | APS450 |
| 5655-M32       | Code Page                             | H0HL15C | J0HL144        | APS450 |
| 5655-M32       | Code Page                             | H0HL15C | H0HL15C        | APS450 |
| 5655-M32       | Code Page                             | H0HL15D | H0HL15D        | APS450 |
| 5655-M32       | Code Page                             | H0HL15D | J0HL14S        | APS450 |
| 5655-M32       | UPLOAD FOR z/OS                       | HPRF417 | HPRF417        | APS450 |
| 5655-M32       | PSF for z/OS Base                     | HPRF450 | HPRF450        | APS450 |
| 5655-M32       | Download for z/OS                     | HPRF450 | JPRF452        | APS450 |
| 5655-M32       | AFP Download Plus                     | HPRF450 | JPRF451        | APS450 |
| 5655-M32       | ACIF                                  | HQN4450 | HQN4450        | APS450 |
| 5698-SA3       | IBM AUTOMATION CONTROL FOR Z/OS - TEP | HKAH35T | HKAH35T        | AUT350 |
| 5698-SA3       | SA z/OS Base Automation               | HWRE350 | HWRE350        | AUT350 |
| 5698-SA3       | SA z/OS CICS AUTO                     | HWRE350 | JWRE35C        | AUT350 |
| 5698-SA3       | SA z/OS UNIQUE                        | HWRE350 | JWRE35F        | AUT350 |
| 5698-SA3       | SA z/OS IMS AUTO                      | HWRE350 | JWRE35I        | AUT350 |
| 5698-SA3       | SA z/OS I/O Operations                | HWRE35D | HWRE35D        | AUT350 |

| Product Number | Description                                      | Fmid    | Dependent FMID | HLQ    |
|----------------|--------------------------------------------------|---------|----------------|--------|
| 5655-R29       | IBM Rational COBOL Runtime                       | HACZ601 | HACZ601        | ELA601 |
| 5655-Q50       | IBM Problem Determination Tools Common Component | HVWR170 | HVWR170        | EQAE00 |
| 5655-Q50       | IBM Z/OS DEBUGGER BASE                           | HADRE00 | HADRE00        | EQAE00 |
| 5655-Q12       | PDTCC                                            | HVWR170 | HVWR170        | FMND10 |
| 5655-Q12       | FILE MANAGER Z/OS BASE                           | HADLD10 | HADLD10        | FMND10 |
| 5655-Q12       | FILE MANAGER Z/OS CICS                           | HADLD10 | JADLD16        | FMND10 |
| 5655-Q12       | FILE MANAGER Z/OS DB2                            | HADLD10 | JADLD12        | FMND10 |
| 5655-Q12       | FILE MANAGER Z/OS IMS                            | HADLD10 | JADLD14        | FMND10 |
| 5655-Q11       | FAULT ANALYZER/ BASE                             | HADQD10 | HADQD10        | IDID10 |
| 5655-PL5       | IBM Enterprise PL/I for z/OS                     | H270510 | H270510        | IEL510 |
| 5655-PL5       | IBM Enterprise PL/I for z/OS FEATURE             | H270510 | J270511        | IEL510 |
| 5655-EC6       | IBM COBOL FOR Z/OS BASE                          | HADB610 | HADB610        | IGY610 |
| 5655-EC6       | IBM COBOL FOR Z/OS HFS                           | HADB610 | JADB61H        | IGY610 |
| 5655-EC6       | IBM COBOL FOR Z/OS US ENGLISH                    | HADB610 | JADB611        | IGY610 |
| 5635-ISP       | IBM DATA SET COMMANDER FOR Z/OS                  | HIQI810 | HIQI810        | IQI810 |
| 5697-NV6       | IBM Tivoli NetView® Full Version                 | HNV621B | HNV621B        | NET621 |
| 5697-NV6       | IBM Tivoli NetView English                       | HNV621B | JNV621E        | NET621 |
| 5697-NV6       | IBM AUTOMATION CONTROL FOR z/OS                  | HNVL12B | HNVL12B        | NET621 |

| Product Number | Description                                | Fmid    | Dependent FMID | HLQ    |
|----------------|--------------------------------------------|---------|----------------|--------|
| 5697-NV6       | IBM AUTOMATION CONTROL FOR z/OS English    | HNVL12E | JNVL12E        | NET621 |
| 5695-014       | IBM Alternate Library for REXX on zSeries  | HWJ9143 | HWJ9143        | FAN140 |
| 5695-013       | IBM Compiler for REXX on zSeries           | HWK0140 | HWK0140        | FAN140 |
| 5668-806       | VS FORTRAN LIBRARY MVS                     | HFL2602 | HFL2602        | AFF260 |
| 5668-806       | LIBRARY MVS DEP MODULES                    | HFL2602 | JFL2611        | AFF260 |
| 5668-806       | MVS IAD                                    | HFR2602 | HFR2602        | AFF260 |
| 5668-806       | MVS IAD (TSO)                              | HFR2602 | JFR2611        | AFF260 |
| 5668-806       | MVS IAD (ISPF)                             | HFR2602 | JFR2620        | AFF260 |
| 5668-806       | COMPILER MVS                               | HFT2602 | HFT2602        | AFF260 |
| 5668-806       | COMPILER MVS DEP MODULES                   | HFT2602 | JFT2611        | AFF260 |
| 5668-806       | PUBS POST SCRIPT                           | HFT2602 | JFT2612        | AFF260 |
| 5668-806       | PUBS BOOK MANAGER                          | HFT2602 | JFT2613        | AFF260 |
| 5655-X09       | IBM Sterling Connect:Direct® for z/OS V5.2 | HDGA520 | JDGA520        | DGA520 |
| 5655-CEE       | IBM z/OS Connect Enterprise Edition V2.0   | HZCN200 | HZCN200        | BAQ200 |
| 5655-D45       | IBM BATCHPIPES FOR OS/390                  | HACH301 | HACH301        | SAS210 |
|                |                                            |         |                |        |

### CICS V5.3:

Describes the CICS v5.3 features shipped with the z/OS v2.2 ADCD.

| Product Number | Description          | Fmid    | Dependent FMID | HLQ    |
|----------------|----------------------|---------|----------------|--------|
| 5655-Y04       | CICS - C feature     | HCI7000 | JCI7003        | DFH530 |
| 5655-Y04       | CICS - COBOL feature | HCI7000 | JCI7001        | DFH530 |

|          |                                                    |         |         |        |
|----------|----------------------------------------------------|---------|---------|--------|
| 5655-Y04 | CICS - PL/1 feature                                | HCI7000 | JCI7002 | DFH530 |
| 5655-Y04 | CICS REXX Development System                       | H0B7110 | H0B7110 | DFH530 |
| 5655-Y04 | CICS REXX Common for CICS/ESA                      | H0Z2110 | H0Z2110 | DFH530 |
| 5655-Y04 | CICS TS Licence                                    | HCTS530 | HCTS530 | DFH530 |
| 5655-Y04 | CICS - JAVA                                        | HCI7000 | JCI700D | DFH530 |
| 5655-Y04 | CICS - WAS Liberty Profile                         | HCI7000 | JCI700L | DFH530 |
| 5655-Y04 | CICS REXX Runtime Facility                         | H0B5110 | H0B5110 | DFH530 |
| 5655-Y04 | CICS Service Flow Feature for CICS/TS              | HCIZ300 | HCIZ300 | DFH530 |
| 5655-Y04 | CICS - System Manager                              | HCI7000 | JCI700M | DFH530 |
| 5655-Y04 | CICS - WS Security                                 | HCI7000 | JCI700W | DFH530 |
| 5655-Y04 | CICS - Base                                        | HCI7000 | HCI7000 | DFH530 |
| 5655-Y50 | CICS Transaction Server Feature Pack for Modern Ba | HCIF51B | HCIF51B | DFH530 |
| 5655-Y47 | CICS Transaction Server Feature Pack for Dynamic S | HCIF51D | HCIF51D | DFH530 |
| 5655-Y49 | CICS TS Feature Pack for Security Token Extensions | HCIF51A | HCIF51A | DFH530 |
| 5655-Y48 | CICS Transaction Server Feature Pack for Mobile Ex | HCIF51C | HCIF51C | DFH530 |

## CICS V5.2:

Describes the CICS v5.2 features shipped with the z/OS v2.2 ADCD.

| Product Number | Description          | Fmid    | Dependent FMID | HLQ    |
|----------------|----------------------|---------|----------------|--------|
| 5655-Y04       | CICS - C feature     | HCI6900 | JCI6903        | DFH520 |
| 5655-Y04       | CICS - COBOL feature | HCI6900 | JCI6901        | DFH520 |
| 5655-Y04       | CICS - PL/1 feature  | HCI6900 | JCI6902        | DFH520 |

|          |                                                    |         |         |        |
|----------|----------------------------------------------------|---------|---------|--------|
| 5655-Y04 | CICS REXX Development System                       | H0B7110 | H0B7110 | DFH520 |
| 5655-Y04 | CICS REXX Common for CICS/ESA                      | H0Z2110 | H0Z2110 | DFH520 |
| 5655-Y04 | CICS TS Licence                                    | HCTS520 | HCTS520 | DFH520 |
| 5655-Y04 | CICS - JAVA                                        | HCI6900 | JCI690D | DFH520 |
| 5655-Y04 | CICS - WAS Liberty Profile                         | HCI6900 | JCI690L | DFH520 |
| 5655-Y04 | CICS REXX Runtime Facility                         | H0B5110 | H0B5110 | DFH520 |
| 5655-Y04 | CICS Service Flow Feature for CICS/TS              | HCIZ300 | HCIZ300 | DFH520 |
| 5655-Y04 | CICS - System Manager                              | HCI6900 | JCI690M | DFH520 |
| 5655-Y04 | CICS - WS Security                                 | HCI6900 | JCI690W | DFH520 |
| 5655-Y04 | CICS - Base                                        | HCI6900 | HCI6900 | DFH520 |
| 5655-Y50 | CICS Transaction Server Feature Pack for Modern Ba | HCIF51B | HCIF51B | DFH520 |
| 5655-Y47 | CICS Transaction Server Feature Pack for Dynamic S | HCIF51D | HCIF51D | DFH520 |
| 5655-Y49 | CICS TS Feature Pack for Security Token Extensions | HCIF51A | HCIF51A | DFH520 |
| 5655-Y48 | CICS Transaction Server Feature Pack for Mobile Ex | HCIF51C | HCIF51C | DFH520 |

## DB2 V12:

Describes the DB2 v12 features shipped with the z/OS v2.2 ADCD.

| Product Number | Description                        | Fmid    | Dependent FMID | HLQ    |
|----------------|------------------------------------|---------|----------------|--------|
| 5697-Q05       | DB2 SPATIAL SUPPORT                | H2AG410 | H2AG410        | DSNC10 |
| 5697-Q05       | DB2 INT COMP UNICODE               | H2AF410 | H2AF410        | DSNC10 |
| 5697-Q05       | JSON capabilities for DB2 for z/OS | H2AS410 | H2AS410        | DSNC10 |



| Product Number | Description                                           | Fmid    | Dependent FMID | HLQ    |
|----------------|-------------------------------------------------------|---------|----------------|--------|
| 5697-Q05       | DB2 Adapter for z/OS Connect                          | H2AZ330 | H2AZ330        | DSNC10 |
| 5770-AF4       | IBM DB2 UTILITY SUITE FOR Z/OS                        | HDBCC1K | HDBCC1K        | DSNC10 |
| 5650-DB2       | DB2 BASE/TSO DB2 FOR Z/OS                             | HDBCC10 | HDBCC10        | DSNC10 |
| 5650-DB2       | IMS ATTACH                                            | HIYCC10 | HIYCC10        | DSNC10 |
| 5650-DB2       | DB2 SUBSYSTEM INITIALIZATION                          | HIZCC10 | HIZCC10        | DSNC10 |
| 5650-DB2       | IRLM V02.03.00                                        | HIR2230 | HIR2230        | DSNC10 |
| 5650-DB2       | RACF AUTHORIZATION EXIT                               | HDREC10 | HDREC10        | DSNC10 |
| 5650-DB2       | DB2 JDBC/SQLJ                                         | HDBCC10 | JDBCC12        | DSNC10 |
| 5650-DB2       | DB2 ENGLISH PANELS                                    | HDBCC10 | JDBCC14        | DSNC10 |
| 5650-DB2       | DB2 ODBC                                              | HDBCC10 | JDBCC17        | DSNC10 |
| 5650-DB2       | Z/OS APPLICATION CONNECTIVITY TO                      | HDDC211 | HDDC211        | DSNC10 |
| 5655-DT2       | IBM DB2 Administration tool for z/OS                  | H0IHC10 | H0IHB20        | ADBC10 |
| 5655-DC2       | IBM DB2 Object Comparison Tool for z/OS               | H25GC10 | H25GB20        | G0CC10 |
| 5697-QM2       | IBM DB2 Query Management Facility™ for zOS NL version | HSQCC10 | HSQCC10        | QMFC10 |
| 5697-QM2       | HHPCC10 (QMF™ HPO)                                    | HHPCC10 | HSQCC10        | QMFC10 |
| 5697-QM2       | HQDCC10 (QMF Data Service)                            | HQDCC10 | HSQCC10        | QMFC10 |
| 5697-QM2       | JYQCC10 (QMF applications)                            | JYQCC10 | HSQCC10        | QMFC10 |
| 5697-QM2       | JSQCC1Q (QMF for z/OS OTC)                            | JSQCC1Q | HSQCC10        | QMFC10 |
| 5697-QM2       | JSQCC51 (QMF Uppercase English)                       | JSQCC51 | HSQCC10        | QMFC10 |

**DB2 V11:**

Describes the DB2 v11 features shipped with the z/OS v2.2 ADCD.

| Product Number | Description                                   | Fmid    | Dependent FMID | HLQ    |
|----------------|-----------------------------------------------|---------|----------------|--------|
| 5697-Q04       | DB2 SPATIAL SUPPORT                           | H2AF310 | J2AG310        | DSNB10 |
| 5697-Q04       | DB2 INT COMP UNICODE                          | H2AF310 | H2AF310        | DSNB10 |
| 5697-Q04       | JSON capabilities for DB2 for z/OS            | H2AF310 | J2AS220        | DSNB10 |
| 5697-Q04       | DB2 Adapter for z/OS Connect                  | H2AZ330 | H2AZ330        | DSNB10 |
| 5655-W87       | IBM DB2 UTILITY SUITE FOR Z/OS                | HDBBB10 | JDBBB1K        | DSNB10 |
| 5615-DB2       | DB2 BASE Z/OS                                 | HDBBB10 | HDBBB10        | DSNB10 |
| 5615-DB2       | DB2 JDBC/SQLJ Z/OS                            | HDBBB10 | JDBBB12        | DSNB10 |
| 5615-DB2       | DB2 ENGLISH Z/OS                              | HDBBB10 | JDBBB14        | DSNB10 |
| 5615-DB2       | DB2 ODBC Z/OS                                 | HDBBB10 | JDBBB17        | DSNB10 |
| 5615-DB2       | DB2 RACF EXIT Z/OS                            | HDREB10 | HDREB10        | DSNB10 |
| 5615-DB2       | IRLM                                          | HIR2230 | HIR2230        | DSNB10 |
| 5615-DB2       | DB2 IMS ATTACH Z/OS                           | HIYBB10 | HIYBB10        | DSNB10 |
| 5615-DB2       | DB2 SUBSYS INIT Z/OS                          | HIZBB10 | HIZBB10        | DSNB10 |
| 5655-DAT       | IBM DB2 Administration tool for z/OS          | H0IHB10 | H0IHB10        | ADBB10 |
| 5655-DOC       | IBM DB2 Object Comparison Tool for z/OS       | H25GB10 | H25GB10        | ADBB10 |
| 5615-DB2       | QMF CLASSIC EDITION - ENGLISH                 | HSQBB10 | HSQBB10        | QMFB10 |
| 5615-DB2       | Z/OS APPLICATION CONNECTIVITY TO DB2 FOR Z/OS | HDDA211 | HDDA211        | DDA211 |

### IMS 14.1:

Describes the IMS v14.1 features shipped with the z/OS v2.2 ADCD.

| Product Number | Description                 | Fmid    | Dependent FMID | HLQ    |
|----------------|-----------------------------|---------|----------------|--------|
| 5635-A05       | IMS                         | HMK1400 | HMK1400        | DFSE10 |
| 5635-A05       | IMS JAVA ON DEMAND FEATURES | HMK1400 | JMK1406        | DFSE10 |
| 5635-A05       | IMS DATABASE MANAGER        | HMK1400 | JMK1401        | DFSE10 |
| 5635-A05       | IMS DLT                     | HMK1400 | JMK1405        | DFSE10 |
| 5635-A05       | IMS EXT. TERMINAL OPT       | HMK1400 | JMK1403        | DFSE10 |
| 5635-A05       | IMS RLT                     | HMK1400 | JMK1404        | DFSE10 |
| 5635-A05       | IRLM V2.3                   | HIR2230 | HIR2230        | DFSE10 |
| 5635-A05       | IMS TRANSACTION MANAGER     | HMK1400 | JMK1402        | DFSE10 |

### IMS 13.1:

Describes the IMS v13.1 features shipped with the z/OS v2.2 ADCD.

| Product Number | Description                 | Fmid    | Dependent FMID | HLQ    |
|----------------|-----------------------------|---------|----------------|--------|
| 5635-A04       | IMS                         | HMK1300 | HMK1300        | DFSD10 |
| 5635-A04       | IMS JAVA ON DEMAND FEATURES | HMK1300 | JMK1306        | DFSD10 |
| 5635-A04       | IMS DATABASE MANAGER        | HMK1300 | JMK1301        | DFSD10 |
| 5635-A04       | IMS DLT                     | HMK1300 | JMK1305        | DFSD10 |
| 5635-A04       | IMS EXT. TERMINAL OPT       | HMK1300 | JMK1303        | DFSD10 |
| 5635-A04       | IMS RLT                     | HMK1300 | JMK1304        | DFSD10 |
| 5635-A04       | IRLM                        | HIR2230 | HIR2230        | DFSD10 |
| 5635-A04       | IMS TRANSACTION MANAGER     | HMK1300 | JMK1302        | DFSD10 |

### WebSphere Application Server V9.0:

Lists information about the WebSphere Application Server v9 available with this z/OS 2.2 ADCD.

| Product Number | Description                           | Fmid    | Dependent FMID | HLQ    |
|----------------|---------------------------------------|---------|----------------|--------|
| 5655-WAS       | WebSphere Application Server for z/OS | HBBO900 | HBBO900        | WAS900 |

### WebSphere Application Server V8.5:

Lists information about the WebSphere Application Server v8.5 available with this z/OS 2.2 ADCD.

| Product Number | Description                           | Fmid    | Dependent FMID | HLQ    |
|----------------|---------------------------------------|---------|----------------|--------|
| 5655-W65       | WebSphere Application Server for z/OS | HBBO850 | HBBO850        | WAS850 |

### Tivoli Workload Scheduler 9.3.0:

Lists information about the Tivoli Workload Scheduler 9.3.0 available with this z/OS 2.2 ADCD.

| Product Number | Description                                           | Fmid    | Dependent FMID | HLQ    |
|----------------|-------------------------------------------------------|---------|----------------|--------|
| 5698-T08       | IBM WORKLOAD SCHEDULER (Agent)                        | HWSZ930 | HWSZ930        | TWS930 |
| 5698-T08       | IBM WORKLOAD SCHEDULER (Engine)                       | JWSZ932 | JWSZ932        | TWS930 |
| 5698-T08       | IBM WORKLOAD SCHEDULER (Engine English)               | JWSZ93B | JWSZ93B        | TWS930 |
| 5698-T08       | IBM WORKLOAD SCHEDULER (End-to-End and Java Enablers) | JWSZ933 | JWSZ933        | TWS930 |
| 5698-T08       | IBM WORKLOAD SCHEDULER (z/OS Connector Enabler)       | JWSZ934 | JWSZ934        | TWS930 |

## Tivoli OMEGAMON V5.4.0:

Lists the Tivoli OMEGAMON® v 5.4.0 features available in the z/OS 2.2 ADCD.

| Product Number | Description                                               | Fmid    | Dependent FMID | HLQ    |
|----------------|-----------------------------------------------------------|---------|----------------|--------|
| 5698-AA8       | IBM Tivoli OMEGAMON Performance Management Suite for z/OS |         |                | KAN541 |
| 5698-AA8       | ITCAM AD Com Svc                                          | HAAD71C | HAAD71C        | KAN541 |
| 5698-AA8       | ITCAM AD for z/OS                                         | HAAD710 | HAAD710        | KAN541 |
| 5698-AA8       | LOG FORWARDER                                             | HGLA220 | HGLA220        | KAN541 |
| 5698-AA8       | z/OS DLA                                                  | HIZD310 | HIZD310        | KAN541 |
| 5698-AA8       | GENERIC CONFIGURATI                                       | HKCF730 | HKCF730        | KAN541 |
| 5698-AA8       | OMEGAMON XE for CICS                                      | HKC5530 | HKC5530        | KAN541 |
| 5698-AA8       | OMEGAMON XE for DB2 for z/OS PE Lic Key                   | HKDB54X | HKDB54X        | KAN541 |
| 5698-AA8       | OMEGAMON XE for DB2 for z/OS PE Base                      | HKDB540 | HKDB540        | KAN541 |
| 5698-AA8       | End to End                                                | HKET620 | HKET620        | KAN541 |
| 5698-AA8       | OMEGAMON XE CICS TG                                       | HKGW530 | HKGW530        | KAN541 |
| 5698-AA8       | OMEGAMON XE for IMS                                       | HKI5530 | HKI5530        | KAN541 |
| 5698-AA8       | OMEGAMON JVM MONITR                                       | HKJJ540 | HKJJ540        | KAN541 |
| 5698-AA8       | CONFIG MGR MQ ON MVS                                      | HKMC730 | HKMC730        | KAN541 |
| 5698-AA8       | OMEG XE WEBSPHMQ 390                                      | HKMQ730 | HKMQ730        | KAN541 |
| 5698-AA8       | OMEGAVIEW                                                 | HKMV310 | HKMV310        | KAN541 |
| 5698-AA8       | OMEGAMON XE ON z/OS                                       | HKM5530 | HKM5530        | KAN541 |
| 5698-AA8       | OMEGAMON XE for MFN                                       | HKN3530 | HKN3530        | KAN541 |
| 5698-AA8       | OMNIMON BASE                                              | HKOB730 | HKOB730        | KAN541 |
| 5698-AA8       | OM XE MSG BROKER ZOS                                      | HKQI730 | HKQI730        | KAN541 |
| 5698-AA8       | Shared Probes                                             | HKSB740 | HKSB740        | KAN541 |

| Product Number | Description                                 | Fmid    | Dependent FMID | HLQ    |
|----------------|---------------------------------------------|---------|----------------|--------|
| 5698-AA8       | OMEGAMON XE Storage                         | HKS3540 | HKS3540        | KAN541 |
| 5698-AA8       | OMEGAVIEW II® ENTERPR                       | HKWO310 | HKWO310        | KAN541 |
| 5698-AA8       | ITCAM AD TEMA                               | HKYN710 | HKYN710        | KAN541 |
| 5698-AA8       | PERF Mgmt TADz ID                           | HPMS541 | HPMS541        | KAN541 |
| 5698-AA8       | Optim™ Extended Insight Client for z/OS     | HPMZ540 | HPMZ540        | KAN541 |
| 5698-AA8       | OM DE INSTALL                               | HKWO310 | JKWO530        | KAN541 |
| 5698-A79       | Installation/ Configuration Assistance Tool | HKCI310 | HKCI310        | KAN541 |
| 5698-A79       | Tivoli Enterprise Monitoring Server         | HKDS630 | HKDS630        | KAN541 |
| 5698-A79       | TMS:Engine                                  | HKLV630 | HKLV630        | KAN541 |

#### Rational Products and Miscellaneous products:

Describes the Rational and miscellaneous product features shipped with the z/OS v2.2 ADCD.

| Product Number | Description                                           | Fmid    | Dependent FMID | HLQ    |
|----------------|-------------------------------------------------------|---------|----------------|--------|
| 5655-W65       | IBM Installation Manager for z/OS                     | HGIN140 | HGIN140        | GIN143 |
| 5724-AC5       | IBM Developer for z Systems                           | HHOPE00 | HHOPE00        | FELE00 |
| 5724-AC5       | IBM Developer for z Systems Host Utilities            | HAKGE00 | HAKGE00        | FELE00 |
| 5655-EX1       | IBM Explorer for z/OS V3.0.1                          | HALG300 | HALG300        | BLZ603 |
| 5724-V04       | RATIONAL BUILD AGENT for z/OS                         | HRBA603 | HRBA603        | BLZ603 |
| 5724-V04       | RATIONAL TEAM CONCERT - BUILD SYSTEM TOOLKIT for z/OS | HRBT603 | HRBT603        | BLZ603 |
| 5725-M54       | UrbanCode Deploy for z/OS V6.2.4                      | HRUC624 | HRUC624        | BUZ624 |

## **Reported problems, fixes, and maintenance**

None.

## **Build structure**

Describes the build structure of the volumes shipped with the z/OS v2.2 ADCD.

### **D2RES1 and D2RES2**

D2RES1 is a 3390-9 volume that contains the base MVS system software target, or run-time libraries, and other system data sets required to IPL the system and use TSO/E and ISPF. This is the MVS IPL volume.

D2RES2 is a 3390-9 volumes that are an extension of the SYSRES volume D2RES1. The volume is also required for IPL.

### **D2SYS1**

This 3390-9 volume contains data sets that you might change, either through normal use of the system or user customization. System control data sets, such as SYS1.IPLPARM,IODF (I/O Definition File), and system's master catalog (CATALOG.Z22D.MASTER) reside on this volume. This volume is required to IPL. D2SYS1 contains the usercatalog USERCAT.Z22D.PRODS containing entries for z/OS related products

### **D2CFG1**

This 3390-9 volume contains data sets for storing user configuration. The 'USER.\*\*' and FEU.\*\* data sets reside on this volume, they are cataloged in the master catalog. The RACF database reside on this volume.

### **D2USS1 and D2USS2**

D2USS1 is a 3390-9 volume that contains the Fonts, Version and zOSMF zFS files for UNIX System Services of z/OS. This volume is required to IPL.

D2USS2 is a 3390-9 volume that contains root zFS and zFS files for UNIX System Services of z/OS. This volume is required to IPL.

### **D2PRD1, D2PRD2 and D2PRD3**

D2PRD1 to D2PRD3 are 3390-9 volumes that contains all z/OS products.

### **D2DIS1 and D2DIS2**

These volumes contains the base MVS system software distribution libraries. They contain the DLIB (Distribution Library) and zone CSI's. These volumes are needed to install service or products on the MVS system using SMP/E. These volumes are all 3390-9.

### **D2PAGA, D2PAGB and D2PAGC**

These volumes contain page data sets. The D2PAGA volume has the PLPA, Common and one Local page data set. Rest of the 2 volumes each have one Local page data set. The loadparms that bring up DB2 or WAS have been defined to use all the above 3 page volumes, all other loadparms use only the first two volumes. If you do not want to use all of these local page data sets, you can modify the PAGE parameter on IEASYSnn member, accordingly.

### **D2DBB1 and D2DBB2**

D2DBB1 is a 3390-9 volume that contains DB2 V11 related data. D2DBB1 contains the DB2 usercatlog, USERCAT.Z22D.DB2VB. Also D2DBB1 contains all DB2 target libs AND DB2 V11 utility products. This volume is not required to IPL but is required to bring up DB2 V11.

D2DBB2 is a 3390-9 volume that contains DB2 catalog and directories with HLQ DSNCB10, which are SMS managed. The volume is not required for IPL but is needed to bring up DB2.

#### **D2DBC1, D2DBC2 and D2DBAR**

D2DBC1 is a 3390-9 volume that contains DB2 V12 related data. D2DBC1 contains the DB2 usercatlog, USERCAT.Z22D.DB2VC. Also D2DBC1 contains all DB2 target libs and DB2 V12 utility products. This volume is not required to IPL but is required to bring up DB2 V12.

D2DBC2 is a 3390-9 volume that contains DB2 catalog and directories with HLQ DSNCC10, which are SMS managed. The volume is not required for IPL but is needed to bring up DB2.

D2DBAR is a 3390-9 volume defined in a new esoteric called DBARCH. This esoteric has been defined in DB2, for allocating the archive logs into this storage volume. This volume is required, if you want to bring up DB2.

#### **D2C531**

D2C531 is a 3390-9 volume that contains CICS 5.3 Target, DLIB and all related data for CICS 5.3. USERCAT.Z22D.CICS530 resides on this volume. This volume is not required to IPL but is required to bring up CICS 5.3 and apply CICS maintenance.

#### **D2C521**

D2C521 is a 3390-9 volume that contains CICS 5.2 Target, DLIB and all related data for CICS 5.2. USERCAT.Z22D.CICS520 resides on this volume. This volume is not required to IPL but is required to bring up CICS 5.2 and apply CICS maintenance.

#### **D2W901 and D2W902**

D2W901 and D2W902 are 3390-9 volumes that contains target and distribution libraries for Websphere Application Services.

#### **D2INM1**

D2INM1 is the 3390-9 volumes that contains IBM Installation Manager libraries.

#### **D2W851, D2W852, D2W853, D2W854 and D2W855**

D2W851, D2W852, D2W853, D2W854 and D2W855 are 3390-9 volume that contains target and distribution libraries for Websphere Application Services. D2W804 and D2W805 are 3390-9 volumes that contain pre-configured zFS files required for manually starting WAS.

#### **D2IME1 and D2IMD1**

D2IME1 is a 3390-9 volume that contains IMS Target, DLIB and all related data for IMS 14. USERCAT.Z22D.IMS14 resides on this volume. This volume is not required to IPL but is required to bring up IMS and perform maintenance on IMS.



D2IMD1 is a 3390-9 volume that contains IMS Target, DLIB and all related data for IMS 13. USERCAT.Z22D.IMS13 resides on this volume. This volume is not required to IPL but is required to bring up IMS and perform maintenance on IMS.

#### **D2KAN1**

D2KAN1 is a 3390-9 volume that contains Target and DLIB libraries for Tivoli OMEGAMON v5.4.0.

#### **D2BLZ1**

D2BLZ1 is a 3390-9 volume that contains Target and DLIB libraries for Rational Team Concert 6.0.3. This volume is not required to IPL but is required to bring up RTC.

#### **SARES1**

This 3390-9 volume contains a single volume stand alone system. This volume can be used to 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 leave accessible. The volume should not be altered. The volume should stay in a background mode and be available for emergency type of 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 Unix system service support. The system cannot be used to install products or apply maintenance via SMP/E.

To IPL the standalone res volume:

Load Address: 0Axx

Load Parms: 0AxxSAM

### **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 D2xxxx volumes except for D2SYS1. D2SYS1 contains RACF, IPL, and catalog datasets that are user dependent.

The ADCD does not contain a generalized migration utility or process. Each user has unique requirements. It is recommended that you build a migration plan based on your unique needs. The ADCD does have some assistance. The ADCD uses a system of concatenated libraries (see below) The highest level of concatenation is USER.xxxxxxxx. The ADCD distributes these libraries empty. It is recommended that the user place changes or overrides in these libraries. It is further recommended that the USER.xxxxxxxx libraries be backed up regularly. When a new ADCD release is installed the USER.xxxxxxxx libraries can be copied from backups to the new USER.xxxxxxxx libraries to provide assistance in migration of programs and parameters.

The ADCD is distributed with a standard library concatenation for the following:

- LINKLST
- PROCLIB (including TSO procedures)

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

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

- USER.LINKLIB
- ADCD.Z22D.LINKLIB
- SYS1.LINKLIB

USER.xxxxx libraries have all been built on C2CFG1 which is the only volume that should contain user data.

USER.xxxxx will not be changed by ADCD or System processes; thus, user updates in USER.xxxxx will be retained between release levels. ADCD.Z22D.xxxxx libraries are allocated on D2SYS1 and are the libraries that are used by ADCD developers. No RACF rules exist on these libraries; however, changes to these libraries could destroy customization necessary to bring up many products. SYS1.xxxxx libraries should NEVER be updated by users or ADCD development. System libraries should ONLY be updated through SMP/E install, maintenance processes, or like processes.

**Note:** Changes to System libraries or ADCD 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:

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

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

## LOADPARMS Options

Lists available preconfigured distributed LOADPARMS.

**Note:** Start JES2 cold the first time you start the system.

*Table 4. Listing of available preconfigured distributed LOADPARMS.*

| LOADPARAM | Description                                                                                                                   |
|-----------|-------------------------------------------------------------------------------------------------------------------------------|
| CS        | CLPA and Cold start of JES2. Base z/OS system functions, that is, no CICS, DB2, IMS, Websphere Application Server, and so on. |
| WS        | CLPA and Warm start of JES2. Base z/OS system functions, that is, no CICS, DB2, IMS, Websphere Application Server, and so on. |
| 00        | CLPA and Warm start of JES2. Base z/OS system functions, that is, no CICS, DB2, IMS, Websphere Application Server, and so on. |
| CI        | CLPA and Warm start of JES2. Loads CICS 5.3 and 5.2 libraries. Starts CICS 5.3, z/OSMF, and IBM Developer for z Systems.      |

Table 4. Listing of available preconfigured distributed LOADPARMS. (continued)

| LOADPARM | Description                                                                                                                                                                                            |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DB       | CLPA and Warm start of JES2. Loads DB2 V11 and V10 libraries. Starts DB2 V11, z/OSMF, and IBM Developer for z Systems.                                                                                 |
| IM       | CLPA and Warm start of JES2. Loads IMS 14 and 13 libraries. Starts IMS 14, z/OSMF, and IBM Developer for z Systems.                                                                                    |
| IZ       | CLPA and Warm start of JES2. Starts z/OSMF and IBM Developer for z Systems.                                                                                                                            |
| WA       | CLPA and Warm start of JES2. Loads Websphere Application Server v9 and v8.5 libraries. Starts IBM Developer for z Systems. Websphere Application Server needs to be manually started.                  |
| AL       | CLPA and Warm start of JES2. Loads all middleware libraries. Starts CICS 5.3, DB2 V11, IMS 14, z/OSMF, IBM Developer for z Systems and UCD. Websphere Application Server needs to be manually started. |
| DC       | CLPA and Cold start of JES2. Loads CICS, DB2 libraries. Starts CICS 5.3, DB2 V11, UCD 6.2.0, RTC 6.0.2, IBM Developer for z Systems and UCD.                                                           |
| DW       | CLPA and Warm start of JES2. Loads CICS, DB2 libraries. Starts CICS 5.3, DB2 V11, UCD 6.2.0, RTC 6.0.2, IBM Developer for z Systems and UCD.                                                           |
| AU       | CLPA and Warm Start of JES2. Loads based on the option what you selected during the provision using tools                                                                                              |

## Console PF Key Settings

Describes the default PF key assignments.

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

## USERIDS

Lists the TSO User IDs and passwords that are set up on your system.

*Table 5. Predefined USERIDS..* The following TSO User IDs and passwords are set up on your system:

| User ID                                 | Password         |
|-----------------------------------------|------------------|
| ADCDMST (RACF special authority)        | SYS1 or ADCDMST  |
| IBMUUSER (RACF special authority)       | SYS1 or IBMUUSER |
| SYSADM (DB2 and RACF special authority) | SYS1 or SYSADM   |
| SYSOPR (DB2 and RACF special authority) | SYS1 or SYSOPR   |
| ADCD - ADCDZ                            | TEST             |
| WEBADM                                  | WEBADM           |
| OPEN1 through OPEN3                     | SYS1             |
| ZOSMFAD                                 | ZOSMFAD          |

## Maintenance Service Levels

All the products on the ADCD 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 ADCD was built would not be added to the ADCD. The following PUT levels might be minus PTFs that were open at build time. Also, some functions might 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 2.2 May 2017 are at RSU level 1703.
- All functions of the CICS TS 5.3 are at RSU level 1703.
- All functions of the CICS TS 5.2 are at RSU level 1703.
- The functions of the DB2 12 base are at RSU level 1703.
- The functions of the DB2 11 base are at RSU level 1703.
- All functions of the IMS 14.1.0 are at RSU level 1703.
- All functions of the IMS 13.1.0 are at RSU level 1703.
- Tivoli OMEGAMON v5.4.0 is at RSU level 1703.

## Hints and tips to manage the z/OS 2.2 ADCD

Learn with the examples in this sections about page data sets available, starting IMS and running an IVP transaction, WLM configuration provided, ACS routines provided, recreating the coupling data sets, allocating DB2 archive logs into an esoteric, steps for dumping SMF data into GDG (Generation Data Group) data sets, using the esoteric devices, and how to start IBM Healthchecker for z/OS, IBM z/OS Management Facility (z/OSMF), and WebSphere Application Server.

### Page data sets

This ADCD system provides you with three volumes that are dedicated for page data sets, D2PAGA through D2PAGC.

The following list shows the content of the volumes.

- D2PAGA SYS1.S0W1.PLPA.PAGE , SYS.S0W1.COMMON.PAGE , SYS1.S0W1.LOCALA.PAGE
- D2PAGB SYS1.S0W1.LOCALB.PAGE
- D2PAGC SYS1.S0W1.LOCALC.PAGE

The preconfigured loadparms to start DB2 and WebSphere Application Server use all of the above listed page data sets. The remaining loadparms use only the data sets on volumes D2PAGA and D2PAGB. Depending upon your requirement, you can add or remove these page data sets by using the following parameter in the IEASYSxx member after you copy it to USER.PARMLIB.

```
PAGE=(SYS1.S0W1.PLPA.PAGE,
 SYS1.S0W1.COMMON.PAGE,
 SYS1.S0W1.LOCALA.PAGE,
 SYS1.S0W1.LOCALB.PAGE,L),
```

## Starting IMS and running an IVP transaction

The instructions for starting and stopping IMS can be found at [This page has pointers to the levels of IMS V13 and V14. On each level, click \*\*Issues, Hints and Tips\*\* to see the instructions for starting and stopping. The IVP jobs are available on the ADCD system, in the data set DFSE10.INSTALIB for IMS V14, and DFSD10.INSTALIB for IMS V13. You must run each series in a specific manner to create the particular IVP environment and data to be able to follow each IVP.](http://dtsc.dfw.ibm.com/MVSDS/'HTTPD2.DFSE10.PUBLIC.SHTML(INDEX)'>http://dtsc.dfw.ibm.com/MVSDS/'HTTPD2.DFSE10.PUBLIC.SHTML(INDEX)'</a>.</p>
</div>
<div data-bbox=)

## WLM configuration provided on ADCD z/OS

The WLM configuration of application environments for DB2 Stored Procedures for DBBG are self-documented in the WLM application environment display.

For DBBG (DB2 V11), the comment describes the usage of each environment.

|                          |                    |          |
|--------------------------|--------------------|----------|
| DBBGENV                  | Default            |          |
| DBBGENVC                 | DSNWLM_DSNACICS    |          |
| DBBGENVD                 | DSNWLM_DEBUGGER    |          |
| DBBGENVG                 | DSNWLM_GENERAL     |          |
| DBBGENVJ                 | DSNWLM_JAVA        |          |
| DBBGENVM                 | DSNWLM_MQSERIES    |          |
| DBBGENVO                 | DSNWLM_JAVA_BIGMEM |          |
| DBBGENVP                 | DSNWLM_PGM_CONTROL |          |
| DBBGENVR                 | DSNWLM_REXX        |          |
| DBBGENVU                 | DSNWLM_UTILS       |          |
| DBBGENVW                 | DSNWLM_WEBSERVICES |          |
| DBBGENVX                 | DSNWLM_XML         |          |
| DBBGENV1                 | DSNWLM_NUMTCB1     | DBBGRFSH |
| Stored Procedure REFRESH |                    |          |

## ACS routines provided on ADCD

DB2 V11 requires that its Catalog and Directory are SMS managed. The ACS routines that are available on ADCD z/OS system were created to handle HLQ DSNCB10, which is the HLQ of the DB2 V11 Catalog and Directory. The logic is to use two qualifiers; the first is DSNCB10, and the second is DSNDBC or DSNDBD. The DSNDBC and DSNDBD indicate the cluster and data portion of the VSAM data set of a DB2 Tablespace. Other data sets for DB2 V11 have a HLQ of DSNB10, and these data sets are not SMS managed. A DB2 STOGROUP called SYSSMS was created for the Catalog and Directory Tablespace with a VCAT of DSNCB10 and VOL of (\*). The (\*) is for SMS managed STOGROUP. The DB2 STOGROUP, SYSDEFLT uses VCAT of DSNB10 and lists actual volumes, and it is not SMS managed. The installation is configured to use SYSSMS when it creates the Catalog and Directory Tablespace. If you create and specify a STOGROUP of your own or default to SYSDEFLT STOGROUP, your Tablespace will fall into non-SMS managed volumes as specified in the STOGROUP.

## Recreating the coupling data sets

The following coupling data sets are provided on ADCD.

```
SYS1.ADCDPL.CDS01
SYS1.ADCDPL.CDS02
SYS1.ADCDPL.CFRM.CDS01
SYS1.ADCDPL.CFRM.CDS02
SYS1.ADCDPL.LOGR.CDS01
SYS1.ADCDPL.LOGR.CDS02
SYS1.ADCDPL.OMVS.CDS01
SYS1.ADCDPL.OMVS.CDS02
SYS1.ADCDPL.WLM.CDS01
SYS1.ADCDPL.WLM.CDS02
```

In case any of these data sets are corrupted or become unusable, delete them and use the JCL that is available in member ALOCCOPL in 'ADCD.LIB.JCL' data set to allocate them again.

**Note:** This procedure is provided to you, assuming that you have a thorough knowledge of setting up a SYSPLEX environment. These steps are a bit involved and could cause disruption of normal functioning of your ADCD system.

## Allocating DB2 archive logs into an esoteric device

Additional storage volume D2DBAR was defined and is pointed to by a new esoteric DBARCH for DB2 archive logs. You can find instructions for adding more volumes to this esoteric in this topic. Ensure to map this volume D2DBAR to device# 0AA3 in your devmap so that it is in sync with the IODF definition.

In this edition of ADCD z/OS, the loadparms to start DB2 were configured to use the esoteric device DBARCH for holding the DB2 archive logs. This means that the archive logs go only into this volume (or volumes if you add more to the esoteric). You still need to manage the space on the volume, and delete archive logs that are no longer needed. The fourth qualifier in the data set name is a date of the format *Dyynnn*, where *yy* is the year, and *nnn* is the Julian day of the year. When you start DB2, examine the console log to ensure that it does not read any archive logs. If no archive logs are used at the start, you have a clean start of DB2, and then you can delete all the prior archive logs. If you are running DB2 and running out of space in the esoteric DBARCH volumes, check the SDSF console log for DB2 and determine the start date. Then, you can delete archive logs before the start date of the last clean start of DB2. You must monitor the space available on your DBARCH esoteric volumes. To achieve better space management, turn off logging on some of your table spaces that can be recovered by just dropping, re-creating, and reloading. You can also reduce logging by using the LOG NO parm in your load utility job. You must reset the COPY pending flag with the NOCOPYPEND parm in the load utility or the REPAIR utility. And note that these options are used only for Tablespace that you can recover on your own without DB2 recovery. A thoughtful management of DB2 archive logs is important in any DB2 environment: test, development, or production.

Also note that even though you delete the DSNx10.DBxG.ARCLOG1.Date.Time.A000000# data sets, you need to keep enough of the DSNx10.DBxG.ARCLOG1.Date.Time.B000000# data sets to cover two successful restarts of DB2. These data sets are the backups for your BSDS data sets. If anything happens to your BSDS, you need these backups to restore your BSDS.

The DBARCH esoteric is coded into DB2 ZPARM by the link-edit JCL member DSNTIJUZ in DSNx10.NEW.SDSNSAMP. You can find the following parameters coded under the DSN6ARVP section of DSNTIJUZ. If you want to create and update with a different esoteric other than the one provided, you can find the information here.

```
TSTAMP=YES, X
UNIT=DBARCH, X
UNIT2=
```

To change an esoteric, you need to update the IODF file. The default active IODF in the ADCD system is SYS1.IODF99. When you modify it, a temporary work IODF is created, which can be named as your preference, such as SYS1.IODF03.WORK. When the changes are made, you can use this work file to create a production IODF file, such as SYS1.IODF03. After you change the loadparm to use SYS1.IODF03, you can re-LPI the system to check if the changes you made are working properly. You can either continue to use SYS1.IODF03 or copy it to SYS1.IODF99. If you choose to copy it to IODF99, you need to backup SYS1.IODF99, and then delete SYS1.IODF99. After you delete SYS1.IODF99, you can copy SYS1.IODF03 to SYS1.IODF99, and update the loadparm member to use SYS1.IODF99.

To add more volumes to the esoteric DBARCH, follow these steps:

1. From the ISPF primary option menu, select option **M.4. Hardware Configuration Tools**.

```
Command ==>
Hardware Configuration

Select one of the following.

0. Edit profile options and policies
1. Define, modify, or view configuration data
2. Activate or process configuration data
3. Print or compare configuration data
4. Create or view graphical configuration report
5. Migrate configuration data
6. Maintain I/O definition files
7. Query supported hardware and installed UIMs
8. Getting started with this dialog
9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.
I/O definition file . . . 'SYS1.IODF99' +
```

2. In the above I/O definition file, type SYS1.IODF99, and then select option 1. Define, modify, or view configuration data.

```
C Define, Modify, or View Configuration Data
S Select type of objects to define, modify, or view data.
1 1. Operating system configurations
 consoles
 system-defined generics
 EDTs
 esoterics
 user-modified generics
2 2. Switches
 ports
 switch configurations
 port matrix
3 3. Processors
 channel subsystems
 partitions
 channel paths
F 4. Control units
I 5. I/O devices
 6. Discovered new and changed control units and I/O devices
```

3. In the **Operating System Configurations List** window, type / to select Config. ID OS390, and then press Enter.

```

Operating System Configuration List Row 1 of 1
Command ==> _____ Scroll ==> PAGE

Select one or more operating system configurations, then press Enter. To
add, use F11.

/ Config. ID Type Gen Description D/R site OS ID
/ OS390 MVS ADCD ZOS IODF
***** Bottom of data *****

```

- When a pop-up window opens as shown below, select option 5. **Work with EDTs.**

```

Actions on selected operating systems
Select by number or action code and press Enter.

1. Add like (a)
2. Repeat (copy) OS configurations . . (r)
3. Change (c)
4. Delete (d)
5. Work with EDTs (s)
6. Work with consoles (n)
7. Work with attached devices (u)
8. View generics by name (g)
9. View generics by preference value . (p)

```

- When a list of EDTs shows, type / to select **EDT# 00**, and then press Enter.

```

- Goto Backup Query Help

C Row 1 of 1
S Command ==> _____ Scroll ==> PAGE
a Select one or more EDTs, then press Enter. To add, use F11.

/ Configuration ID . : OS390 ADCD ZOS IODF
/
* / EDT Last Update By Description
/ 00 2012-05-17 IBMUSER Add new esoterics
***** Bottom of data *****

```

- When another pop-up window opens as below, select option 4. **Work with esoterics.**



```

- Goto Backup Query Help EDT List

C Actions on selected EDTs
S Command =
a Select on
4 1. Repeat (copy) EDTs (r)
 2. Change (c)
 3. Delete (d)
 4. Work with esoterics (s)
 5. Work with generics by name (g)
 6. Work with generics by pref. value . (p)

/ Configura
/
* / EDT Las
 / 00 201

```

7. When the available esoteric list is displayed as shown below, type / to select esoteric **DBARCH**, and then press Enter for further options.

```

- Goto Filter Backup Query Help Esoteric List

C Row 1 of 7
S Command ==> Scroll ==> PAGE
a Select one or more esoterics, then press Enter. To add, use F11.

/ Configuration ID . : 0S390 ADDC ZOS IOBF
/ EDT identifier . . : 00 Add new esoterics
*

/ Esoteric VIO Token State
- DASD Yes 1
/ DBARCH Yes 3
- SORT Yes 5
- SYSOA Yes 7
- TEMP Yes 9
- VIO Yes 11
- WORK Yes 13
***** Bottom of data *****

```

8. In the Actions on selected esoterics window, select option 4. **Assign devices**.

```

- Goto Filter Backup Query Help Esoteric List

C Actions on selected esoterics
S Command =
a Select on
4 1. Repeat (copy) esoterics (r)
 2. Change (c)
 3. Delete (d)
 4. Assign devices (s)
 5. View assigned devices (v)

/ Configura
/ EDT ident
* / Esoteri
 _ DASD
 / DBARCH
 _ SYSOA
 _ VIO Yes
***** Bottom of data *****

```

9. When the available devices are listed as shown below, select the appropriate device or device range that you want to add to the DBARCH esoteric. For example, to add one more device at address 0AA7, change the status of 'Assigned' column from 'No' to 'Yes' across the device range that you want to add, and also specify **Starting Numbers** and **Number of Devices** to add a range. Then press enter.

```

- Assign/Unassign Devices to Esoteric
- Goto Filter Backup Query Help

C Command ==> _____ Row 11 of 17
S Scroll ==> PAGE
a
Specify Yes to assign or No to unassign.
/ Configuration ID . . : OS390 ADCC ZOS IODF
/ EDT.Esoteric . . . : 00.DBARCH VIO eligible . : No
*
Devices Device Type Generic Assigned Starting Number of
 Name Number Devices
0900,13 3270-X 3277-2 No _____
090E,18 3270-X 3277-2 No _____
0A80,38 3390 3390 No _____
0AA6,1 3390 3390 Yes _____
0AA7,73 3390 3390 Yes 0AA7 1

0E20,4 CTC CTC No _____
0E40,4 CTC CTC No _____
***** Bottom of data *****
To leave the panel press EXIT or CANCEL.

```

- When a pop-up window opens as below, type a name for the temporary work IODF, for example, SYS1.IODF03.WORK, and then press Enter.

```

- Esoteric List
- Goto Filter Backup Query Help
- Add Esoteric

C Create Work I/O Definition File
S
The current IODF is a production IODF and therefore cannot be
/ updated. To create a new work IODF based on the current
/ production IODF, specify the following values.
*
/ IODF name 'SYS1.IODF03.WORK'
- Volume serial number . SDSYS1 *
-
*** Space allocation . . . 1024 (Number of 4K blocks)
Activity logging . . . Yes (Yes or No)
Multi-user access . . No (Yes or No)

```

Then The Assign/Unassigned Devices to Esoteric window is updated to show the list of devices assigned. Because device# 0AA6 is already assigned, it is updated to include two devices in that range: 0AA6 and 0AA7.

```

- Assign/Unassign Devices to Esoteric
- Goto Filter Backup Query Help

C Command ==> _____ Row 11 of 17
S Scroll ==> PAGE
a
Specify Yes to assign or No to unassign.
/ Configuration ID . . : OS390 ADCC ZOS IODF
/ EDT.Esoteric . . . : 00.DBARCH VIO eligible . : No
*
Devices Device Type Generic Assigned Starting Number of
 Name Number Devices
0900,13 3270-X 3277-2 No _____
090E,18 3270-X 3277-2 No _____
0A80,38 3390 3390 No _____
0AA6,2 3390 3390 Yes _____
0AA7,72 3390 3390 No _____
0E20,4 CTC CTC No _____
0E40,4 CTC CTC No _____
***** Bottom of data *****

```

- Return to the primary option menu of HCD, type SYS1.IODF03.WORK in the I/O definition file field, and then select option 2. **Activate or process configuration data.**

```

Command ==> _____

Hardware Configuration

Select one of the following.

2 0. Edit profile options and policies
1. Define, modify, or view configuration data
2. Activate or process configuration data
3. Print or compare configuration data
4. Create or view graphical configuration report
5. Migrate configuration data
6. Maintain I/O definition files
7. Query supported hardware and installed UIMs
8. Getting started with this dialog
9. What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'SYS1.IODF03.WORK' +

```

12. In the Activate or Process Configuration Data window, select option 1. **Build production I/O definition file.**

```

C Activate or Process Configuration Data
Select one of the following tasks.

S 1 1. Build production I/O definition file
2 2. Build IOCDs
3 3. Build IOCP input data set
4 4. Create JES3 initialization stream data
5 5. View active configuration
6 6. Activate or verify configuration dynamically
7 7. Activate configuration sysplex-wide
8 8. *Activate switch configuration
9 9. *Save switch configuration
F 10. Build I/O configuration data
I 11. Build and manage S/390 microprocessor IOCDs and IPL attributes
12. Build validated work I/O definition file ed.

* = requires TSA I/O Operations +

```

The changes that are made are verified at this stage. Any errors or warnings that are displayed at this stage can be ignored.

```

Message List
Save Query Help

Row 1 of 2
Command ==> _____ Scroll ==> PAGE
Messages are sorted by severity. Select one or more, then press Enter.

/ Sev Msg. ID Message Text
/ W C8DA333I EDT 00 of OS configuration OS390 does not use tokens for
its esoterics.
***** Bottom of data *****

```

13. Press F3 to return to the Activate or Process Configuration Data window, and select option 1. **Build production I/O definition file.**
14. To create the production I/O definition file:
  - a. In the **Work IODF name** field, type SYS1.IODF03.WORK.
  - b. In the **Production IODF name** field, type SYS1.IODF03.
  - c. In the **Volume serial number** field, specify the same volume where IODF99 is.

- d. In the **Continue using as current IODF:** field, specify option 2. The new production IODF specified above.
- e. Press Enter.

```

C Activate or Process Configuration Data
S Build Production I/O Definition File
S 1 Specify the following values, and choose how to continue.
2 Work IODF name . . . : 'SYS1.IODFC3.WORK'
 Production IODF name . 'SYS1.IODFC3'
 Volume serial number . SDSYS1
 Continue using as current IODF:
2 1. The work IODF in use at present
2 2. The new production IODF specified above
F
I * = requires TSA I/O Operations

```

15. When a pop-up window shows as below, specify values for descriptor fields 1 and 2, and then press Enter.

```

C Activate or Process Configuration Data
S Build Production I/O Definition File
S 1 Specify the following values, and choose how to continue.
2 Work IODF name . . . : 'SYS1.IODFC3.WORK'
 Production IODF name . 'SYS1.IODFC3'
 Vo Define Descriptor Fields
 Co
2 Specify or revise the following values.
 Production IODF name . : 'SYS1.IODFC3'
 Descriptor field 1 . . . SYS1
 Descriptor field 2 . . . IODFC3
F
I * = re

```

When the creation is successful, the message Production IODF SYS1.IODFC3 created. is displayed as below.

```

C Activate or Process Configuration Data
S Select one of the following tasks.
S 1 1. Build production I/O definition file
2 2. Build IOCCDS
 3. Build IOCP input data set
 4. Create JES3 initialization stream data
 5. View active configuration
 6. Activate or verify configuration dynamically
 7. Activate configuration sysplex-wide
 8. *Activate switch configuration
 9. *Save switch configuration
 10. Build I/O configuration data
 11. Build and manage S/390 microprocessor IOCCDSs and IPL attributes
 12. Build validated work I/O definition file
F ed.
I * = requires TSA I/O Operations
 Production IODF SYS1.IODFC3 created.

```

16. Exit HCD, and then create a new loadparm member by using LOADCS as a template as shown in this example. To use IODF03, change the IODF parm value 99 - 03.

```

EDIT SYS1.IPLPARM(LOAD01) - 01.00 Columns 00001 00072
Command ==> _____ Scroll ==> CSR
***** ***** Top of Data *****
000001 IODF 03 SYS1
000002 SYSCAT Z1SYS1113CCATALOG.Z21Z.MASTER
000003 SYSPARM CS
000004 IEASYM 00
000005 PARMLIB USER.PARMLIB Z1SYS1
000006 PARMLIB ADCD.Z21Z.PARMLIB Z1RES1
000007 PARMLIB SYS1.PARMLIB Z1RES1
000008 NUCLEUS 1
000009 SYSPLEX ADCDPL
***** ***** Bottom of Data *****

```

17. IPL the system with loadparm 01 to ensure that IODF03 works without error. If you want to continue using IODF03, modify other loadparms as required. Otherwise, if you want to copy the changes to IODF99, backup IODF99 by using option 6. **Maintain I/O definition files**, and then option 2. **Copy I/O definition file**. Then, delete file IODF99 and repeat options 6 and 2 to copy IODF03 into IODF99.

### Steps for dumping SMF data into GDG data sets

The system is configured to clear SMF data by using the procedure SMFCLEAR. This procedure is triggered by the exit routine IEFU29, and the source code of which is available in ADCD.LIB.JCL. If you want to save the SMF data into GDG data sets, modify this exit routine to call procedure SMFDUMPS. Edit the exit routine IEFU29 to change occurrences of string SMFCLEAR to SMFDUMPS, and assemble or link edit the load module by using the JCL member IEFU290, which is also available in ADCD.LIB.JCL. To take this change effect, you must re-IPL the system. As a GDG base SYS1.SMF.DATA was defined by using the JCL in ADCD.LIB.JCL(SMFGDG), the procedures SMFCLEAR and SMFDUMPS are available in ADCD.Z22C.PROCLIB.

### Using the esoteric devices

Esoteric devices that are called WORK, TEMP, and SORT are defined and mapped to devices in the range 0600 – 060F. To use these esoterics, create emulated 3390 volumes in your zPDT system, and map them to these device addresses by using the devmap file.

Note that you must update the appropriate VATLSTxx member (volume attribute list) with necessary statement, as shown highlighted in the following sample. This example uses volumes WORK01 through WORK12, which are marked as WORK\*. This member defines the mount and uses attributes of direct-access volumes.

```

VATDEF IPLUSE(PRIVATE),SYSUSE(PRIVATE)
C2SYS1,0,0,3390 ,Y
WORK* ,0,0,3390 ,Y

```

### Steps for starting IBM Health Checker for z/OS

With this edition, IBM Health Checker is preconfigured to start automatically.

To start this utility manually, use the following start command.


```
/START HZSPROC
```

To stop IBM Health Checker, enter the following command manually, or include it in your shutdown script SHUTxx in parmlib.

```
/STOP HZSPROC
```

For more information, see Stopping and starting IBM Health Checker for z/OS Manually.

## Related information:

 IBM Health Checker for z/OS

## Steps for starting IBM z/OS Management Facility (z/OSMF)

A new load parm (IZ) is configured to start the z/OSMF product. These required commands to start z/OSMF are listed below for reference. However, these commands are included in the startup script member VTAMCI/DB/IM/IZ/WA/AL so that z/OSMF starts automatically. The first command starts the CIM (Common Information Model) server, and the second command starts z/OSMF. You must start the CIM server before starting the z/OSMF product. The WebSphere Application Server OEM provides a native application server runtime environment for z/OSMF.

```
S CFZCIM
```

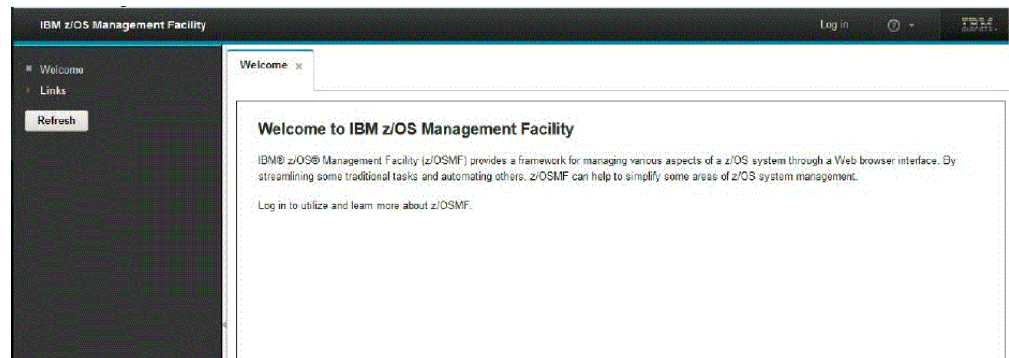
It might take several minutes for these applications to start, which depends on your zPDT system. Refer to the following messages for successful start of the z/OSMF product.

```
S IZUANG1
CWWKB0056I INITIALIZATION COMPLETE FOR ANGEL
```

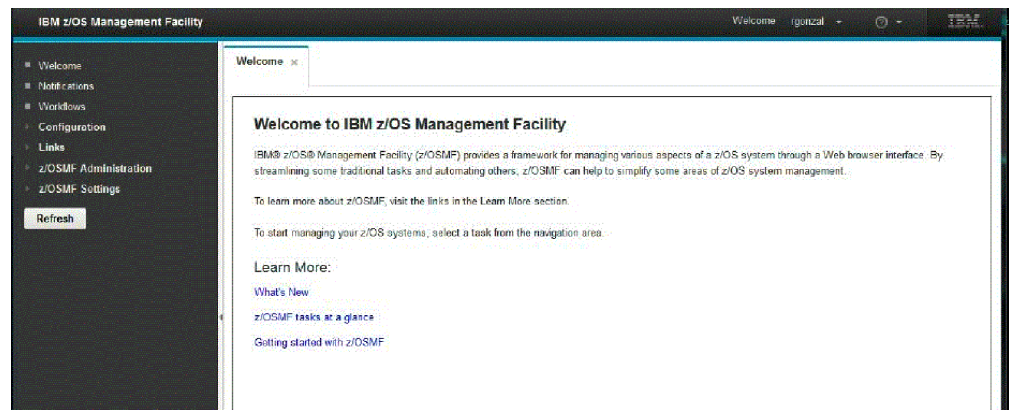
```
S IZUSVR1
IZUG400I: The z/OSMF Web application services are initialized.
+CWWKF0011I: The server zosmfServer is ready to run a smarter planet.
```

To access the z/OSMF console, you can use the URL: <https://s0w1.dal-ebis.ihost.com:10443/zosmf>

The initial login opens as below.



The welcome screen of z/OSMF console opens as below.



To stop z/OSMF, enter these commands:

```
P IZUSVR1
P CFZCIM
P IZUANG1
```

## LOADPARMS Options

Lists available preconfigured distributed LOADPARMS.

**Note:** Start JES2 cold the first time you start the system.

*Table 6. Listing of available preconfigured distributed LOADPARMS.*

| LOADPARM | Description                                                                                                                                                                                            |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CS       | CLPA and Cold start of JES2. Base z/OS system functions, that is, no CICS, DB2, IMS, Websphere Application Server, and so on.                                                                          |
| WS       | CLPA and Warm start of JES2. Base z/OS system functions, that is, no CICS, DB2, IMS, Websphere Application Server, and so on.                                                                          |
| 00       | CLPA and Warm start of JES2. Base z/OS system functions, that is, no CICS, DB2, IMS, Websphere Application Server, and so on.                                                                          |
| CI       | CLPA and Warm start of JES2. Loads CICS 5.3 and 5.2 libraries. Starts CICS 5.3, z/OSMF, and IBM Developer for z Systems.                                                                               |
| DB       | CLPA and Warm start of JES2. Loads DB2 V11 and V10 libraries. Starts DB2 V11, z/OSMF, and IBM Developer for z Systems.                                                                                 |
| IM       | CLPA and Warm start of JES2. Loads IMS 14 and 13 libraries. Starts IMS 14, z/OSMF, and IBM Developer for z Systems.                                                                                    |
| IZ       | CLPA and Warm start of JES2. Starts z/OSMF and IBM Developer for z Systems.                                                                                                                            |
| WA       | CLPA and Warm start of JES2. Loads Websphere Application Server v9 and v8.5 libraries. Starts IBM Developer for z Systems. Websphere Application Server needs to be manually started.                  |
| AL       | CLPA and Warm start of JES2. Loads all middleware libraries. Starts CICS 5.3, DB2 V11, IMS 14, z/OSMF, IBM Developer for z Systems and UCD. Websphere Application Server needs to be manually started. |
| DC       | CLPA and Cold start of JES2. Loads CICS, DB2 libraries. Starts CICS 5.3, DB2 V11, UCD 6.2.0, RTC 6.0.2, IBM Developer for z Systems and UCD.                                                           |
| DW       | CLPA and Warm start of JES2. Loads CICS, DB2 libraries. Starts CICS 5.3, DB2 V11, UCD 6.2.0, RTC 6.0.2, IBM Developer for z Systems and UCD.                                                           |
| AU       | CLPA and Warm Start of JES2. Loads based on the option what you selected during the provision using tools                                                                                              |

## Steps for starting WebSphere Application Server (WAS V8.5)

You are recommended to have more than 4 GB real memory on your system and code at least 3000 m (3 GB) in the devmap. IPL the system with loadparm WA and follow these steps to start WebSphere Application Server V8.5:

1. In SDSF, type / and press Enter to open the System Command Extension window.
2. To start the Deployment Manager, enter this command in the System Command Extension window: START XFDCR,JOBNAME=XFDMGR,ENV=XFCELL.XFDMNODE.XFDMGR. And then press Enter.

```

System Command Extension

Type or complete typing a system command, then press Enter.

==> START XFDCR,JOBNAME=XFDMGR,ENV=XFCELL.XFDMNODE.XFDMGR
==>

Place the cursor on a command and press Enter to retrieve it.
More: +

=> P IMS11RL1
=> 05,/CHE FREEZE
=> 04,/CHE SNAPQ
=> 03,/DIS LTERM ALL
=> 02,/NRE CHKPT 0 FORMAT ALL
=> P IMS12RL1
=> -DBAG STO DB2

_ Wait 1 second to display responses (specify with SET DELAY)
_ Do not save commands for the next SDSF session

F1=Help F5=FullScr F7=Backward F8=Forward F11=ClearLst F12=Cancel

```

This start command might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000019I  
INITIALIZATION COMPLETE FOR WEBSphere FOR Z/OS CONTROL 478 PROCESS  
XFDMGR.

3. Then, to start the Node Agent, enter this command: START  
XFACR1,JOBNAME=XFAGNT1,ENV=XFCELL.XFNODE1.XFAGNT1.

This command might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000019I  
INITIALIZATION COMPLETE FOR WEBSphere FOR Z/OS CONTROL 346 PROCESS XFAGNT1.

4. Now, to connect to the Admin console, open a browser and use this URL:  
<http://s0w1.dal-ebis.ihost.com:9705/ibm/console>.

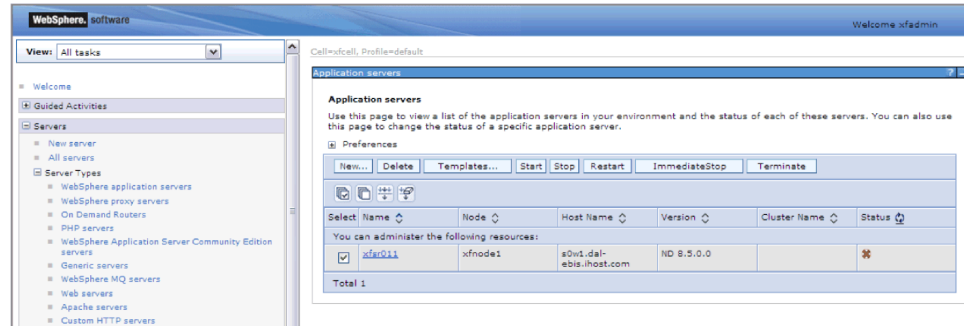
A login window with the User ID **xfadmin** opens as below.



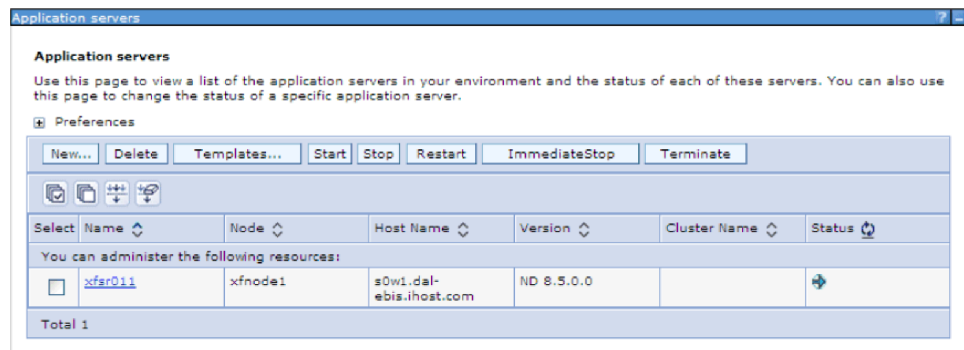
5. Click **Log in**.  
The Admin console opens.



6. To start the Application Server from the Admin console, follow these steps:
  - a. In the navigation list, click **Servers > Server Types > Websphere application servers**.
  - b. In the Application servers view, select the check box beside server **XFSR011**.
  - c. Click **Start**.



This operation might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000019I INITIALIZATION COMPLETE FOR WEBSPPHERE FOR Z/OS CONTROL 541 PROCESS XFSR011. When the server is started, the **Status** column displays a green arrow:



To stop the application server, do one of these actions:

- From the browser:
  1. Select the server, and then click **Stop**.
  2. On the confirmation window, click **OK**.
- From SDSF, enter this command: /P XFSR011.

When the application server is shut down, this message is displayed in the system log: BB000002I WEBSPPHERE FOR Z/OS CONTROL PROCESS XFSR011 ENDED NORMALLY.

To stop the Deployment Manager and Node Agent, enter this command: /P XFDEM1.

The stop command might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000008I WEBSPPHERE FOR Z/OS DAEMON S0W1 ENDED NORMALLY.

## Steps for starting WebSphere Application Server (WAS V9.0)

You are recommended to have more than 4 GB real memory on your system and code at least 3000 m (3 GB) in the devmap. IPL the system with loadparm WA and follow these steps to start WebSphere Application Server V9.0:

1. In SDSF, type / and press Enter to open the System Command Extension window.
2. To start the Deployment Manager, enter this command in the System Command Extension window: START XADCR,JOBNAME=XADMGR,ENV=XACELL.XADMNODE.XADMGR. And then press Enter.

```
-- Edit Options Help --
SD
CO System Command Extension
NO
NCO ==> START XADCR,JOBNAME=XADMGR,ENV=XACELL.XADMNODE.XADMGR
MRO ==>
LR STORELIMIT
LR Comment
DR Group Show * (F4 for list)
DR More: +
DR =>
DR =>
DR =>
DR =>
DR =>
DR =>
DR =>
ER F5=FullScr F6=Details F7=Up F8=Down F10=Save F11=Clear F12=Cancel
NO

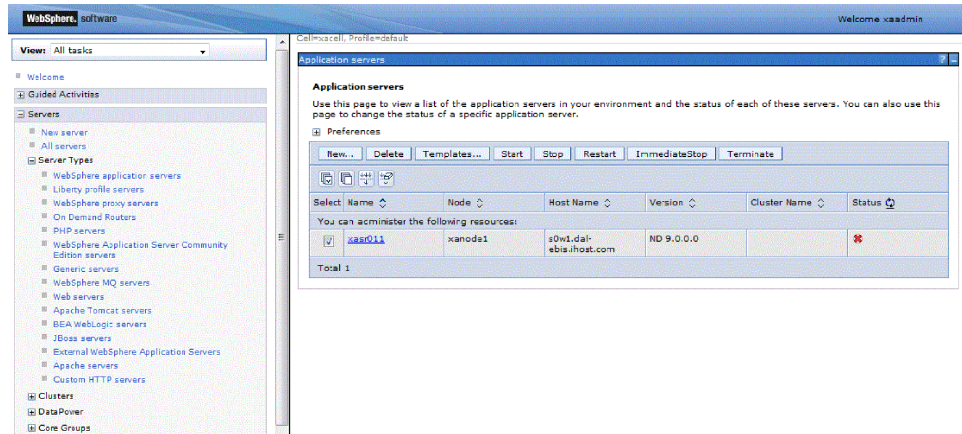
```

The start command might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000019I  
INITIALIZATION COMPLETE FOR WEBSHERE FOR Z/OS CONTROL 478 PROCESS  
XADMGR.

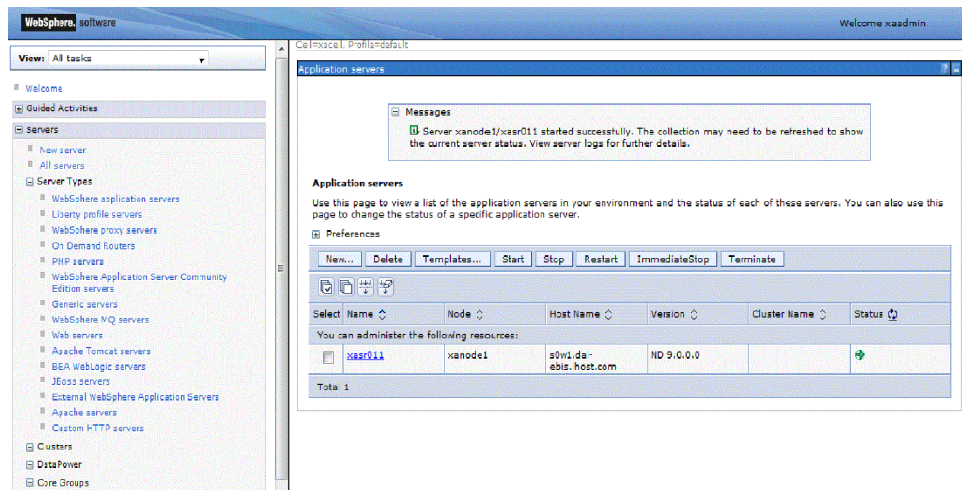
3. To start the Node Agent, enter this command : START XAACR1,JOBNAME=XAAGNT1,ENV=XACELL.XANODE1.XAAGNT1.  
This command might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000019I  
INITIALIZATION COMPLETE FOR WEBSHERE FOR Z/OS CONTROL 346 PROCESS XAAGNT1.
4. To connect to the Admin console, open a browser and use this URL:  
<http://s0w1.dal-ebis.ihost.com:9305/ibm/console>.  
A login window with the User ID **xaadmin** opens as below.



5. Click **Log in**.  
The Admin console opens.
6. To start the Application Server from the Admin console, follow these steps:
  - a. In the navigation list, click **Servers > Server Types > Websphere application servers**.
  - b. In the Application servers view, select the check box beside server **XFSR011**.
  - c. Click **Start**.



This operation might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000019I INITIALIZATION COMPLETE FOR WEBSHERE FOR Z/OS CONTROL 541 PROCESS XASR011. When the server is started, the **Status** column displays a green arrow:



To stop the application server, do one of these actions:

- From the browser:
  1. Select the server, and then click **Stop**.
  2. On the confirmation window, click **OK**.
- From SDSF, enter this command: /P XASR011.

When the application server is shut down, this message is displayed in the system log: BB000002I WEBSHERE FOR Z/OS CONTROL PROCESS XASR011 ENDED NORMALLY.

To stop the Deployment Manager and Node Agent, enter this command: /P XADEMN.

The stop command might take several minutes to complete. When it is completed, this message is displayed in the system log: BB000008I WEBSPHERE FOR Z/OS DAEMON S0W1 ENDED NORMALLY.

---

## Chapter 9. Deploying z Systems Development and Test Environment for Parallel Sysplex and the coupling facility

z Systems Development and Test Environment can be used to enable a Sysplex environment that is running within z/VM. To run a Sysplex environment, you must first purchase entitlement to z Systems Development and Test Environment with Parallel Sysplex. A z/VM distribution for z Systems Development and Test Environment is made available to customers entitled to this product. You then must obtain, install, and configure the z/VM distribution and the z/OS guests that run on it.

Configuring and using z/PDT in an IBM Parallel Sysplex environment with a coupling facility is complex. The following sections describe only the basics of licensing and starting a z/VM environment on which the sysplex environment can be built. For detailed instructions on configuring your system to use the coupling facility, see Chapter 9, “Other z Systems Operating Systems”, and Part 5, “IBM Parallel Sysplex” in the zPDT Guide and Reference and the zPDT 2016 Sysplex Extensions.

To enable the coupling facility after you purchase entitlement to z Systems Development and Test Environment with Parallel Sysplex, do the steps that are described in these topics. For reference information about the z/VM 6.3 distribution that is supplied with z Systems Development and Test Environment, see “z/VM 6.3 for z Systems Development and Test Environment notes” on page 187. For a list of the products that are contained in the z/VM 6.3 distribution and a list of z/VM user IDs and passwords available with the product, see “Products contained in this release” on page 188.

---

### Installing z Systems Development and Test Environment with Parallel Sysplex and z/OS guests

Before you install this z/VM 6.3 distribution, you must first install IBM z Systems Development and Test Environment with Parallel Sysplex and any z/OS distributions that run as guests on this z/VM system and in the sysplex. You can install either your own migrated z/OS distribution or a z/OS ADCD for z Systems Development and Test Environment.

The installation of the Parallel Sysplex product and the z/OS guests has the same initial steps for the sysplex and non-sysplex offerings of the product.

- Do the steps in Chapter 4, “Installing z System Development and Test Environment,” on page 37.
- Do the steps in Chapter 5, “Enabling product operation,” on page 61. The license key must have Parallel Sysplex capability. Beginning with z Systems Development and Test Environment with Parallel Sysplex V10.0, software-based licensing must be used to run Parallel Sysplex.
- Do the steps in “Installing an operating system on z Systems Development and Test Environment” on page 55 for the z/OS guests that run in the sysplex.
- Do the steps in Chapter 6, “Configuring an instance of z Systems Development and Test Environment,” on page 103.

---

## Downloading a z/VM distribution

The coupling facility feature is available to z/OS only when z/OS is running within virtual machines under z/VM. All the images that comprise z/VM 6.3 are available if you have entitlement to IBM z Systems Development and Test Environment with Parallel Sysplex.

For the steps to download z/VM 6.3 for z Systems Development and Test Environment, select the appropriate version tab and follow the instructions in the z Systems Development and Test Environment download information technote. All volume images that are downloaded from Passport Advantage are compressed (gzip) files in the emulated 3390 format compatible with zPDT. Volumes from your z/OS distribution can then be added to create as many virtual z/OS systems as needed. For specific instructions on downloading available z/VM distributions, and the customizations made available with it, such as z/VM user IDs, see *z/VM for IBM Rational Development and Test Environment for z Systems Installation and Configuration (SC27-4618)*, available by following the instructions in the download information technote.

A sample Parallel Sysplex configuration is available for download through Passport Advantage. This sample system, which is supplied on two DVDs, consists of a z/VM 6.2 system that hosts two virtual coupling facilities and two z/OS 1.13 systems. For information on how to obtain this media pack through Passport Advantage, see the download information technote.

---

## Enabling the coupling facility

To enable the coupling facility, do these steps.

### Enabling the coupling facility in the device map

Configuration of each instance of z Systems Development and Test Environment is done through a file that is called a device map or devmap. The device map can be created manually or use the `create_devmap.pl` sample that is included with z Systems Development and Test Environment to create the device map semi-automatically.

To enable the use of the coupling facility, you must modify your device map:

#### Modifying the [system] stanza

In the [system] stanza, add this line: `cpuopt zvm_couplingfacility` .

If you start a z Systems Development and Test Environment instance with `cpuopt zvm_couplingfacility` specified in the [system] stanza and are unable to obtain a license, your USB hardware device might not be activated to enable the coupling facility.

Any instance that is defined with `cpuopt zvm_couplingfacility` in the device map requires one coupling facility license from the USB hardware device for each CP in the instance. If `cpuopt zvm_couplingfacility` is not specified in the device map, then a standard, non-coupling license activation is required from the USB hardware device.

Device maps that include the `cpuopt zvm_couplingfacility` definition can define up to eight processors in the [system] stanza. The number of processors cannot exceed the number of physical cores available on your system.

**Tip:**

The `cpuopt zVM_coupling` facility definition affects the output of the **token** command. In particular, if `cpuopt zVM_coupling` facility is not specified in the devmap, the **token** command does not display a license for sysplex activation, even if an update file with parallel sysplex support enabled was successfully installed on your USB hardware device. Ensure that your devmap coupling facility setting matches the type of license activation on your USB hardware device before you perform any operations that use the USB hardware device.

**Adding z/VM volumes to an [awsckd] stanza**

You must add the z/VM volumes to an `awsckd` stanza in your device map. These volumes must be in addition to your z/OS volume definitions in the device map. Typically, the z/VM volumes are assigned device addresses in a different range from the z/OS volumes. Following is a sample of the volume images for z/VM.

```
device 0200 3390 3390 /home/ibmsys1/z1090/disks/M01RES
device 0201 3390 3390 /home/ibmsys1/z1090/disks/630RL1
device 0202 3390 3390 /home/ibmsys1/z1090/disks/630RL2
device 0203 3390 3390 /home/ibmsys1/z1090/disks/M01W01
device 0204 3390 3390 /home/ibmsys1/z1090/disks/M01S01
device 0205 3390 3390 /home/ibmsys1/z1090/disks/M01P01
device 0206 3390 3390 /home/ibmsys1/z1090/disks/VMCOM1
device 0207 3390 3390 /home/ibmsys1/z1090/disks/VMCOM2
device 0208 3390 3390 /home/ibmsys1/z1090/disks/M01W02
device 0209 3390 3390 /home/ibmsys1/z1090/disks/M01W03
```

In addition to these changes, you might need to add more 3270 device addresses for consoles to your `[aws3274]` stanza. z/VM requires a console, in addition to needing an MVS console for each z/OS guest, and 3270 device addresses for any TSO sessions.

**Starting IBM z Systems Development and Test Environment and IPLing z/VM**

To start IBM z Systems Development and Test Environment, issue `awsstart` to under `ibmsys1` user id or equivalent home by using directory structure `/home/ibmsys1`.

**Restriction:** You cannot use the `runzpd` script when you perform a sysplex activation.

After you start the zPDT emulator with `awsstart`, at a minimum start the 3270 sessions you will use for the z/VM console and each z/OS system. This sample command starts these 3270 sessions from a Linux console by using `x3270`

```
x3270 -port 3270 <session_name_in_devmap>@localhost &
```

**Tip:** 3270 emulators can also be used for the z/VM and MVS console that are installed remotely, and remove the need to have a 3270 emulator on the native Linux. For more information, see “IPLing z Systems Development and Test Environment from a remote emulated terminal for the system console” on page 109.

IPL the z/VM system by entering this command, where address 200 is the device address in the devmap of the system residence volume for z/VM (M01RES), and address 700 is the device address of the console to use.

```
$ ipl 200 parm 0700
```

The initial IPL might produce the Stand-alone Program Loader window, or it might go directly to the OPERATOR session, depending on z/VM customization. The Stand-alone Program loader requires that you enter the z/VM console address (700 in the example) and the z/VM configuration volume (parameter pdvol, which with the z/VM for z Systems Development and Test Environment with Parallel Sysplex is set to volume VMCOM1). Other IPL parameters might need to be set. For more information, see "IPL and logon" in the zPDT Guide and Reference.

During z/VM startup, you are prompted to "warm start" or "cold start" the system. Use warm start unless you have a specific reason for cold-starting z/VM. Reply warm when prompted.

This 3270 session now becomes the z/VM master console, also known as the OPERATOR session.

**Tip:**

- Do not confuse z/VM cold start with the z/OS cold start function.
- When z/VM is IPLing, if the **More** indicator is shown, use the clear button to scroll to the next page.

## IPLing the z/OS guests

After z/VM is IPLed, you can IPL a z/OS guest. Since each z/OS guest is started from a 3270 session that is logged in with a valid z/VM user ID, start this process by logging in as a z/VM user from one of your 3270 sessions. If the 3270 sessions are active, they will switch to the z/VM logo display after z/VM was IPLed. If they do not, enter the ENABLE ALL command from the z/VM operator console. After the z/VM logo appears, log in with a valid z/VM user ID and password. Then, enter the TERM CONMODE 3270 command to place the console in 3270 mode. Now you can IPL the z/OS guest. From the command line of the z/VM 3270 session, enter an IPL command similar to the following example.

```
$ ipl 0A80 loadparm 0A82xx
```

0A80 is the device address of the system volume for z/OS, 0A82 is the device address of the volume that contains the SYS1.IPLPARM loadparms, and xx is the two character load parm to be IPLed. While the command syntax is slightly different from the IPL command that IPLs z/OS without z/VM, the device addresses have the same meaning, and must match the device addresses of the z/OS system in your devmap. This 3270 session becomes the z/OS console.

**Note:** You cannot run the script runzpdtd to start z/OS in this environment

Multiple guests can be started by logging on as a different z/VM user, and IPLing another z/OS system.

## Starting a TSO session

You start a TSO session to a z/OS guest from an active 3270 session that is connected to z/VM.

From the 3270 session with the z/VM logo display, in the command field enter this command:

```
Dial <userid>
```

where <userid> is the z/VM userid used to IPL the z/OS guest.



This 3270 session is now connected to the z/OS guest system, and the standard z/OS UNIX System Services message prompt to log in to TSO is displayed.

## Configuring a working z/OS 2.2 ADCD

You can customize any z/OS distribution that runs as a guest on z/VM.

For the steps to customize a z/OS ADCD, see “Customizing the May 2017 Edition z/OS 2.2 ADCD for z Systems Development and Test Environment” on page 124. If you are importing your own z/OS distribution, see “Customizing a migrated z/OS system” on page 111.

## Shutting down all systems and stopping IBM z Systems Development and Test Environment

Follow these steps to properly shut down all operating systems and stop IBM z Systems Development and Test Environment.

- From each z/OS master console, shut down the z/OS system as described in “Stopping the z Systems Development and Test Environment” on page 128.
- Issue the QUIESCE command from each z/OS master console after all z/OS systems are stopped. This command returns each session to the z/VM user ID session that you originally logged in to.
- Issue the logoff command from all z/VM user IDs except the z/VM master console
- From the z/VM master console, enter the SHUTDOWN command
- After this command completes, from the Linux console where you performed the original awsstart, issue awsstop

---

## z/VM 6.3 for z Systems Development and Test Environment notes

The z/VM 6.3 distribution provides a base z/VM 6.3 operating system with software maintenance. A sample Parallel Sysplex configuration is not available for this z/VM 6.3 distribution. However, a sample Parallel Sysplex system based on a z/VM 6.2 distribution is available for download through Passport Advantage. This sample system, which is supplied on two DVDs, consists of a z/VM 6.2 system that hosts two virtual coupling facilities and two z/OS 1.13 systems.

For information on how to obtain this media pack through Passport Advantage, see z Systems Development and Test Environment download information technote .

**Note:** The z/VM 6.3 functions of Single System Image (SSI) and Live Guest Relocation (LGR) were not tested on the zPDT and are not supported.

### Required Prerequisites

To use the z/VM 6.3 distribution, you need to be at the zPDT Version 1 Release 6 level of zPDT code that is shipped with z Systems Development and Test Environment v9.5.

### User IDs

The user IDs that are configured with this distribution have the initial password SSI1, except for BASEAD, S0W1, S0W2, and so on. Each has an initial password that is the same as the user name. You can see which user IDs are configured by logging on as the MAINT user ID, and browsing the z/VM directory named USER DIRECT.

## Products contained in this release

Lists the products contained in this release.

| Component ID | Release | Component Name                                           | Product ID |
|--------------|---------|----------------------------------------------------------|------------|
| 568411201    | 630     | Conversational Monitoring System (CMS)                   | 6VMCMS30   |
| 568411202    | 630     | Control Program (CP)                                     | 6VMCPR30   |
| 568411204    | 630     | AVS                                                      | 6VMAVS30   |
| 568411205    | 630     | REstructured eXtended eXecutor (REXX)                    | 6VMREX30   |
| 568411206    | 630     | Transparent Services Access Facility (TSAF)              | 6VMTSA30   |
| 568411208    | 630     | Dump Viewing Facility (DVF)                              | 6VMDVF30   |
| 568411209    | 630     | VM Serviceability Enhancements Staged/Extended (VMSES/E) | 6VMSES30   |
| 568411211    | 630     | Group Control System (GCS)                               | 6VMGCS30   |
| 568411220    | 620     | Language Environment (LE)                                | 6VMLEN20   |
| 568819802    | 620     | LE COBOL RTL                                             | 6VMLEN20   |
| 568819803    | 620     | LE PL/I RTL                                              | 6VMLEN20   |
| 568411201    | A63     | CMS JAPANESE KANJI                                       | 6VMCMS30   |
| 568411202    | A63     | CP JAPANESE KANJI                                        | 6VMCPR30   |
| 568411204    | A63     | AVS JAPANESE KANJI                                       | 6VMAVS30   |
| 568411205    | A63     | REXX JAPANESE KANJI                                      | 6VMREX30   |
| 568411206    | A63     | TSAF JAPANESE KANJI                                      | 6VMTSA30   |
| 568411208    | A63     | DVF JAPANESE KANJI                                       | 6VMDVF30   |
| 568411209    | A63     | VMSES/E JAPANESE KANJI                                   | 6VMSES30   |
| 568411211    | A63     | GCS JAPANESE KANJI                                       | 6VMGCS30   |
| 568411220    | A62     | LE JAPANESE KANJI                                        | 6VMLEN20   |
| 568819802    | A62     | LE COBOL RTL JAPANESE KANJI                              | 6VMLEN20   |
| 568819803    | A62     | LE PL/I RTL JAPANESE KANJI                               | 6VMLEN20   |

| Component ID | Release | Component Name                                                    | Product ID |
|--------------|---------|-------------------------------------------------------------------|------------|
| 568411219    | 620     | HCD and HCM for z/VM                                              | 6VMHCD20   |
| 565510400    | 440     | Open Systems Adapter/Support Facility (OSA/SF) for VM             | 4OSASF40   |
| 5735FAL00    | 630     | Transmission Control Protocol/Internet Protocol (TCP/IP) for z/VM | 6VMTCP30   |
| 568411218    | 630     | Performance Toolkit for VM                                        | 6VMPTK30   |
| 5749DVH00    | 630     | DirMaint™                                                         | 6VMDIR30   |
| 576700201    | 630     | Resource Access Control Facility (RACF)                           | 6VMRAC30   |
| 568409601    | 630     | RSCS                                                              | 6VMRSC30   |
| 569623401    | 36A     | VM HLASM TOOLKIT                                                  | 5696234K   |
| 569623400    | 360     | VM HILEVEL ASSMBLR                                                | 5696234J   |
| 5654A2200    | 130     | C/C++ COMP Z/VM                                                   | 5654A22A   |
| 569501304    | 140     | REXX COMPILER                                                     | 5695013H   |
| 569501404    | 140     | REXX LIBRARY                                                      | 5695014H   |



---

## Chapter 10. Troubleshooting and support

Learn how to troubleshoot the product.

---

### Finding known problems

Known problems are documented in the form of individual technotes in the Support knowledge base.

As problems are discovered and resolved, the IBM Support team updates the knowledge base at the IBM Support Portal. By searching the knowledge base, you can quickly find workarounds or solutions to problems. The following link runs a customized query of the live Support knowledge base: View all known problems for IBM z Systems Development and Test Environment.

For a complete listing of the currently available technotes, see IBM z Systems Development and Test Environment Tech Notes®.

---

### IBM Software Support

Learn how to contact IBM Software Support for help with problems, fixes, and other support information.

For contact information and guidelines or reference materials, read the *IBM Software Support Handbook*.

For FAQs, lists of known problems and fixes, and other support information, see the product support page for IBM z Systems Development and Test Environment.

For product news, events, and other information, see the Multiplatform development home page.

Before you contact IBM Software Support, gather the background information that you need to describe your problem. When you describe a problem to an IBM software support specialist, be as specific as possible and include all relevant information so that the specialist can help you solve the problem. To save time, know the answers to these questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, or messages that are related to the problem?
- Can you reproduce the problem? If so, what steps do you take to reproduce it?
- Is there a workaround for the problem? If so, be prepared to describe the workaround.

For answers to "how to" questions, see the z Systems Development and Test Environment Forum.

---

### Obtaining the latest editions of Redbooks

IBM Redbooks® provide information about major components of IBM z Systems Development and Test Environment.

## IBM z Systems Development and Test Environment

To obtain the latest edition of the *zPDT Guide and Reference (SG24-8205)* for use with IBM z Systems Development and Test Environment, go to the IBM Redbooks website and search for *zPDT Guide and Reference (SG24-8205)*.

### Troubleshooting tips

Table 7. Troubleshooting tips

| Symptom                                                                                                                                                                                                                            | Problem Determination                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| For any problem after installation of z Systems Development and Test Environment (creating a devmap, installing z/OS, starting or stopping z Systems Development and Test Environment, z/OS networking issues, performance issues) | Validate the environment and the installation by running the <code>z1090instcheck</code> command and fixing errors. Do not worry about errors that are related to <code>kernel.core_pattern</code> unless the emulator is terminating abnormally.                                                                                |
| AWSccnnns message that is issued from the Linux terminal that is running z Systems Development and Test Environment.                                                                                                               | <p>Any message beginning with AWS is a zPDT message. Enter the following zPDT command</p> <pre>msgInfo message-number</pre> <p>Where message-number is the 10 character message. This command displays more information about the reason for the message. See 4.1.36, “The msgInfo command” in the zPDT Guide and Reference.</p> |

### Installing z Systems Development and Test Environment

Table 8. Troubleshooting the Installation of the z Systems Development and Test Environment

| Symptom                                                                                                                                                                                                         | Problem Determination                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Message CRIMA1076E in the install log when you are uninstalling a version of z Systems Development and Test Environment                                                                                         | For the step-by-step process to perform if an uninstall fails., see “Uninstalling a previous version of z Systems Development and Test Environment” on page 46.                                                                                                                                                                                                                                                                                                                                                                           |
| When you are attempting to run <code>launchpad.sh</code> or <code>launchpad-console.sh</code> , you get the following error:<br><pre>./InstallerImage_linux.gtk.x86_64/<br/>tools/imcl: Permission denied</pre> | <p>This issue can happen for several reasons:</p> <ul style="list-style-type: none"><li>• You are attempting the launchpad command without running as root. Ensure that you are executing this command as root</li><li>• You copied the installation disc to a location, and then attempted executing the launchpad commands from that new location. Sometimes when you copy files in Linux, the execution permission bits are not retained during the copy. Ensure that the particular launch command is marked as executable.</li></ul> |

*Table 8. Troubleshooting the Installation of the z Systems Development and Test Environment (continued)*

|                                                                                                                                                                                                                                                                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>When you are attempting to perform an Installation Manager update for z Systems Development and Test Environment, or to run <code>launchpad.sh</code> or <code>launchpad-console.sh</code>, you get one of the following errors:</p> <pre>sntl-sud-xxxx is required     by z1091-1-xxxx but is already     installed</pre> <p>Or</p> <pre>zpdt-shk-server-xxxx is     required by z1091-1-xxxx     but is already installed</pre> | <ul style="list-style-type: none"> <li>• If these messages are from an attempt to migrate to a new version of z Systems Development and Test Environment by performing an Update in Installation Manager, perform an uninstall of the previous version instead, followed by a new installation. For the step-by-step process to perform an uninstall, see “Uninstalling a previous version of z Systems Development and Test Environment” on page 46.</li> <li>• If these messages are from running <code>launchpad.sh</code> or <code>launchpad-console.sh</code>, perform an uninstall of the previous version before you attempt a new installation. For the step-by-step process to perform an uninstall, see “Uninstalling a previous version of z Systems Development and Test Environment” on page 46.</li> </ul>                                                                                                                                                                                                                                                                                 |
| <p>You receive any error message when you are uninstalling a previous version of z Systems Development and Test Environment using the steps outlined in “Uninstalling a previous version of z Systems Development and Test Environment” on page 46.</p>                                                                                                                                                                              | <ol style="list-style-type: none"> <li>1. If you are uninstalling Version 9.0 or 9.1, run the <code>z1091_removeall</code> command <ul style="list-style-type: none"> <li>• Start a Linux console as user root.</li> <li>• Change Directory (<code>cd</code>) to <code>/usr/z1090/bin</code>, the folder that contains the <code>z1091_removeall</code> command.</li> <li>• Run the command as follows: <pre>./z1091_removeall</pre> </li> <li>• Reply <b>Yes</b> to any prompts that ask if you are sure that you want to uninstall</li> </ul> </li> <li>2. If you still get some form of error messages, such as failure to uninstall the Sentinel Keys Server (<code>zpdt-shk-server</code>) or Sentinel System Driver (<code>sntl-sud</code>), attempt these steps in the following order: <ul style="list-style-type: none"> <li>• Enter <code># rpm -e --allmatches z1090</code></li> <li>• Enter <code># rpm -e --allmatches z1091</code></li> <li>• Enter <code># rpm -e --allmatches zpdt-shk-server</code></li> <li>• Enter <code># rpm -e --allmatches sntl-sud</code></li> </ul> </li> </ol> |

*Table 8. Troubleshooting the Installation of the z Systems Development and Test Environment (continued)*

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                   |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| When an installation fails for the z Systems Development and Test Environment License Manager, and the log displays the following entries:<br>error: Failed dependencies:<br>libc.so.6 is needed by aksusbd-7.40-1.i386<br>libc.so.6(GLIBC_2.0) is needed by aksusbd-7.40-1.i386<br>libc.so.6(GLIBC_2.1) is needed by aksusbd-7.40-1.i386<br>libc.so.6(GLIBC_2.2) is needed by aksusbd-7.40-1.i386<br>libc.so.6(GLIBC_2.3.4) is needed by aksusbd-7.40-1.i386<br>libc.so.6(GLIBC_2.4) is needed by aksusbd-7.40-1.i386<br>libpthread.so.0 is needed by aksusbd-7.40-1.i386<br>libpthread.so.0(GLIBC_2.0) is needed by aksusbd-7.40-1.i386<br>libpthread.so.0(GLIBC_2.1) is needed by aksusbd-7.40-1.i386<br>libpthread.so.0(GLIBC_2.3.2) is needed by aksusbd-7.40-1.i386 | 32-bit versions of the Linux glibc libraries are not installed. Install the appropriate 32 bit glibc library for your Linux system and try again. |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|

## USB Hardware device activation and licensing

Troubleshooting the USB Hardware device activation and licensing

*Table 9. USB Hardware device activation and licensing. Troubleshooting USB Hardware device activation and licensing*

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Symptom</b>               | SecureUpdateUtility or Z1091_token_update does not apply the update file to the USB hardware device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Problem Determination</b> | <ul style="list-style-type: none"> <li>Ensure that the last 5 hexadecimal digits of the serial number that is etched on the USB hardware device match the serial number in the file name of the update file you are applying. Sometimes the etched number is hard to read. If you are unable to read the etched serial number, IBM support can assist you with an alternative method of obtaining the serial number.</li> <li>Ensure that the hardware device is recognized by your Linux system by plugging in the device and issuing the lsusb command. If the device is available, you see an entry that shows "Rainbow Technologies Inc".</li> </ul> |
| <b>Symptom</b>               | Unit not found message that is issued during the SecureUpdateUtility -u or the z1091_token_update command                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Problem Determination</b> | <p>This issue can happen for several reasons:</p> <ul style="list-style-type: none"> <li>You try to apply an update file with one serial number to a USB hardware device with a different serial number.</li> <li>You try to update a USB hardware device from a client machine that is connected to a product license server that has no local USB Hardware device</li> </ul> <p>Ensure the update file that is being applied was generated for the correct USB Hardware Device. Ensure that the USB Hardware device is securely plugged in to the local device or remote product license server as is appropriate.</p>                                 |



Table 9. USB Hardware device activation and licensing (continued). Troubleshooting USB Hardware device activation and licensing

|                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Symptom</b> | <p>When you are starting z Systems Development and Test Environment, (awsstart) you get any of the following messages at the Linux terminal that indicates a failure to get a license:</p> <ul style="list-style-type: none"> <li>•             <div style="margin-left: 20px;">                 Error: Failed to get license.<br/>                 Return code: 312             </div> </li> <li>• Error: Unable to get expected license: xxxx</li> </ul> <p>or</p> <p>Error: Unable to locate expected license: xxxx</p> <ul style="list-style-type: none"> <li>•             <div style="margin-left: 20px;">                 AWSEMI315E zPDTA License Unavailable<br/>                 for CPU 0<br/>                 AWSEMI005I Waiting for 1090 license             </div> </li> <li>•             <div style="margin-left: 20px;">                 CPU Address Out Of Range             </div> </li> <li>•             <div style="margin-left: 20px;">                 SFNTGetLicense failed.<br/>                 Return code: 312             </div> </li> </ul> |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Table 9. USB Hardware device activation and licensing (continued). Troubleshooting USB Hardware device activation and licensing

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem Determination</b> | <p>This issue can happen for several reasons:</p> <ul style="list-style-type: none"> <li>• The local USB Hardware device is not plugged in properly. Attempt to “reseat” the USB Hardware device. Unplug the device for 15 seconds, then reinsert it.</li> <li>• The correct update file is not applied to the matching USB Hardware device, or it did not apply successfully. <ul style="list-style-type: none"> <li>– Ensure the update file that is requested and applied has the same serial number within the file name as the serial number of the USB Hardware device. Serial numbers can be hard to read. Look carefully.</li> <li>– Ensure the SecureUpdateUtility -u or Z1091_token_update -u command responded with a "Success" message.</li> </ul> </li> <li>• The requested license type is not correct. <ul style="list-style-type: none"> <li>– Validate that if your device map includes the <code>cpuopt zvm_couplingfacility</code> directive, your USB hardware key includes sysplex enabling licenses. If no sysplex license is available, your device map cannot enable sysplex operation.</li> <li>– Validate that if your device map includes the <code>RDTSERVER=</code> directive or you have an <code>RDTSERVER</code> environment variable set to enable participation in a Rational Token licensing system, your USB hardware key includes token enabling licenses. If token enabling licenses are not available on your USB hardware key, your z Systems Development and Test Environment instance cannot participate in a Rational Token licensing system.</li> </ul> </li> <li>• The system clock is incorrect on the system with the USB hardware device plugged in to it. Ensure that the system clock is correct. Never set the system clock to a future date or time when the USB hardware key is plugged in or you will damage the USB hardware key.</li> <li>• You are trying to use a local USB Hardware device, but your client configuration points to a product license server <ul style="list-style-type: none"> <li>– From a Linux terminal that is running as root, enter the <code>clientconfig</code> command that is found in <code>/usr/z1090/bin</code>. Ensure the License ContactServer is <code>localhost</code> if you are attempting to use a local USB. No other fields need be completed.</li> </ul> </li> <li>• The <code>/usr/z1090/bin/sntlconfig.xml</code> file is not readable by the programs that need it. Verify that <code>/usr/z1090/bin/sntlconfig.xml</code> is "world readable". If it is not, use <code>chmod</code> to change the permission bits to 644.</li> <li>• You do not have the correct type of USB Hardware Device. <ul style="list-style-type: none"> <li>– Perform the following command from a Linux terminal that is running as root: <pre>rpm -qa   grep z109</pre> <p>If Rational Development and Test Environment for z System is properly installed, you should see an rpm of the form <code>z1091-xxx</code> and not <code>z1090-xxx</code></p> </li> <li>– Ensure the USB Hardware device that you are using is a Rational Development and Test Environment for z System Hardware device, and not a z1090 hardware device. A z1090 hardware device has a metal cable with a plastic tag attached to the end of the device. A z1091 hardware device does not.</li> </ul> </li> </ul> |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem Determination</b><br>(continued) | <ul style="list-style-type: none"> <li>• You do not have connectivity from the instance that is performing the awsstart to the product license server. <ul style="list-style-type: none"> <li>– Enter a UIMCHECK on the client instance to verify whether there is proper connectivity. You see the message<br/>The remote QUERY command executed successfully</li> <li>– If this procedure does not complete successfully, try the following procedure. <ul style="list-style-type: none"> <li>- From a Linux terminal that is running as root, enter the clientconfig command that is found in /usr/z1090/bin. Ensure the License ContactServer is the IP address of the product license server, that the License PortNumber is 9450, and that License ipv6 is yes or no as is appropriate for your installation.</li> <li>- From a Linux terminal on the instance that is running z Systems Development and Test Environment, enter a ping to the IP address configured as the license server. If the ping command fails, you have an IP connectivity problem or the server is not started. <ul style="list-style-type: none"> <li>• Ensure that the server was started once by a non-root ID with the uimserverstart command.</li> </ul> </li> <li>- If you believe that there is connectivity to the network, from a Linux terminal that is running as root, check the firewall configuration by entering the following command<br/>iptables -L -n</li> </ul> </li> </ul> </li> </ul> <p>Ensure ports 9450 and 9451 are allowed. For information on how to configure the firewall to allow the appropriate ports, see Firewall considerations in “Configuring the base Linux system” on page 42.</p> <ul style="list-style-type: none"> <li>- If this installation is a newly installed product license server, check the firewall configuration on the product license server. If security is being used on the product license server, ensure that the sentinel configuration file on the server allows for the IP address of the client. This file is in</li> <li>- /opt/safenet_sentinel/common_files/<br/>sentinel_keys_server/sntlconfigsrv.xml</li> </ul> <p>For more information, see paragraph 8.4.2, “Security” in chapter 8 of the zPDT Guide and Reference.</p> <p>In some cases, it might be helpful to restart the license daemon where the USB key is plugged in. Restarting the license daemon is done as root with the command:</p> <pre>/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart</pre> |
| <b>Symptom</b>                              | <p>When you are starting Rational Development and Test Environment for z Systems, (awsstart) you get the following messages at the Linux terminal that indicate an expired license</p> <pre>CPU 0, No Sentinel License Available 65535 Unknown Error Code CPU 1, No Sentinel License Available 65535 Unknown Error Code CPU 2, No Sentinel License Available 65535 Unknown Error Code Expired License. EXP=3/1/2015  AWSEMI315E zPDTA License Unavailable for CPU x</pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem Determination</b> | A replacement update file must be obtained from the Rational License Key Center when a license expires, and the update file must be applied to the USB hardware device being used. For more information, see “Steps to get the replacement file” in the <i>z Systems Development and Test Environment Activation Guide</i> (SC27-6630).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Symptom</b>               | Time cheat errors                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Problem Determination</b> | <ul style="list-style-type: none"> <li>Time cheat errors occur when the time stored on the USB hardware device is in the future when compared to the system clock of the Linux system it is plugged into. The USB device stores the latest date of the system clock it sees and will not provide licenses until the Linux system clock matches. Verify that the Linux system clock is correct. Have your system clock set to UTC to avoid errors that are caused by semi-annual time changes. There is no way to inspect the time on the USB key. However, you might wait a short time to see whether the Linux clock catches up to the time stored on the USB key. Do NOT set the Linux clock ahead of the actual time. If you suspect that the time stored on the USB key is in the future, contact IBM support.</li> </ul> |

## Troubleshooting installing, operating, and uninstalling the License Manager

Learn how to troubleshoot problems installing, operating, and uninstalling the License Manager.

### Symptom

Uninstalling the License Manager fails.

### Problem Determination

Always stop the aksusbd daemon and perform a **uimserverstop** before attempting to uninstall. If the **uimserverstop** is not performed and the uninstalling operation fails, perform the following commands:

```
rpm -e --allmatches UIM
rpm -e --allmatches aksusbd
```

## Installing z/OS volume images

Table 10. Trouble shooting the Installation of z/OS volume images

| Symptom                                                                                                                                                                                                  | Problem Determination                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>You get the following errors while performing a z1091_ADCD_install command:</p> <pre>Error: Failed to get license.       Return code: 312. Error: Unable to locate expected       license: 0D98</pre> | <p>For information on possible causes, see the same symptom in “USB Hardware device activation and licensing” on page 194.</p> <p>Also this can occur if the USB Hardware device, either local or in a product license server, has been updated with a .upw update file instead of a .zip update file. If you are migrating a z Systems Development and Test Environment instance from version 9 to version 9.1, you must request a new update file from the Rational License Key Center to install the AD CD S1RES1 and SARES1 volumes.</p> |

Table 10. Trouble shooting the Installation of z/OS volume images (continued)

|  |  |
|--|--|
|  |  |
|--|--|

## Defining the z Systems Development and Test Environment machine characteristics (DEVMAP)

Table 11. Defining the z Systems Development and Test Environment machine characteristics (DEVMAP). Troubleshooting defining the z Systems Development and Test Environment machine characteristics (DEVMAP)

| Symptom                                           | Problem Determination                                                                                                                                                                                                   |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Unable to run find_io command at create_devmap.pl | The path to the find_io command is not available. Ensure that you run aws_bashrc from the user ID that runs create_devmap.pl and z Systems Development and Test Environment. This updates the path to the needed files. |

## Starting and stopping z Systems Development and Test Environment

Table 12. Starting and stopping z Systems Development and Test Environment. Troubleshooting the starting and stopping of z Systems Development and Test Environment

| Symptom                                                                                                                                                                   | Problem Determination                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AWSSTA140E – Initialization failure after license obtained                                                                                                                | <p>Ensure that you are starting z Systems Development and Test Environment from the home directory of the user who is created for running z Systems Development and Test Environment.</p> <p>Ensure that you have run the aws_bashrc command while logged on as that user, and from that user's home directory.</p>                                 |
| <p>The following messages occur during startup:</p> <p>AWSECH004S Unable to define RAS/FEDC memory, RC=-3.</p> <p>AWSECH004S Unable to define RAS/FEDC memory, RC=-3.</p> | <p>This message is seen with many awsckd definitions.</p> <p>The Linux kernel setting SHMMNI, which sets the system-wide maximum number of shared memory segments, is probably too low. The default is 4096.</p> <p>Try increasing this setting by adding kernel.shmmni = 8192 to /etc/sysctl.conf and activate it using the command sysctl -p.</p> |
| <p>The following message occurs when iplimg z/VM 6.2</p> <p>“CP requires hardware features not available on this processor”</p>                                           | Older z/VM V6R2 maintenance levels may not work with zPDT Version 1 Release 6. Either migrate to z/VM V6R3 or ensure PTF VM65007 is applied to your z/VM V6R2 system.                                                                                                                                                                               |

## z/OS networking issues

Table 13. z/OS networking issues. Troubleshooting z/OS networking issues

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Symptom</b>               | General z/OS networking configuration errors                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Problem Determination</b> | <ul style="list-style-type: none"><li>• Check syslog (z/OS console messages) for errors during the start of TCP/IP or VTAM.</li><li>• Validate configuration files match for device numbers and names in the device map, VTAM definitions, and TCP/IP configuration profile. Specifically, follow the definitions to ensure that the tunnel and external addresses in the device map are defined through VTAM and TCP/IP to be the correct device names and addresses in z/OS. These configuration errors can be tricky because device names in the TCP/IP profile are arbitrary.</li></ul>                                                                                                                                                                                                                                                              |
| <b>Symptom</b>               | Unable to establish a network tunnel on a 10.x.x.x network.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Problem Determination</b> | <p>On a 10.x.x.x network, you can establish the tunnel by configuring the tunnel to be on a 192.168.1.x address. For example, configure the tunnel device in the devmap as:</p> <pre>name awsosa 0009 --path=A0     --pathtype=OSD     --tunnel_intf=y     --tunnel_ip=192.168.1.1</pre> <p>Configure z/OS TCP IP profile as a tunnel IP of<br/>192.168.1.2 netmask 255.255.255.0</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Symptom</b>               | Cannot connect to z Systems Development and Test Environment z/OS using a remote 3270 terminal emulator                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Problem Determination</b> | <ul style="list-style-type: none"><li>• Ensure that you have network connectivity to the Linux adapter by pinging the Linux adapter IP address from the machine that is running your remote 3270 terminal emulator</li><li>• Ensure that you have a tunnel OSA defined with an internal 10.x.x.x address. Ensure that there is a matching route that sends all 10.x.x.x traffic through the tunnel OSA</li><li>• Ensure that the remote 3270 terminal emulator configuration is attempting to connect to the Linux IP address or host name, and port 3270 (or whatever port is defined on the 3270port statement)</li><li>• Ensure that your firewall configuration allows traffic to the 3270port. See Firewall considerations in “Configuring the base Linux Server” above for how to configure the firewall to allow the appropriate ports.</li></ul> |
| <b>Symptom</b>               | Cannot ping to a z/OS defined IP address on the z Systems Development and Test Environment instance                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

Table 13. z/OS networking issues (continued). Troubleshooting z/OS networking issues

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem Determination</b> | <ul style="list-style-type: none"> <li>• Ensure that you have a tunnel OSA defined with an internal 10.x.x.x address. Ensure that there is a matching route that sends all 10.x.x.x traffic through the tunnel OSA</li> <li>• You can only ping to the z/OS IP address when an IP address is assigned that is advertised to the network attached to the Linux ethernet adapter. The customizations in this book define one way to do this, described as scenario 4 in the zPDT Guide and Reference. If you are using that scenario, ensure that you have completed all of these steps, which are defined in “Setting up TCP/IP” on page 117: <ul style="list-style-type: none"> <li>– Defined a 2nd OSA, with an IP address in the same subnet as the IP address of the Linux ethernet adapter</li> <li>– Defined a default route through that OSA</li> <li>– Updated your TCPIP.DATA files to point to your domain name server and the host name for your z/OS</li> <li>– Updated your procedures to point to the new TCPIP.DATA files</li> <li>– Updated your IPNODES definitions to include the z/OS host name and IP address</li> <li>– Updated your resolver parms to point to the new IPNODES definitions</li> <li>– Updated your VTAM definitions to add the tunnel OSA and any other OSAs defined</li> </ul> </li> <li>• If you are using a configuration similar to scenario 5 in the zPDT Guide and Reference, the z/OS IP address is advertised through the network using Linux Network Address Translation (NAT) functions. <ul style="list-style-type: none"> <li>– Check your firewall configuration with the command <code>iptables -L -n</code> <ul style="list-style-type: none"> <li>- Ensure the correct chpid address, z/OS IP address, broadcast address, netmask for the broadcast address, and the tunnel IP address are all defined in the table</li> </ul> </li> <li>– Check your routing definitions in BEGINROUTES or OMROUTE. Ensure the firstthop address is the Linux tunnel address if all traffic is being routed through the Linux.</li> <li>– For more information, see Chapter 7 “LANs” in the zPDT Guide and Reference.</li> </ul> </li> </ul> |
| <b>Symptom</b>               | Cannot FTP or Telnet to a z/OS defined IP address on the z Systems Development and Test Environment instance                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>Problem Determination</b> | <ul style="list-style-type: none"> <li>• Verify that you can ping to the address from the remote location. If you cannot, follow the steps defined here.</li> <li>• If you can ping, check your firewall configuration with the command <code>iptables -L -n</code> <ul style="list-style-type: none"> <li>– Ensure that the appropriate ports are allowed; typically ports 21 for FTP and 23 for Telnet</li> </ul> </li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

## Performance

Table 14. Performance. Troubleshooting performance issues

|                |                                                                                                                                              |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Symptom</b> | High volume traffic through an emulated OSA-Express® ethernet adapter on z Systems Development and Test Environment suffers poor throughput. |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------|

Table 14. Performance (continued). Troubleshooting performance issues

|                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Problem Determination</b> | <p>If newer Linux kernels are installed, you might experience a drastic slowdown of OSA performance. This slowdown is immediately obvious and is due to Linux attempting to offload various functions into the adapter, which is not acceptable to the current awsOSA implementation. One or more of the following commands, intended to disable the Linux offloading of IP functions, might improve the situation:</p> <pre># ethtool -K eth0 rx off (disable RX checksumming offload) # ethtool -K eth0 tso off (disable TCP segmentation offload) # ethtool -K eth0 gso off (disable generic segmentation offload) # ethtool -K eth0 gro off (disable generic RX offload) # ethtool -K eth0 lro off (disable large RX offload) # ethtool -K eth0 rxvlan off (if you are using VLANs) # ethtool -k eth0 (display status of NIC) # ethtool -S eth0 (display statistics) # ethtool -K em1 rx off (newer style of NIC naming) # ethtool -K enp0s25 rx off (newer style of NIC naming)</pre> <p>You might need to experiment with these commands.</p> <p>These commands must be entered after each Linux boot. If a script is used to start z Systems Development and Test Environment, such as the runzpdtd script shipped with the product, that script can be modified to enter one or more of these commands each time z Systems Development and Test Environment is started.</p> <p>Effective combinations of these options differ with various Linux levels and with various NIC adapters. This problem was first noticed with Linux kernel level 2.6.36.2</p> <p>If frames larger than expected are used, an excessive number of frames might be dropped (causing a re-transmission). This may not be noticed unless careful measurements or comparisons are made. This problem might be resolved by including the sysctl parameter that is now recommended:</p> <pre>net.core.rmem_max=1048576</pre> <p>For more information on zPDT performance and performance problems, see paragraph 2.3.4 "Performance" and 7.5.9 "Performance problems" in the zPDT Guide and Reference.</p> |
| <b>Symptom</b>               | z Systems Development and Test Environment performs slowly. CPUs reach 100% utilization.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Problem Determination</b> | Verify that you are meeting the minimum requirements for both the hardware and the software. See "z Systems Development and Test Environment prerequisites" and paragraph 2.3.2 "zPDT instances" in the zPDT Guide and Reference.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

## Connecting with Rational License Key Servers

Activity that is associated with the use of Rational Tokens is logged for diagnostic purposes in the log directory \$HOME/z1090/logs in files that start with the name feutlicm. Messages from a Rational License Key Server and diagnostic information can be found in these logs. These logs are intended for use by IBM service but might provide useful information for quick diagnosis of problems when Rational Tokens cannot be obtained. In some cases when Rational Tokens cannot be obtained, the messages that are issued by the Rational License Key Server are also



written to the Linux console on which the `awsstart` command was entered. The `feutlcm` log can be viewed with the `less` command while z Systems Development and Test Environment is running.

Two environment variables can be used to help in troubleshooting connections to Rational License Key Servers and problems that occur when you are obtaining Rational Tokens.

Variable `RDTLG=TTY`, if set before you start z Systems Development and Test Environment, routes all Rational Token-related logging to the Linux terminal in addition to the log.

Variable `RDDEBUG=DEBUG`, if set before you start z Systems Development and Test Environment, adds more information to the logs. If `RDTLG=TTY` is also set, these additional messages are also written to the Linux terminal.

Do not set `RDTLG=TTY` in everyday use, because it sends frequent unsolicited messages to your Linux terminal. Setting `RDDEBUG=DEBUG` in regular use has no negative side effects, other than slightly larger logs.

Rational Tokens are checked out and checked back in so that they will become available automatically after 30 minutes unless z Systems Development and Test Environment renews them before that time. z Systems Development and Test Environment renews tokens approximately every half hour.

When z Systems Development and Test Environment ends, tokens are returned immediately. However, it can take up to 2 minutes for those tokens to become available for use again. If network connectivity is lost to the Rational License Key Server, or if anything prevents a normal return, the Rational Tokens become available within 30 minutes.

To limit unnecessary log file growth, logging of successful interactions with the Rational License Key Server is suspended after about 30 successful token renewal cycles. Logging resumes if any errors are encountered.

---

## Troubleshooting license manager and USB hardware device activation

Configuration and usage for z Systems Development and Test Environment is described in Chapter 6, “Configuring an instance of z Systems Development and Test Environment,” on page 103 and the zPDT Guide and Reference.



---

## Chapter 11. Reference

The reference material in this section of the IBM Knowledge Center provides supporting information for the instructions for using IBM z Systems Development and Test Environment.

---

### Setting up Rational Tokens

Rational Token licensing is an entitlement that you can purchase and use to run z Systems Development and Test Environment. With Rational Tokens, z Systems Development and Test Environment maintains a connection to a Rational License Key Server and starts and continues to run only when sufficient Rational Tokens are available.

Use of Rational Tokens does not replace the requirement for a license key for z Systems Development and Test Environment. Either a software-based license key file or a USB hardware device with a valid update file is still required. In either case, the license key file indicates that Rational Tokens are required.

After you purchase entitlement to Rational Token licensing, do these steps to use Rational Token licensing to run z Systems Development and Test Environment. These steps assume that you already have a USB hardware device.

### Installing, configuring, and administering a Rational License Key Server

Setting up the Rational License Key Server is beyond the scope of this document, but additional information can be found in the Rational License Key Server documentation that is provided in the Rational License Key Server media.

If you need to use configuration files to specify extra Rational licensing configurations, you must specify the variable `RDTCNF` to point to a directory that contains the Rational configuration files. For example:

```
export RDTCNF=/etc/yourconffiles/
```

### Returning any existing license key for the USB hardware device or license manager

If you want to allow Rational Token licensing for a USB hardware device or license manager you are already using, you must first return the license key that was previously created for that device. The Rational License Key Center does not generate new license keys unless any previously created license keys for that device are returned.

To return an existing license key for a license manager, see “Returning a software-based license key less than 31 days before expiration” on page 74.

To return the existing update file for a USB hardware device, log in to the Rational License Key Center, select your account and use the **Return Keys** link to return the old update file. For instructions for returning your existing update file, see “Getting the replacement file” on page 91.

## Obtaining and activating a new license key for Rational Token licensing

Links to information about how to obtain and install a new license key for Rational Token licensing for license managers and for USB hardware devices.

To obtain and install a license key file for a license manager, see “Activating a license manager” on page 71.

To obtain a license key file for a USB hardware device, use the same procedures that are documented in “Obtaining update files for Rational Tokens” on page 91 to obtain an update file for Rational Token Licensing. To activate the update file, use the steps that are described in “Activating the USB hardware device” on page 93.

**Important:** In a product license server configuration, do not mix USB hardware devices that require Rational Tokens with USB hardware devices that do not require Rational Tokens. This approach is not supported and can result in unpredictable behavior. For more information about Rational Tokens and activating the USB hardware device, see “Product enablement checklist” on page 61.

### Activating the USB hardware device

Applying the update file that you obtained from the Rational License Key Center to your USB hardware device activates the device. Each time that you apply an update file, it overwrites the previous activation on the USB hardware device. To properly activate, the update file must be generated with the same serial number as the USB hardware device that it applies to. To change the activation of a USB hardware device, you must obtain and apply a new update file that activates the total number of license entitlements you intend to use on that device. Changing the activation of a USB hardware device involves returning and regenerating licenses in the Rational License Key Center.

The process for applying update files to USB hardware devices recently changed. For instructions on applying update files to USB hardware devices for z Systems Development and Test Environment version 9.1 or later, see “Activating a USB hardware device” on page 94

For instructions on applying update files to USB hardware devices for Rational Development and Test Environment for z Systems version 9.0 or earlier, see “Activating a USB hardware device for version 9.0 or earlier” on page 95

**Important:** In a product license server configuration, do not mix USB hardware devices that require Rational Tokens with USB hardware devices that do not require Rational Tokens. This approach is not supported and can result in unpredictable behavior.

## Updating the device map to point to a Rational License Key Server

The Rational License Key Server that is used to distribute Rational Tokens must be available through TCP/IP. It is likely that your installation already has such a server configured. Installation materials and documentation for the server are available on separate e-images that are included with the z Systems Development and Test Environment offering.

When you are using Rational Tokens, an instance of z Systems Development and Test Environment must be able to locate the specific Rational License Key Server

you intend to use with that instance. The port and location of the server is specified either in the Linux environment variable that is named `RDTSERVER` or in the device map. If both are specified, the device map setting is used. To enable Rational Tokens in the device map, add the `rdtserver` statement to the `[system]` stanza. For example, to have z Systems Development and Test Environment retrieve Rational Tokens from port 27000 on a server at address `sampsrvr.yournetwork.com`, add the following line to your `[system]` stanza in the device map:

```
rdtserver 27000@sampsrvr.yournetwork.com
```

To have z Systems Development and Test Environment retrieve Rational Tokens from port 27000 on a server at address `sampsrvr.yournetwork.com`, you can export the `RDTSERVER` environment variable in your `.bashrc` script. For example, if you run z Systems Development and Test Environment with ID `ibmsys1`, you would edit `/home/ibmsys1/.bashrc`, and add the line:

```
export RDTSERVER=27000@sampsrvr.yournetwork.com
```

## Troubleshooting connections with Rational License Key Servers

Activity that is associated with the use of Rational Tokens is logged for diagnostic purposes in the log directory `$HOME/z1090/logs` in files that start with the name **feutlicm**. Messages from a Rational License Key Server and diagnostic information can be found in these logs. These logs are intended for use by IBM service but might provide useful information for quick diagnosis of problems when Rational Tokens cannot be obtained. In some cases when Rational Tokens cannot be obtained, the messages that are issued by the Rational License Key Server are also written to the Linux console on which the **awsstart** command was entered. The **feutlicm** log can be viewed with the **less** command while z Systems Development and Test Environment is running.

Two environment variables can be used to help in troubleshooting connections to Rational License Key Servers and problems that occur when you are obtaining Rational Tokens.

Variable `RDTLG=TTY`, if set before you start z Systems Development and Test Environment, routes all Rational Token-related logging to the Linux console in addition to the log.

Variable `RDDEBUG=DEBUG`, if set before you start z Systems Development and Test Environment, adds more information to the logs. If `RDTLG=TTY` is also set, these additional messages are also written to the Linux console.

Do not set `RDTLG=TTY` in everyday use because it sends frequent unsolicited messages to your Linux console. Setting `RDDEBUG=DEBUG` in regular use has no negative side effects, other than slightly larger logs.

Rational Tokens are checked out and checked back in so that they will become available automatically after 30 minutes unless z Systems Development and Test Environment renews them before that time. z Systems Development and Test Environment renews tokens approximately every half hour.

When z Systems Development and Test Environment ends, tokens are returned immediately. However, it can take up to 2 minutes for those tokens to become

available for use again. If network connectivity is lost to the Rational License Key Server, or if anything prevents a normal return, the Rational Tokens become available within 30 minutes.

To limit unnecessary log file growth, logging of successful interactions with the Rational License Key Server is suspended after about 30 successful token renewal cycles. Logging resumes if any errors are encountered.

---

## zPDT license servers

Alternative zPDT license and serial number servers that provide enterprise-wide management are available for IBM z Systems Development and Test Environment systems.

**Note:** For definitions of some of the terms used in these topics, see the Chapter 12, “z Systems Development and Test Environment Glossary,” on page 231.

A zPDT system must have a license supplied by a 1090 or 1091 token or by a *software license server*. The tokens identified as 1091 tokens are for z Systems Development and Test Environment customers. The material in this section applies to both 1090 and 1091 tokens, and to software-only (LDK) license users. For several reasons, simple local token usage is not always appropriate:

- Due to security concerns, some PCs no longer have usable USB ports. The physical distribution of tokens might present a problem.
- Rack-mounted blade PCs might not have normal, dedicated USB ports. A token in a work location can be easily carried away.

In virtual environments the dedicated use of a USB port might be a problem.

- If multiple tokens are used, or are changed, the CP serial numbers become unpredictable. The consistency of the z Systems serial numbers might be important for some software licenses (for z Systems software) and might be important for some z Systems operating systems.
- In some cases, especially related to cloud usage, a hardware token at any location is difficult to manage.

Recognizing these concerns, alternative zPDT license and serial number servers that provide enterprise-wide management are available for z Systems Development and Test Environment systems. Figure 5 on page 209 shows the available options: a simple local configuration, a remote LDK-SL license server, and a remote SHK license server.

**Important:** The SHK and LDK-SL terminology associated with these servers, as shown in this figure, is used throughout this section. SHK servers have physical 1090 (or 1091) tokens and LDK-SL servers do not have physical tokens. The terms *license manager* and *license server* are used interchangeably. zPDT refers to both the ISV zPDT product (1090) and the z Systems Development and Test Environment (1091) product except where distinctions are noted.

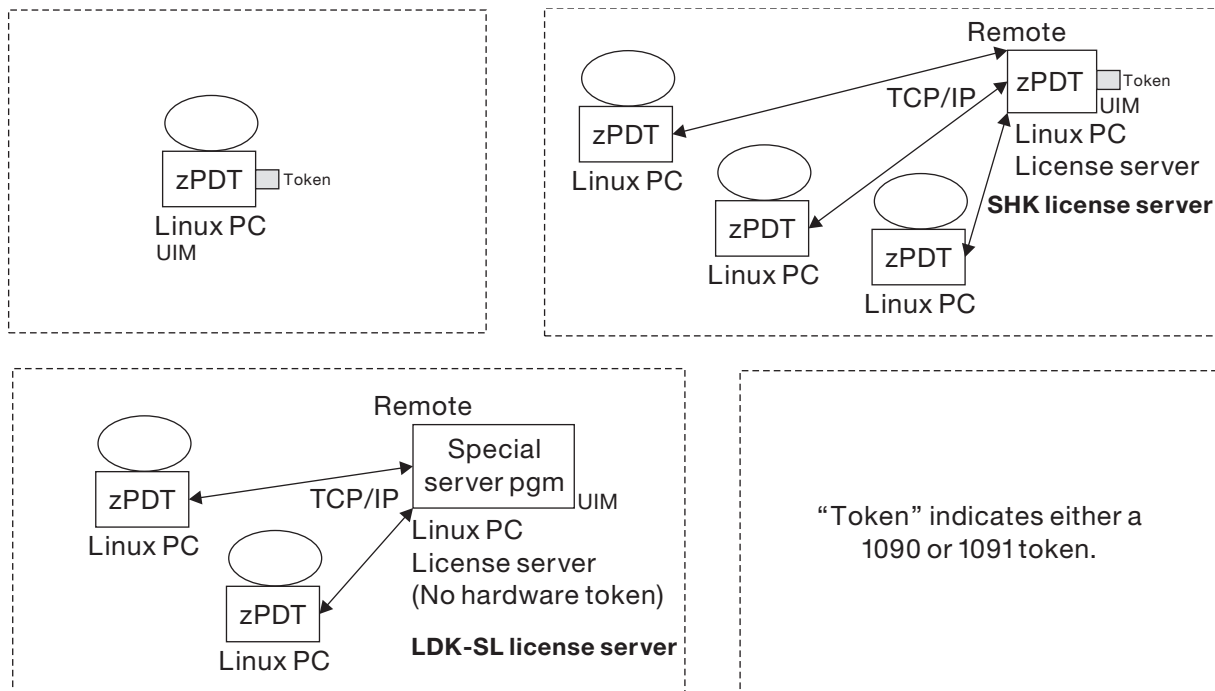


Figure 5. Options for obtaining zPDT licenses

In a simple configuration, a *local token* is installed in a USB port on the base machine running zPDT. In this case (one token installed in a local USB port), the token supplies both the zPDT license and the serial number used for the z Systems CPs, assuming that the local zPDT system has never been connected to a remote license manager or server, and has never used multiple local tokens. This configuration is used by the majority of zPDT users.

The SHK server uses a hardware token, while the LDK-SL server uses a software-only license with no physical token. An SHK server can be shifted to another physical PC by moving the token(s) and reinstalling zPDT software. An LDK-SL license server cannot be moved to a different PC. To move the server to a different PC, you must obtain new LDK-SL server software. Also, additional license charges may be associated with the use of LDK servers; consult your zPDT provider for more details.

**Restriction:** LDK-SL server function is intended primarily for systems accessed in the cloud.

Figure 5 also indicates UIM components. UIM means Unique Identification Manager; this is a function that provides a consistent z System serial number to zPDT. The UIM function can be used with remote UIM servers. In principle, these are separate servers from the license servers and might be on different Linux PCs. In practice, the remote UIM servers are almost always installed on the same Linux PC having the remote license server. These topics assume that a UIM server is installed concurrently with an LDK-SL or SHK license server. There is also a local UIM component with operational zPDT systems (clients) not indicated in the figure.

A *license server* is accessed (via TCP/IP) by a *client* PC running zPDT and the zPDT operational license is supplied this way. The licenses needed to decrypt z/OS IPL volumes are also provided by the server. The client machine does not have a token

and does not need a USB port. A client machine must have access to the license server as long as zPDT is operational on the client. Likewise, the client machine has access to a UIM server that supplies consistent serial numbers for the z Systems CPs.

All zPDT systems have remote client functionality but, by default, it is not configured for remote operation. If a token is installed zPDT operates normally (with a local token). If a remote client function is configured, then zPDT attempts to connect to remote servers to obtain a zPDT license and serial number.

The owner of the client machine must do some minor configuration work to enable clients to use remote license servers and UIM servers; the enabling this interface differs for SHK and LDK-SL servers. Before enabling client access to a remote server the server networking environment (IP address, domain name, firewall controls, appropriate tokens for the server) must be arranged.

The remote license and UIM servers are normally on a single remote system. However, the two servers could be on separate machines. A UIM server and/or an SHK server could be on the same machine as the client, but would still be considered remote servers in the context described here. All the following text assumes that the license server and the UIM server are on the same machine. An LDK-SL server cannot be present on the same PC running zPDT.

**Tip:** The LDK and SHK terminology represents different generations of license management functions from Safenet, with LDK being the newer technology. (The company is now owned by Gemalto, but these help topics continue to refer to the Safenet “token” products.) The LDK technology can use both “software license” (denoted by LDK -SL) or new hardware tokens (denoted by LDK-HL). At the time of writing, zPDT does not use the newer hardware tokens (LDK-HL).

## Concepts

z Systems CECs have unique serial numbers, allowing software to identify the machine and LPAR. Some operating systems verify that the “IPLed” machine has the same serial number as the machine that last used that copy of the operating system and may react differently if there is a mismatch. Some software products are licensed by machine serial number.

A simple zPDT system has a simple unique serial number design: the serial number of the zPDT token becomes the serial number of the z Systems created by zPDT. Figure 6 on page 211 illustrates the conceptual operation.



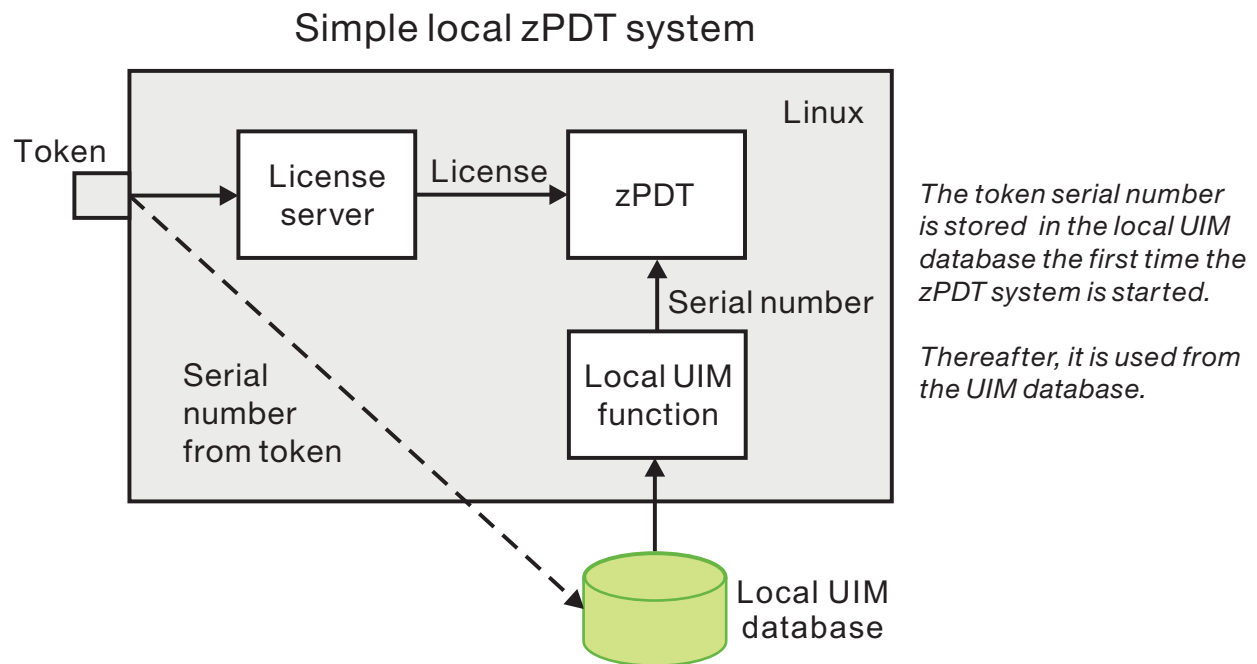


Figure 6. Simple local operation

When a remote license server is used (or if multiple local tokens are used) there needs to be a method of assigning unique serial numbers that do not change after they are assigned. Figure 7 on page 212 illustrates the general concept.

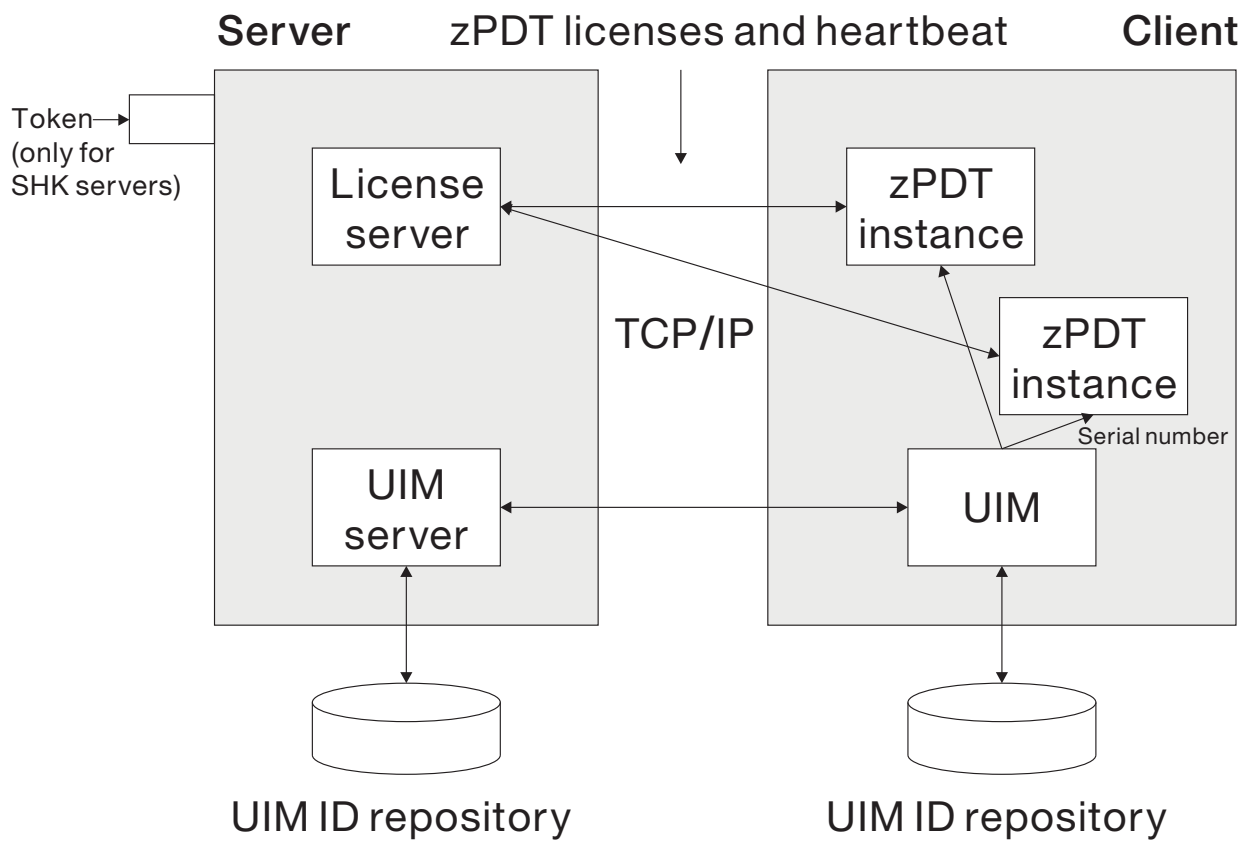


Figure 7. License and UIM servers

There are two modes of operation, *local* and *remote*. In the simple local mode both the license function (by a local token) and the UIM function run in the same machine as the client, as shown in Figure 6 on page 211, but are generally invisible to the user. An LDK-SL server cannot be used in local mode. In remote mode, the license server and UIM server program are in a remote machine which can serve licenses and serial numbers to a multitude of clients via TCP/IP.

Each Linux zPDT instance is assigned a unique serial number, either from a local token or by a UIM server. Every zPDT instance (running under a Linux user ID) has an LPAR ID assigned to it. An *instance* refers to multiple zPDT copies used on a base Linux system. The *LPAR ID* is not the same as the LPAR name. The *LPAR name* is the same as the Linux user ID that started the zPDT instance. zPDT instances have some of the characteristics of an LPAR, but full LPAR functionality is not provided by zPDT. The combination of serial number and LPAR ID becomes part of the CPUID. The CPUID is the information provided by the z Systems instruction Store CPU ID (STIDP).

Once assigned a serial number, the number is not changed even if the corresponding token (or software license) numbers are changed. The user must use the **uimreset** command to allow a serial number change. A user cannot assign an arbitrary serial number; the serial numbers are generated by UIM or taken from a token.

There are several notes relevant to Figure 7:

- The default port number for the SHK remote license server is 9450 and the default port number for the UIM server is one greater than the license server port number (and is 9451 by default) . The port number for the LDK-SL remote license server is 1947. The SHK server and UIM server port numbers are configurable; the LDK-SL port number is not configurable.
- After a zPDT instance is started (on a client) access to the UIM server is no longer needed.
- After a zPDT instance is started (on a client) the license access must be maintained for the life of the zPDT instance. If the access is dropped, the zPDT instance stops. (If the access is recovered, zPDT starts again.)
- The servers must be identified by resolvable domain names or by IP addresses. This is easy if they have direct, fixed IP address or domain names. It is not easy if DHCP-assigned addresses or NAT functions or VLAN networks are involved. Skilled network planning is required for any but the simplest environments.
- As a general statement, any PC system that can access the IP subnet of a license server can obtain a zPDT license there. Network security and license server security configurations may be important. This aspect is further described in "Security" on page 223.

Firewalls between the servers and clients must allow the required IP and port access.

- A client machine may be changed to a stand-alone machine (with token) by changing a configuration file, and vice versa.
- In normal operation, a client machine always has the same z Systems serial number. This number, once assigned via a local or remote function, might not be related to any physical token number.

Any license or UIM configuration changes should be made when zPDT is not operational.

The rules for obtaining a zPDT license are straight-forward. Either a local token is used or a remote license server. The indicated source must have an appropriate token or software license pool to provide a zPDT license.

The rules for zPDT serial numbers are more complex. The goal is to always have the same unique serial number for a given zPDT instance. The following general rules are used to determine the z Systems serial number for a zPDT instance. The term *UIM serial number* means a serial number generated and assigned by a UIM server. The term *random serial number* is also used for serial numbers created by a UIM server. After a random serial number is generated and assigned to a client, it is used consistently. The term *random* applies only to the initial generation of a serial number by a UIM server and indicates that the serial is not related to a specific token serial number. You cannot create the random number yourself.

If a single local token is used (and no previous serial has been assigned):

- The first zPDT startup will take the z Systems serial number from the token. This serial number is then written in the local UIM database.
- Subsequent zPDT startups must use the same token.
- If a different token is used, the `uimreset -l` command must be issued first (before zPDT is started). This erases the existing serial number in the local UIM database, allowing a new token (with a different serial number) to be used.

- Or, the RANDOM parameter may be specified in the XML configuration file. This allows any token to be used with an existing serial number in the local UIM database. (The **clientconfig** command is used to change this parameter.)

If a single local token is used and if a UIM serial number is present in the local UIM database (due to a previous connection to a UIM server) then the UIM serial number is used and the local token serial number is ignored. (The local token still supplies the zPDT license unless a remote license server is configured.)

If multiple local tokens are present (and no previous serial number exists in the local UIM database) the serial number of one of the tokens is accepted and stored in the client UIM database. This stored serial number is used subsequently, without further reference to the serial numbers of the tokens. In this case the RANDOM option must be specified by using the **clientconfig** command.

If the client is configured for a remote UIM server the following information applies:

- If no serial number is known for the client system, the UIM server generates a random serial number, that is, one that is not related to a token serial number, and sends it to the client UIM database.
- If the local client UIM database already contains a valid serial number that does not conflict with another client's serial number (as stored in the UIM server database) that serial number is used.

If the client serial number (in the client UIM database) conflicts with a serial number in the UIM server database, the client operation fails. In this case, the client system may use the `uimreset -l` command to remove the serial number in the local UIM database.

If the client changes to a local configuration after previously using a remote configuration the previously assigned serial number (from the remote server and stored in the local UIM database) is used. The local token serial number is ignored.

## Client installation and configuration

All client functions (for both licenses and UIM functions) are included and installed by the zPDT installation package. Whether the remote functions are used depends on configuration file options. For more information about the basic zPDT client installation process see Chapter 5 of the zPDT Guide and Reference.

### SHK client configuration

After a normal zPDT installation, SHK client operation is configured by settings in file `/usr/z1090/bin/sntlconfig.xml`. This file is referred to as the XML file.

This example shows the general syntax of this file. The actual XML file might have different spacing and more comments than shown here.

```
<SentinelConfiguration>
 <SentinelKeys>
 <ContactServer>localhost</ContactServer>
 <ServerPort>9540</ServerPort> <Protocol>SP_TCP_PROTOCOL</Protocol>
 </SentinelKeys>
 <UniqueIdentificationManager>
 <UIMContactServer></UIMContactServer>
 <UIMServerPort></UIMServerPort>
 <UIMProtocol></UIMProtocol>
 <UIMLocalSerialMethod></UIMLocalSerialMethod>
 </UniqueIdentificationManager>
</SentinelConfiguration>
```

Do not modify this file directly. Direct editing of an XML file is error prone and can be difficult to debug. Instead, use the **clientconfig** command to make changes. You must operate as root to modify the file or to use the **clientconfig** command. This command produces a display similar to this example:

```
Gen2 ContactServer..... _____ (default is blank)
Gen2 BackupServer..... _____ (default is blank)
Gen1 ContactServer..... _____ (default is blank)
Gen1 BackupServer..... _____ (default is blank)
UIM ContactServer..... _____ (default is blank)
UIM Local Serial Random.. _ (y or blank)
Factory Reset..... _ (Enter "y" to reset file)
```

To change parameters values, overwrite them.

- Leave Gen2 Contact Server empty. This field is used for software-based license manager.
- Leave Gen2 BackupServer empty. This field is used for software-based license manager.
- Set Gen1 Contact Server to point to IP address or hostname of product license server.
- Set Gen1 BackupServer if you want to set up the failover product license server. Set this field to point to the IP address or hostname of failover product license server.
- Set UIM ContactServer if you want to use different UIM server than product license server. Otherwise, leave this field empty. The field is empty.
- Set UIM Local Serial Random to "y", or leave this field empty. This field is optional. The UIM Local Serial Random specification is needed if multiple tokens are used on a local client or if different tokens are used at different times.
- If the Factory Reset option is set to "y", all other parameters are ignored, and the XML file is restored to the original values shipped with zPDT.
- Press Enter twice to save these values.

Changes to the configuration file are not dynamic. They take effect only when zPDT is started.

By default, the clientconfig command operates on the sntlconfig.xml file located in directory /usr/z1090/bin. The file name sntlconfig.xml is constant, but you may specify an alternate directory location as an operand:

```
clientconfig /my/special/directory/
```

## LDK client configuration

After the normal zPDT package is installed the LDK client can be activated. The LDK client (and server) requires a 32-bit version of the Linux glibc library and the client installation process automatically accesses several Internet sites to obtain the latest version of this library.

Be certain you have a working Internet connection before starting this process. Your base Linux might already have glibc-32bit installed. If glibc-32bit is not already installed on your base Linux and if you cannot connect to the Internet (perhaps due to firewalls) then you must obtain and install glibc-32bit in some other way. The LDK functions (client and server) will not operate without this rpm. After checking your Internet connectivity, and working as root, issue this command:

```
/usr/z1090/bin/gen2_init
```

The resulting display depends on your Linux distribution, but might look like the following:

```
[root@zdt-dev3 ~]# /usr/z1090/bin/gen2_init

Script for installing 32-bit compatibility packages for 64-bit Linux.
Copyright (C) 2013, SafeNet, Inc. All rights reserved.

Detected glibc 32bit support already installed

Installing LDK client side license daemon
Preparing... ##### [100%]
Updating / installing...
 1:aksusbd-7.40-1 ##### [100%]
Starting aksusbd (via systemctl): [OK]

..Done.
```

This setup is done only once. Thereafter the LDK client is started automatically when the client Linux system is booted.

Issue `./query_license` in the path `/opt/IBM/LDK` to see if the client is correctly configured and able to get license during IPL△

## Client UIM configuration

The client UIM information is held in `/etc/z1090/uim/uimclient.db`. In unusual error situations you might be advised to delete this file. Deleting this file causes the UIM function to obtain or create a new serial number (working with your local token or with a remote UIM server) when zPDT is next started.

To configure the client UIM:

- For a local token client, the UIM function is normally transparent; no action is needed.
- For an SHK license server, the license server configuration (with the **clientconfig** command) also configures access to the UIM server. By default, the UIM server is assumed to be at the same IP address as the SHK or LDK server..
- To set up the different UIM server than the SHK/LDK server, set the UIM Contact Server field.

For more information about license server configuration, see “SHK client configuration” on page 214 and “LDK client configuration” on page 215

## Server installation and configuration

Both the SHK license server and UIM server are included in the standard zPDT package. The license server runs as a daemon and is automatically started when Linux is booted. (This is true even for local token use.) Both servers are TCP/IP users and your network configuration (including firewalls) must allow connectivity to the servers. The default port numbers are 9450 (license server) and 9451 (UIM server).

The LDK-SL license server and UIM server are not part of the standard zPDT package. A separate package with these two components is available as a separate offering. The LDK-SL offering is only for z Systems Development and Test Environment customers.

### UIM server

The UIM server is automatically installed when installing either the SHK or the LDK-SL license server packages provided for zPDT.

Once installed, the remote UIM server must initially be started manually; thereafter it is automatically managed by cron. It must not run as root. It runs under a normal Linux userid and places its database in the home directory of that userid. It also places small log files in the home directory. For this reason, the same Linux userid (not root) should always be used to run the UIM server.

Two commands are associated with running the UIM server:

```
$ uimserverstart
 Start the UIM server.
$ uimserverstop
 Stop the UIM server.
```

The **uimserverstart** command, in addition to starting the server, places entries in the Linux cron files such that the UIM server is restarted automatically (after 10 minutes) if it fails. It is also started automatically during a Linux reboot. The **uimserverstop** command stops the server and removes these cron entries.

No other configuration is needed for the UIM server. You must not edit the UIM database file that is created in a subdirectory of the home directory of the userid running the UIM server.

### SHK license server

The SHK license server is part of the standard zPDT package and is installed as if you were installing a zPDT client. It is activated by the actions of the two token “driver” components that are part of zPDT installation.

One or more 1090 or 1091 tokens must be installed in the license server machine before it can be used. The license server configuration file is located in:  
`/opt/safenet-sentinel/common_files/sentinel_key_server/sntlconfigsrvr.xml`

This file typically does not require any additional configuration. If you want to change the license server port number, you can edit and change this file. You would then need to restart the server by using these commands:

```
cd /opt/safenet_sentinel/common_files/sentinel_keys_server
./loadserv restart
```

Several security functions may be specified in the `sntlconfigsrve.xml` file.

### LDK-SL License server

Several steps are involved in preparing an LDK- SL license server. The license server (and the associated UIM server) are supplied in a file with a name similar to that shown in the following command.

Place this file in a convenient directory and, working as root, execute this file:

```
./zPDT_LS-1.6.49.20L-x86_64
```

This file must be executable. This might require a **chmod u+x** operation. Also, the exact file name may change slightly to match newer levels of zPDT.

The installation process causes an Internet search for the latest version of the 32-bit glibc library, as described in “LDK client configuration” on page 215. Both the LDK-SL license server and a UIM server are installed. The LDK server is installed in `/opt/IBM` instead of the traditional `/user/z1090/bin` that was used for other zPDT modules. The next step is to obtain licenses that can be “served” by the license server. Working as root, issue this command:

```

/opt/IBM/LDK/request_license
```

This command creates a file named *hostname\_XXXXXX.zip* in root's home directory, where *hostname* is your Linux system's name and *XXXXXX* is a timestamp. This file contains a fingerprint of the license server. You must send this file to the appropriate IBM licensing facility (as identified by your zPDT contract). In return you will receive a "v2c" file containing the number and type of licenses your server can supply to clients. Receive this file into a convenient directory and install it by using this command:

```
/opt/IBM/LDK/update_license hostname_XXXXXX_update.zip
```

Then restart the license server daemon by using one of these commands:

```
#systemctl restart aksusbd.service
```

Used with newer Linux distributions.

```
service aksusbd restart
```

Used with older Linux distributions.

This completes the LDK-SL license server installation. You might need to start the UIM server on your server system.

The v2c file that conveys licenses to the server also contains ADCD decryption licenses that become available to the client systems.

## Notes

Learn about UIM function commands, how to manage firewalls involved with remote servers, how changing the Linux disk (HDD) might change the identifier that is part of the identification used by UIM, cloning a zPDT system, and removing zPDT functions.

### UIM function commands

Several commands are associated with the UIM function.

```
uimreset [-l] [-r]
```

This command clears the serial number in the local UIM database [-l] or in both the remote and local UIM database [-r]. This command must be run by root.

If you decide to "start all over" and reinstall your zPDT system, there might be a problem with serial numbers. If you use the same single local token that was used previously, zPDT will obtain the same serial number from it. If you use a remote license server and used the **uimreset - r** command to delete any previous references or if you have multiple local tokens your new zPDT installation might not have the same serial number as the previous setup. If you do not care about z System serial numbers then this is not a problem. If you do care about z System serial numbers because of software contracts or software sensitivity this can be a problem. The only certain way to obtain the same z System serial number is to use the same single local token.

### \$uimserverstart and \$uimserverstop

These commands start and stop a UIM server. A UIM server can run under any user ID (except root) on the server machine, but it should always be the same user ID. These commands are not normally used in a purely local client environment. These commands cannot be run by root.



### **\$ uimcheck**

The **uimcheck** command should be used if there is any question about the state of the serial number on a zPDT machine. Any user may issue this command.

### **SecureUpdateUtility, Z1090\_token\_update, and Z1091\_token\_update**

Do not run **SecureUpdateUtility** or **Z1090\_token\_update** from a client zPDT machine when using a remote license server. This utility cannot affect tokens or licenses in the remote license server, but will attempt to access a token in the local PC. You may run **SecureUpdateUtility** or **Z1090\_token\_update** in the SHK license server, to update the tokens in the server. Normal guidelines for **SecureUpdateUtility** or **Z1090\_token\_update** and **Z1091\_token\_update** apply. For example, only one token should be connected to the PC when you use these commands.

The administrator of a license server is responsible for ensuring the license keys do not expire while in use. The situation in which multiple tokens are installed (in an SHK license server) and the licenses in one token expire can be complex. Clients see license expiration warning messages starting a month before the license expires. However, if multiple tokens are present it is not predictable which token will furnish the license (or licenses) for a zPDT startup.

### **\$ token**

The license expiration date displayed by the **token** command (in a client machine) may not reflect the effective expiration date of all the active tokens in a license server. The **token** command (when zPDT is running) produces additional information:

```
CPU 0, zPDTA (1090) available and working. Serial 6186(0x182A)
Lic=88570(0x159FA) EXP=4/15/2017
```

In this example, the zPDT license was obtained from token 0x159FA (decimal 88570) and the CP serial number used by zPDT is 0x182A. There is no indication of whether a license server and UIM server are being used. Because the serial number and license number are different, we know that at some point the serial number was obtained from a license server. However, it is possible that the token is in the local client but that the serial number previously obtained from a UIM server is being used. This fulfills the goal of using a consistent serial number once it is assigned.

## **Firewalls**

You or your network administrators must manage any firewalls involved with remote servers. If you are initially installing in a test environment, disable all firewalls until you are satisfied with your zPDT license operations. Disabling firewalls helps distinguish network-related problems from license manager issues. If you operate through firewalls you must ensure that the relevant port numbers can pass through the firewalls.

There are many management techniques for firewalls, depending on what product is being used. Many Linux systems respond to **iptables** commands:

```
iptables -I INPUT -p tcp --dport 1947 -j ACCEPT
iptables -I INPUT -p tcp --dport 9450 -j ACCEPT
iptables -I INPUT -p tcp --dport 9451 -j ACCEPT
```

## **Disk and Linux changes**

Changing the Linux disk (HDD) might change the identifier that is part of the identification used by UIM. After changing the hard disk, you might need to use

the **uimreset -l** command to reset the local serial number or the **uimreset -r** command to reset the remote serial number.

Upgrading to a new Linux kernel might change the identification used by UIM. You might need to reset the local serial number or the remote serial number. If this does not solve the problem, delete the UIM database at `/usr/zpdt/uim`.

## Cloning zPDT

If you clone a zPDT system, you must delete the files in `/usr/z1090/uim` on the new system. This is because the UUID of the new system differs from that of the old system. zPDT will build new uim files when the new system is started.

## Removing functions

Use these commands to remove zPDT functions.

All SHK server functions (and associated UIM) can be removed by simply removing zPDT on that server. You can use either of these methods. In the first command, notice that the **--removeall** parameter is preceded by two dashes.

```
z1090-1-6-49.17.x86_64 --removeall
rpm -e z1090
rpm -e z1091
```

The LDK client function can be removed with a command like the following (where the exact file should match whatever name was used to install the LDK client function). Notice that the **--remove** parameter is preceded by two dashes.

```
/usr/z1090/bin/LDKc_setup.sh --remove
```

An LDK server is a normal rpm package that can be removed with this command, which also automatically removes the UIM server that was associated with the LDK server.

```
rpm -e aksusbd
```

## Scenarios

Learn about common usage scenarios.

### License search order

zPDT attempts to obtain a license from an LDK server if one is configured, then attempts to obtain a license from an SHK server if one is configured, and lastly attempts to obtain a license from a local token. There is a considerable timeout involved in trying to access the two servers and depending on this automatic search order is not reasonable for normal operation. The **--localtoken** option of the **awsstart** command simply short circuits any attempts to use remote license servers.

### Switch from local to remote server

Learn how to switch from a local to a remote license server.

In this scenario, two zPDT systems, A and B, each use a different PC for zPDT. System A has a zPDT token with serial number 12345.

1. The system A owner installs token 12345 in the PC and starts zPDT. When this is done, serial 12345 is recorded in the local system A UIM database. This scenario assumes no prior conflicting information was in the local UIM database. System A can be used in this configuration indefinitely until the token license expires with no reference to remote license or UIM servers.

2. The token is taken from system A for some reason, and the system A owner now wants to use remote license and UIM servers. With zPDT not running and working as root, the owner configures a client as described in “SHK client configuration” on page 214 or “LDK client configuration” on page 215.
3. The remote UIM server with the SHK or LDK-SL server, whichever one is being used sees that system A has serial number 12345 recorded in its local UIM database. The server checks whether this serial number is assigned to any other system. If there are no conflicts, the server records serial 12345 in the server database as belonging to system A.
4. Separately, the remote license manager serves a zPDT license based on a token present in the license server machine if it is using SHK, but the serial number of that token is not relevant.

So far, system A retains a consistent serial number, 12345, when switching from a local token to remote token or UIM servers. It retains this serial number every time this zPDT instance is used. If Multiple zPDT instances run on the same machine, they must run under different Linux user IDs. The serial number for each of the instances uses the LPAR portion of the serial number to differentiate the instances.

5. If token 12345 is transferred to the owner of system B, and the owner installs and uses it locally with no connection to the remote license or UIM servers, then both A and B have the same zPDT serial number. There is no way to avoid this.
6. If the system B owner then connects to the license or UIM servers, the UIM server sees serial 12345 in B’s local UIM database and terminates the zPDT instance because 12345 has already been assigned to system A.

This scenario is problematic because both A and B are attempting to use the same serial number, 12345, but the UIM server has it assigned to A. There are two ways to resolve this conflict:

- The system B owner can issue **uimreset -l** to clear the serial number in the local UIM database. The owner can then connect to the remote servers and receive a new random serial number.
- The system A owner can issue **uimreset -r** to clear the system A serial number from both the local and remote UIM databases. The next time system A zPDT starts, it requests a new random serial number from the server. System B can then use serial number 12345, which is stored in its local UIM database.

## Temporarily switch from server to local

Learn how to temporarily switch a client from a remote license and UIM server to local license.

In this scenario a notebook zPDT system is normally used with remote license and UIM servers. The owner wants to take the system home overnight, but the servers cannot be accessed from home.

If a token is available, you can start zPDT with the local option:

```
$ awsstart devmap_name --localtoken
```

In this case there is no need to use the **clientconfig** command to change the configuration file. The **--localtoken** option overrides the configuration file. The user must, of course, have a token to supply a license. In this case the serial number stored in the local UIM database is used and the serial number of the temporary token is ignored.

## Switch from remote server to local

Learn how to switch from a remote license and UIM server to a local license.

A system owner has been using a remote SHK license server and UIM server. To change to a local token, the owner used the **clientconfig** command to change the `LicenseContactServer` value to `localhost`. This command has the following effects:

- It effectively removes the `UIMContactServer` stanza from the XML file. The absence of this stanza indicates that no UIM server is to be used.
- In this case, `zPDT` looks in the local UIM database for a serial number. If one is present, it is used. If the local UIM database does not exist, or if the **uimreset -l** command was used, the serial number of the local token is placed in the local UIM database and then used by `zPDT`.

## Using zPDT on the license and UIM server

Learn how to run `zPDT` on the same machine that is running the SHK license server and UIM servers.

To run `zPDT` on the same machine that is running the SHK license server and UIM servers, use the **clientconfig** command to specify `LicenseContactServer` as `localhost` and `UIMContactServer` as `localhost`. This command has these effects:

- The presence of the `UIMContactServer` stanza means that a UIM server must be available on the indicated system, which is `localhost` in this example. Before starting `zPDT` on this system the user must issue a **uimserverstart** command.
- Give some thought to the Linux user ID that issues the **uimserverstart** command. The same user ID must always be used for this command because the UIM server database is created in the home directory of this Linux user ID.
- No special setup is needed for the license server. Any `zPDT` system, meaning the SafeNet server that is installed with `zPDT`, can act as a license server.
- Combined operation, as server and client, is not possible with an LDK server.

## Switching tokens

Learn how to switch tokens in a local UIM database.

In this scenario, token 12345 is used with a newly installed `zPDT` system. When `zPDT` is first started, this serial number is written in the local UIM database. If a different token is used on a subsequent startup, the `zPDT` startup fails. Use the **uimreset -l** command to remove serial 12345 from the UIM database. After the original serial number is removed, a new token can be used.

If the serial number in the local UIM database was assigned by a UIM server, or if the **RANDOM** parameter was used with the **clientconfig** command, then any local tokens can be used; the operational serial number is taken from the local UIM database.

The important point is that `zPDT` recognizes the difference between a UIM server-assigned serial number, which can be used with any token, and a locally installed serial number, which is taken from a local token. A locally installed serial number must match the token being used, unless the **RANDOM** option is set.

## Change from single token to multiple tokens

Learn how to switch from a single token to one of several other tokens.

You can switch from a single token one of several tokens. This procedure assumes that you are not using a remote license server.

1. From a user ID with root authority, enter a **uimreset -l** command.
2. Use the **clientconfig** command to set the UIM Local Serial Random value to Y.
3. Select the token containing the serial number you want assigned to the zPDT system. Start zPDT using this token.

Now you can start zPDT with any token. The serial number you selected in step 3 is used, regardless of which token you are currently using.

## Display serial number assignments

To display the zPDT serial number assignments, open a browser to the remote UIM server (<http://uimserveraddress:9451>). Port 9451 is the default UIM port.

The browser displays information similar to this example:

Serial	Host	UUID	Year	Day
2099	hostname.domain.com	56D96D01-493E-11CB-AD29-B8F42F7F8461	2016	009

## Security

If the license managers are used only from a single subnet, or a well-designed VPN, then security is not a major issue. If the license servers are accessed from the general Internet then security can be a significant issue. For example, your license server could provide zPDT licenses to someone completely unassociated with your enterprise.

### SHK server:

The SafeNet SHK license server can have three lists of IP addresses, domain names, or ranges of IP addresses.

- The Authorized User List determines which systems can use a web interface to manage the SafeNet license server. The default list contains only one address: 127.0.0.1, which is the local host and is always allowed whether specified or not.
- The Allowed Site Address list determines which clients can obtain zPDT licenses from the server. If the list is empty (the default) then any client can obtain a license from the server.
- The Blocked Site Address list specifies client addresses that cannot obtain a license from this server. If the list is empty (the default) then no client addresses are blocked.

Each list is limited to 32 entries. These lists are in the `sntlconfigsrv.xml` file in `/opt/safenet_sentinel/common_files/sentinel_keys_server/` and can be edited there. They can also be managed by opening a browser on port 7002 on the machine running the SafeNet license server:

<http://localhost:7002>

**Restriction:** The browser function provided by SafeNet appears to depend on specific Java levels. It might not work with the default Java level on current Linux systems.

If a different machine is used to access the server web interface, then the IP address of that machine must be listed in the Authorized User List. Use the

browser method, if possible, because directly editing this XML file is prone to introducing syntax errors that might cause the license server to fail. List entries might take any of these forms:

**127.0.0.1**

A simple IP address.

**my.local.domain.com**

A domain name.

**10.1.1.2-10.3.255.254**

A range of domain addresses.

If you are using the browser interface, be certain to click the update button on the web page after entering updates to the lists. You must then restart the SafeNet server:

```
cd /opt/safenet_sentinel/common_files/sentinel_keys_server
./loadserv restart
```

These lists provide one way to secure use of a zPDT license server. Other methods, such as restricted router interfaces or nonroutable IP addresses, might be more appropriate.

**LDK-SL server:**

You can use the browser interface to control access to the server.

Open a browser on URL address localhost:1947. In the Configuration page, under Access from Remote Clients, you can enter Access Restrictions. These can consist of mixtures of IP addresses in either numeric or domain name form, and user IDs. The browser pages contain useful help information for this functions. The security list is evaluated in the order of the statements. This example denies licenses to all requesters who are not on the 9.12.45.\* subnet.

```
allow=9.12.45.*
deny=all
```

**Firewalls:**

Working with the zPDT default port numbers, a firewall on a license and UIM server must allow connections to ports 9450 and 9451. One solution is to simply disable the firewall on the license server. Another solution is to enable the firewall and open the required ports.

To enable the firewall and open the required ports, issue these commands. These commands must be entered from a root user ID after the server Linux system is booted.

**Important:** Network management skills are needed to properly implement the server functions.

```
iptables -I INPUT -p tcp --dport 9450 -j ACCEPT
iptables -I INPUT -p tcp --dport 9451 -j ACCEPT
iptables -I INPUT -p tcp --dport 1947 -j ACCEPT
```

**Resetting UIM**

You can usually remove the local UIM serial numbers with the **uimreset -l** command. You can remove both the local UIM serial numbers and corresponding entries in the UIM server database with the **uimreset -r** command.

If the local UIM database is corrupted, the **uimreset** command might fail. You can delete the files in the `/usr/z1090/uim` directory. However, the previous UIM serial for the client is still provided by a UIM server if the client XML file is configured for connection to the server. In this case, you can use the **uimreset -r** command to remove the relevant entry from the UIM server database.

The UIM server can be reinitialized by removing everything in the `UIMserver` subdirectory in the home directory of the Linux user ID that runs the UIM server. This action should not be done in normal operational environments. If the `UIMserver` directory is cleared, some of the entries will be restored by future client connections in which the client still has previous UIM local data.

The client configuration file can be restored to its original state, which does not reference any remote servers, by using the Factory Reset option with the **clientconfig** command.

## Restarting Safe Net modules

Two SafeNet functions are involved with zPDT: the SHK or LDK-SL license servers and a daemon, or token driver, that communicates with tokens in USB ports. After zPDT is installed, both these functions are started automatically when Linux is started. Changing the license server files requires restarting the license server. It should not be necessary to restart the token driver except in unusual situations.

To restart the USB token daemon, enter these commands from root:

```
$ su
#cd /opt/safenet_sentinel/common_files/sentinel_usb_daemon
#./load_daemon.sh restart
```

To restart the SHK server, enter these commands:

```
cd /opt/safenet_sentinel/common_files/sentinel_keys_server
./loadserv restart
```

To restart the LDK-SL server, enter this command on newer Linux distributions:

```
systemctl restart aksusbd.service
```

Enter this command on older Linux distributions:

```
service aksusbd restart
```

## Renewing licenses

zPDT licenses in a token or in a software license server are usually valid for a year and must be renewed after that time. The procedure for renewing token licenses varies with different categories of zPDT users.

Users with 1090 tokens (typically ISVs or IBM internal users) normally create a request file by using the **Z1090\_token\_update -r** command and send this file to their zPDT provider. The provider, in turn, returns an update file that is installed with the **Z1090\_token\_update -u** command. This example shows one sequence of commands for requesting and installing the update file.

To request an update file, be sure that only a single token is connected to the computer. zPDT must be stopped and you must be logged in as root and in the `/usr/z1090/bin` directory.

```
$ awsstop
$ su
cd /usr/z1090/bin
Z1090_token_update -r mytoken.req
```

In this sequence of commands, the file name `mytoken.req` is an arbitrary name that you provide for the request. The `mytoken.req` file is sent to the zPDT provider. The provider returns the file `mytoken.zip`.

To process the returned file, you must be logged in as root in the `/usr/z1090/bin` directory. zPDT must be stopped .

```
Z1090_token_update -u mytoken.zip
exit
```

Older versions of zPDT use the **SecureUpdateUtility** command instead of **Z1090\_token\_update** and work with `.upwreturned` files instead of `.zip` files.

The returned `.zip` file also installs the token licenses needed to decrypt z/OS IPL volumes. After installing the new file, the token must be removed from the computer for about 15 seconds. Removing the token forces the token drivers to reread the token when it is reinstalled.

For IBM internal users, the provider is Resource Link®.

Users with 1091 tokens, who are typically z Systems Development and Test Environment users, might have a somewhat different process that does not require a request file. This process sends the customer a customized `.zip` file, which is installed by using this sequence of commands. Be sure that only a single token is connected to the computer. zPDT must be stopped and you must be logged in as root and in the `/usr/z1090/bin` directory. In this sequence of commands, `xxxxx.zip` is a file name assigned by the zPDT provider.

```
$ awsstop
$ su
cd /usr/z1090/bin
Z1091_token_update -u xxxxx.zip
exit
```

After you install the `.zip` file, the token must be removed for about 15 seconds. Software licenses (for LDK-SL servers) are installed by installing a new `v2c` file as described in “LDK-SL server” on page 224.

Licenses must be renewed on the computer that runs the server. You cannot renew or update the token licenses remotely. The **Z1090\_token\_update** command or the older **SecureUpdateUtility** command work only with a single token installed in a local USB port.

## Searching servers

More than one SHK or LDK-SL server can be specified for a client. Additional servers are simply listed by domain names or IP addresses in the respective client setup for the two license servers. The servers are searched for an appropriate license in the order listed. There is no coordination among the servers; each must have available licenses in the form of additional tokens for SHK servers or software entitlements for LDK-SL servers in order to serve them to clients. This means that the customer installation has purchased additional licenses or has split the available licenses among multiple servers in some way.

A zPDT client searches all available license sources until it finds the licenses it requires. If any LDK-SL servers are defined for the client, they are searched first, followed by SHK servers, followed by locally installed USB tokens. If remote



license servers are defined for a client but cannot be accessed by a TCP/IP connection, there will be delays while the access attempts timeout before another license server is tried.

If multiple license servers are routinely used it is possible, but unlikely, that a duplicate UIM serial number could be assigned to zPDT instances. If you use multiple license servers, reserve them for failover situations and not for routine use.

An LDK-SL license server cannot be shifted to another computer. Moving an LDK-SL license server function to a different computer involves multiple interactions with your zPDT license provider to ensure that the license entitlement information is removed from the old server and that a new license entitlement v2c file is created for the new server.

## Allocating multiple licenses

Learn how the zPDT system allocates multiple licenses to clients.

This example uses a remote SHK or LDK-SL license server with five zPDT licenses to illustrate how it allocates licenses to clients. A single client could request all five licenses by coding processors 5 in the devmap. Or five different clients could each request a single license. Or there could be a combination of clients that consume the five available licenses. When a client zPDT ends (with the **awsstop** command) the licenses used by that client are available to other clients. At any given instant no more than five zPDT client licenses, representing five CPs, can be allocated to clients.

Over time, many client zPDT systems might connect to this remote license server provided that not more than five licenses are allocated at any one time. Each of the many clients has a unique serial number provided by the remote UIM server. In this case, where five licenses are available, ten serial numbers can be associated with these five licenses. This distinction between numbers of licenses and numbers of serial numbers might be important for some ISV software license situations.

A single zPDT instance cannot have more than eight CPs, each requiring a zPDT license. IBM contract conditions might have a smaller limit. Assuming that the maximum of eight could be used, the devmap for an instance could request eight licenses from the remote server. In our example, only five licenses are available and the client would receive all five licenses (if no one else is using any licenses). Perhaps the intention of the customer is to share his five licenses among several development systems. There is no technical way to prevent a single user (that is, a single development system) from using all the licenses (up to eight, if that many are available). Management control is needed to ensure “fair” sharing of zPDT licenses in situations where a limited number of licenses are serving multiple remote clients.

## Switch from Product License Server to License Manager

Learn how to migrate from a Product License Server (hardware-based license) to a License Manager (software-based license).

To migrate from hardware-based license to software-based license, follow below process step-by-step.

- “Returning any existing license key for the USB hardware device or license manager” on page 205.

You can return a license entitlement in the Rational License Key Center in several ways. The easiest method is to use the View Keys by Host link. Also, you can use the Return Keys link.

For z Systems Development and Test Environment, the term host in the Rational License Key Center refers to the USB hardware device that is uniquely identified by its serial number.

1. Log in to Rational License Key Center at <https://licensing.subscribenet.com/control/ibmr/login>, and select your account.
  2. On the left side of the screen, select **View Keys by Host**.
  3. Select the serial number of the USB hardware device you want to work with. This serial number is in the Host column.
  4. A table is displayed with data for the USB hardware device selected. At the far right of the table, click the Change link.
  5. A list of devices with license entitlements that are assigned to them from the same Order Line is displayed. Locate the serial number of the USB hardware device you are working with, and click Return. A message is displayed to confirm that the license entitlements were successfully returned.
- “Uninstalling a previous version of z Systems Development and Test Environment” on page 46.

Before configuring the license manager, you must uninstall the product license server by following the steps in this topic.

- “Installing and starting the license manager” on page 68.  
To install the license manager, follow the steps in this topic.
- Issuing command to reset the UIM server.  

```
uimreset -l
```
- “Activating a license manager” on page 71.  
To activate the software license manager, follow the steps in this topic.
- “Activating a license manager with Rational Tokens” on page 73.  
If you want to use the Rational tokens, follow the steps in this topic.
- “Activating and configuring a license manager client” on page 77.  
To configure and activate the license manager client, follow the steps in this topic.

---

## Accessibility features

Users who have a physical disability, such as limited vision, can review the available accessibility features to use their information technology products successfully.

Accessibility features are product dependent and might include one or more of the following aspects:

- Keyboard-only operation
- Screen reader usage
- Color and typeface preferences

**Note:** The accessibility features mentioned here apply to the Windows operating system. Some of these features might also work on Linux, but are not officially supported.

## z Systems Development and Test Environment

You can run many optional, supporting programs on the z Systems Development and Test Environment emulator. A Voluntary Product Accessibility Template (VPAT) is available for an optional program upon request.

### Keyboard shortcuts for the help system in the product

You can use shortcuts to control the help system by using the keyboard.

Table 15. Help system keyboard shortcuts

Key combination	Context	Function
F6	Anywhere in the help browser	Puts focus in the next browser frame.
F6+Shift	Anywhere in the help browser	Puts focus in the previous browser frame.
Right Arrow	Navigation tree	Expand section
Left Arrow	Navigation tree	Collapse section
Down Arrow or Tab	Navigation tree	Move to next topic node
Up Arrow or Shift+Tab	Navigation tree	Move to previous topic node
Enter	Navigation tree	Displays the selected topic in the Content frame
Tab	Content frame	Next link or toolbar icon
Home	Content frame	Move to top of frame
End	Content frame	Move to bottom of frame
Alt+Left Arrow	Content frame	Back
Alt+Right Arrow	Content frame	Forward
Ctrl+P	Content frame	Print

### Example

To open a topic by using keyboard shortcuts and have the content read by a screen reader:

1. Start the screen reader application.
2. Open the help system in the browser.
3. Press F6 three times to move the focus into the Contents pane.
4. Press Tab to navigate to a container that you want to open. Example:  
Installing
5. Press the Right Arrow to expand the section.
6. Press Tab to navigate to a topic that you want to open.
7. Press Enter to open the topic contents.
8. Press F6 three times to move the focus into the topic pane. If you are using Mozilla Firefox, the topic contents are read by the screen reader application.
9. If you are using Internet Explorer, press the Down Arrow to make the screen reader application read the topic contents.



---

## Chapter 12. z Systems Development and Test Environment Glossary

Find terms that are used in the z Systems Development and Test Environment IBM Knowledge Center.

### **activation**

See “USB hardware device activation” on page 234.

### **clientconfig**

A program (Linux command) that can be used to change parameters in the client XML file.

### **client configuration file**

A file (in XML format) used by zPDT to obtain parameters to access both an SHK License Server and a UIM server. It is at this location:  
`/usr/z1090/bin/sntlconfig.xml` .

A different client configuration exists for an LDK server.

### **generated license file**

In the Rational License Key Center, the file that is generated and applied to the USB hardware device to enable z Systems Development and Test Environment operation.

### **identification**

A serial number and instance number, as stored by the z Systems STIDP instruction. The instance number is similar to an LPAR number on a larger z Systems.

### **instance**

A single installation of z Systems Development and Test Environment on a physical or virtual machine.

### **instance number**

A number in the range of 1 and 255 assigned to each zPDT instance on a base Linux machine. Each zPDT instance must operate under a different Linux user ID and the instance number is assigned to the user ID. The instance number is used in the same manner as the LPAR number on a larger System z®.

### **license**

The product entitlement that you purchased and the corresponding terms and conditions under which use is allowed.

A technical indicator on the USB hardware device that is used to activate the hardware device and allow an emulated z Systems instance to operate. The zPDT Guide and Reference and the z Systems emulator messages use the words license and licenses to refer to these indicators stored on the hardware device. You can use a remote license server with the offering that derives its name from this particular usage of the word.

In the Rational® License Key Center, a quantity of product entitlement parts purchased.

In zPDT, a logical function that enables one z Systems CP for a zPDT system. Multiple CPs require multiple licenses. The token functions provide licenses.

**license entitlement**

The Rational License Key Center interface and the quantity of entitlement parts purchased.

**license key file**

A file that represents the user's license entitlement. License key files must be obtained, installed, and available to operate the product.

**license manager**

The remote z Systems Development and Test Environment that distributes authentication by using software-based license keys.

See also "license server."

**license manager update file**

The specific license key file that is generated in the Rational License Key Center and installed on a license manager by using an `update_license` command. This file is used for software-based licensing.

**license monitor**

A web browser interface that displays information about Sentinel Keys and clients that use them. It is accessed at port 7002 on a Linux system that runs a license server, but might not be functional on recent Linux distributions.

**license server**

The remote server that is set up to distribute authentication from a single high-capacity USB hardware device to remote instances of the product as an alternative to attaching discrete USB hardware devices to each computer that hosts one or more instances of the product.

The License Development Kit–Software License (LDK-SL) server that distributes software-based license keys without the need for a USB hardware device. This server is also referred to as a license manager.

The Rational License Key Server that is used to share license entitlements among several IBM® products.

In zPDT, a network-accessible service that manages and dispenses zPDT licenses from a token or a "software-only" service. It operates as a Linux daemon and is automatically started (after it is installed) when Linux is started. A "local" zPDT installation internally accesses the license server through internal TCP/IP. Remote license servers are accessed through network TCP/IP.

**local mode**

A situation in which a remote license server and a UIM server are not used. zPDT obtains its serial number from the local UIM database. The client configuration XML file specifies localhost as the Contact Server. In local mode, the serial number might be a previously assigned number from a server or from a token. If an existing serial number is not present, the serial number is taken from the local token.

**local to remote**

A situation in which the serial number (in the local client database) was previously obtained from a local token, but zPDT is now configured to run with remote servers. If the remote license server determines that the serial number is valid and not being used elsewhere, that serial number is used.

**product entitlement**

The terms and conditions under which the product can be run by the licensee.

**product license server**

See “license server” on page 232.

**random serial number**

A serial number that is unique, but is not tied to a token serial number. The UIM server generates and assigns these numbers. A random serial number can be used (by zPDT) with a license from any token. Do not take the “random” word too literally; in this case it means that tokens with serial numbers other than the one used to set the UIM serial number might be used. It does not mean you can select a random number.

Regarding serial numbers that are assigned from a token, in some cases (such as a “simple local system”) the z Systems serial number that is used by zPDT is taken from the token.

**Rational License Key Server**

The license server used exclusively to share Rational Token license entitlements among a set of IBM products.

**Rational License Server**

In zPDT, provides controlled access to multiple IBM software products and might be used along with zPDT license servers. The Rational License Server has no relation to zPDT license servers.

**remote mode**

The zPDT instance obtains licenses and UIM identification from a remote license server and UIM server.

**remote to local**

A situation in which the serial number was previously assigned by a UIM server (and stored in the local client UIM database) and zPDT is now being used without remote servers. In this case, the previously assigned serial number is used and the serial number of the local token (which must be present to provide a license) is ignored.

**SafeNet**

The company that provides the USB keys and the software that directly supports them. The software includes the USB driver, the license manager, and a web interface to the license manager. The owning company is now Gemalto N.V., but the SafeNet name is used with the products that are described in this IBM Knowledge Center.

**SafeNet Sentinel Key**

The USB “token” from the SafeNet company. This token provides zPDT license information.

**serial number**

A value in the range of 1 and 65535 (4 hex digits). The serial number is assigned by the UIM function to the base Linux and used by zPDT to provide the z Systems serial number.

**server configuration file**

A file (in XML format) used by the SHK Sentinel Key Server to obtain networking and logging parameters. It is at this location: /opt/safenet\_sentinel/common\_files/sentinel\_keys\_server/Sntlconfigsrvr.xml

**software-based license server**

See “license server” on page 232.

**time cheat**

The Sentinel Key records the current date and time each time the key is

accessed. If the Linux system clock contains a time earlier than the last recorded time in the token, the license is unusable.

**token** See “USB hardware device.”

A type of IBM Rational product entitlement that allows great flexibility in the deployment and use of associated, purchased products. The product documentation uses the phrase Rational Token to refer to IBM Rational Token licensing.

In zPDT, another term for a SafeNet Sentinel Key. The terms token, key, SafeNet key, and Sentinel key are used interchangeably.

**token serial number**

The license information in the token contains a unique serial number that is assigned by IBM. This serial number might be used as the basis for the z Systems CP serial number in some cases.

**UIM or unique identification manager**

A server (or local function of zPDT) that helps maintain unique enterprise-wide z Systems serial numbers for zPDT systems. The license server and the UIM server (or local function) are separate but parallel functions.

**UIM client**

Each Linux machine that runs zPDT has a client function. In a local operation, a remote UIM server might not be involved. The UIM client might operate solely from the local UIM database.

**UIM database**

A file that contains UIM information. The files are not directly editable. UIM databases are of two types. One exists in every Linux zPDT machine, and the other exists in a UIM server (if it is used). The local database (on a zPDT client) is at this location: /usr/z1090/uim/uimclient.db.

**UIM server**

A centralized service that maintains unique zPDT serial numbers for multiple zPDT machines within an enterprise. Clients access the server through TCP/IP. The server runs under a normal Linux user ID (and not under root).

**update file**

The specific license key file that is generated in the Rational License Key Center and applied to the USB hardware device to activate it.

**USB hardware device**

The device that is required by z Systems Development and Test Environment to operate the product. The zPDT Guide and Reference and z Systems emulator messages use the words token and tokens to refer to these hardware devices. In such places, you can find further reference to the 1091 token that is intended for use with the z Systems Development and Test Environment offering.

**USB hardware device activation**

The condition of the hardware device and its readiness for operation with the emulator.

**USB server**

A driver that is provided by SafeNet to access tokens on USB ports. It operates as a Linux daemon and is installed when zPDT is installed.



**UUID** A universally unique identifier. It is obtained from the Intel machine BIOS. It is used to uniquely associate a UIM serial number with a particular machine.



---

# Index

## Special characters

/etc/sysctl.conf, updating 54  
/tmp and /u, create new zFS files  
for 114, 131  
.upw update files 95  
.upw update files, Stand-alone 95  
\$HOME/.bashrc, updating 54  
135

## A

access, product license server 98  
accessibility  
IBM support 6  
keyboard shortcuts  
help system 229  
overview 228  
activate a USB hardware device for  
v9.0 95  
activate a USB hardware device for  
v9.1 94  
activating the product  
license manager 61  
USB hardware device 61  
Activating the USB hardware device 93,  
206  
activation 87  
activation process 65  
activation, license 87  
activation, Parallel Sysplex 86  
activation, Rational Token 73, 86  
activation, replacement update file 91  
activation, troubleshooting USB hardware  
device 203  
activation, USB hardware device 85, 87  
activation, USB hardware device and  
Sysplex 95  
ADCD.Z111S.PROCLIB 112, 131  
administering the license manager 69

## B

Backup license managers 70

## C

capacity, hardware device 81  
capacity, license server 68  
Configuration overview 113, 131  
configuration, USB hardware device 85  
configuring a working z/OS system 129  
connections, troubleshooting Rational  
License Key Servers 202, 207  
consideration, security 65  
considerations, terminology 64, 79  
coupling facility, enabling in the device  
map 184

## D

dates, expiration 82  
device activation 85, 94, 95  
device activation, troubleshooting 203  
device map, defining 103  
device map, enabling the coupling  
facility 184  
device, activating 93, 206  
devices, high capacity 81  
devices, low capacity 81  
devices, serial number 82

## E

emulators not native to the Linux 122,  
141  
enabling product operation 61  
enabling the coupling facility in the  
device map 184  
entitlement, license 64, 79  
entitlement, product 64, 79  
environment, starting z Systems  
Development and Test  
Environment 107, 127  
example, expiration notification 82  
expiration 82  
expiration dates 82  
expiration notification example 82  
expiration notification sample script 84  
expiration notification sample script  
output 83  
expiration notification sample script  
setup 83

## F

file, update 92, 93

## G

getting the update file 89

## H

hardware device activation 85, 94, 95  
hardware device activation,  
troubleshooting 203  
hardware device serial number 82  
hardware device, activating 93, 206  
hardware device, capacity 81  
hardware device, high capacity 81  
hardware device, low capacity 81  
hardware device, state 81  
hardware, required 37, 39  
help  
keyboard shortcuts 229  
high capacity devices 81  
host ID and host name of the license  
server 69

## I

IBM support  
accessibility 6  
IEASYSxx 112, 130  
initial Rational Token license 92  
initial Rational Token license, Steps to  
get 92  
initial update file, Steps to get 92  
installation, checking 55  
installing 37, 55  
installing and starting the license  
manager 68  
IPL process 112, 130

## J

JES NJE connectivity, defining 124, 143

## K

key 79  
key, activating 93, 206  
keyboard shortcuts  
help system 229

## L

license 64, 79  
license entitlement 64, 79  
license file replacement 91  
license key 79  
license key expiration 66  
license key, activating 93, 206  
license keys  
ADCD 66, 85  
license manager  
installing and starting 68  
license server  
host ID 69  
host name 69  
license server, capacity 68  
Linux routing, setting up 117, 135  
Linux setup 42  
LOADxx 112, 130  
log on, z/OS 114, 131  
logon procedure, defining a new 123,  
142  
Logon to TSO 122, 141  
low capacity devices 81

## M

migrating a working z/OS system 111  
MSTJCLxx 112, 131

## N

new logon procedure, defining 123, 142  
Next steps 101  
non-native Linux emulators 122, 141  
notification example, expiration 82

## O

obtain an update file 89  
Obtain an update file 87  
Obtaining update files for Rational Tokens 91  
optional tasks 122, 141

## P

Parallel Sysplex activations 67, 86  
pointing to a Rational License Key Server 206  
prerequisites 37  
prerequisites, Linux setup 42  
process, activation 65  
product entitlement 64, 79  
product license server access, set up 98  
product license server, set up 97  
product operation  
  enabling 61  
PROFILE.TCPIP, modifying 118, 136

## R

Rational License Key Center 89  
Rational License Key Server, pointing to 206  
Rational License Key Servers,  
  troubleshooting connections 202, 207  
Rational Token 64, 79  
Rational Token activation 67, 73, 86  
Rational Token license, initial 92  
Rational Token license, replacement 93  
Rational Tokens 205  
Rational Tokens, update file 91  
replacement license file 91  
replacement Rational Token license 93  
replacement Rational Token license, Steps to get 93  
replacement update file 91  
replacement update file, Steps to get 93  
requirements, hardware 39  
routing, Linux, setting up 117, 135

## S

sample script output, expiration notification 83  
sample script setup, expiration notification 83  
sample script, expiration notification 84  
script output, expiration notification sample 83  
script setup, expiration notification sample 83  
script, expiration notification sample 84  
security consideration 65  
serial number, hardware devices 82  
serial number, USB device 82  
set up product license server access 98  
Set up the product license server 97  
shortcuts  
  keyboard  
    help system 229  
  software, required 41  
Stand-alone .upw update files 95  
starting z Systems Development and Test Environment environment 107, 127  
state of the hardware device 81  
Steps to get the initial update file 92  
Steps to get the Rational Token license 92, 93  
Steps to get the replacement update file 93  
support  
  accessibility 228  
SYS1.PROCLIB 112, 131  
sysctl.conf, updating 54  
Sysplex activation 95  
Sysplex and USB hardware device activation 95  
system configuration, introduction 111, 129  
system procedure library 112, 131  
system shutdown script, altering 123, 142  
system startup script, altering 123, 142

## T

TCP/IP, setting up 117, 134  
TCPIP.DATA, modifying 119, 137  
TCPPARMS files, modifying 118, 136  
terminology considerations 64, 79  
token 64, 79  
Tokens 205  
troubleshooting and support 191  
troubleshooting connections, Rational License Key Servers 202, 207  
troubleshooting USB hardware device activation 203

TSO logon 122, 141  
TSO user IDs, creating 122  
TSO user IDs, creating new 141

## U

uninstalling the license manager 69  
update file 64, 79, 89  
update file, initial 92  
update file, obtain 87  
update file, Rational License Key Center 89  
update file, replacement 91, 93  
update file, Rational Tokens 91  
update files, Stand-alone .upw 95  
USB devices, high capacity 81  
USB devices, low capacity 81  
USB devices, serial number 82  
USB hardware device 79, 87, 93, 206  
USB hardware device activation 79, 85, 94, 95  
USB hardware device activation, troubleshooting 203  
USB hardware device and license activation 87  
USB hardware device and Sysplex activation 95  
USB hardware device configuration 85  
USB hardware device licensing 79  
USB hardware device serial number 82  
USB hardware device, activating 93, 206  
USER.PROCLIB 112, 131  
USER.TCPPARMS, modify TCP/IP procedures to point to 120, 138

## V

V9.0 USB hardware device activation 95  
V9.1 USB hardware device activation 94

## W

working z/OS system, configuring 129  
working z/OS system, migrating 111

## Z

z Systems Development and Test Environment environment,  
  starting 107, 127  
z/OS system configuration, introduction to 111, 129  
z/OS, logging on 114, 131  
zFS file creation, /tmp and /u 114, 131