

*IBM Z Development and Test
Environment v12.0.5*



Contents

- IBM Z Development and Test Environment..... 1**

- Overview.....2**
 - What's new in ZD&T V12.0.5..... 3
 - Learning resources.....4
 - Notices.....5
 - Trademarks.....6
 - Terms and conditions for product documentation.....6

- Personal Edition..... 7**
 - Checklist.....9
 - Planning.....11
 - Prerequisites..... 11
 - Installation packages..... 12
 - Installing Personal Edition.....16
 - Enabling a license key.....18
 - Obtaining an update file from Rational License Key Center..... 18
 - Applying the update file to the ZD&T Personal Edition machine..... 19
 - Returning an existing license key..... 20
 - Configuring Z instances..... 20
 - Setting up the ADCD package.....20**
 - Creating the device map..... 21
 - Starting Z instances..... 21
 - Stopping Z instances22
 - Configuring the network..... 23
 - Updating Personal Edition..... 26
 - Uninstalling the earlier versions of ZD&T 27
 - Updating Personal Edition to the latest version..... 28
 - Uninstalling Personal Edition.....29

- Enterprise Edition.....30**
 - Checklist.....34
 - Planning.....37
 - Prerequisites..... 37
 - Installation packages..... 41
 - Enabling the license servers.....44
 - Software-based licensing.....46
 - Hardware-based licensing..... 54
 - Installing Enterprise Edition..... 61
 - Default authentication and authorization.....62
 - Installing ZD&T emulator.....62
 - Optional: Installing the z/OS files.....63
 - Starting and accessing the Enterprise Edition web server..... 64
 - Authentication for the Enterprise Edition application.....64
 - User's Guide..... 65
 - Enterprise Edition User's Guide..... 65
 - REST APIs User's Guide..... 97
 - ZD&T response error codes..... 110
 - Updating Enterprise Edition..... 124
 - Uninstalling Enterprise Edition.....124

Troubleshooting.....	125
Setting up a Windows machine as an FTP server.....	125
Troubleshooting the configuration of the Application Discovery server	125
Troubleshooting the application components creation from IBM Application Discovery projects.....	125
Scenarios.....	126
Provisioning CICS subsystem from a source Z.....	126
Parallel Sysplex.....	127
Checklist.....	131
Planning.....	134
Prerequisites.....	134
Installation packages.....	136
Enabling the License Server.....	139
Software-based licensing.....	141
Hardware-based licensing.....	149
Installing Parallel Sysplex.....	155
Activating ZD&T Parallel Sysplex instances.....	156
Software-based license.....	156
Hardware-based license.....	158
Configuring Z instances.....	159
Setting up the ADCD and z/VM package.....	159
Creating the device map.....	159
Starting z/VM.....	160
Starting Z instances.....	162
Stopping Z instances and z/VM.....	163
Updating Parallel Sysplex.....	163
Uninstalling the earlier versions of ZD&T	163
Updating the installed Parallel Sysplex.....	165
Uninstalling Parallel Sysplex.....	166
Troubleshooting and support.....	166
Finding known problems.....	166
IBM Software Support.....	167
Obtaining the latest editions of Redbooks.....	167
Troubleshooting tips.....	167
Installing ZD&T.....	168
USB Hardware device activation and licensing.....	169
Installing, operating, and uninstalling the Software-based License Server.....	173
Installing z/OS volume images.....	173
Defining the ZD&T machine characteristics (DEVMAP).....	174
Starting and stopping ZD&T.....	174
z/OS networking issues.....	174
Performance	176
Connecting with Rational License Key Servers.....	177
Troubleshooting Software-based License Server and USB hardware device activation	178
Reference.....	178
z/OS 2.4 ADCD reference.....	178
ADCD z/OS V2R4 November Edition of 2019.....	178
Hints and tips to manage the z/OS 2.4 ADCD.....	178
Customizing ADCD for ZD&T.....	179
NetView automation for ZD&T.....	183
Setting up Rational Tokens.....	185
Installing, configuring, and administering a Rational License Key Server.....	185
Updating the device map to point to a Rational License Key Server.....	185
Troubleshooting connections with Rational License Key Servers.....	186
zPDT license servers.....	186

Concepts.....	188
Client installation and configuration	191
Server installation and configuration.....	193
Notes.....	195
Scenarios.....	196
Renewing licenses.....	201
Searching servers.....	202
Allocating multiple licenses.....	202
Glossary.....	202
Index.....	207

IBM Z Development and Test Environment

This page provides an entry point to product information about ZD&T. 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.

Getting Started

[Product overview](#)

[What's new](#)

ZD&T V12.0.5 now supports z15 hardware, and ADCD is updated to z/OS 2.4. All installers for all editions are updated with the latest emulator version.

[Personal Edition](#)

ZD&T Personal Edition enables a single user to run an IBM Z distribution on a personal computer.

[Enterprise Edition](#)

ZD&T Enterprise Edition enables enterprises to host an IBM Z distribution on low-cost Intel-based x86 machines. It creates an environment for mainframe application demonstration, development, testing, and education without Z mainframe hardware, enables z/OS, middleware, and other z/OS software to run on Intel and other compatible computers, and emulates z/Architecture with virtual I/O and devices. Enterprise Edition also provides a web-based interface. You can extract, deploy, and manage the application images from an existing Z or ADCD packages.

[Parallel Sysplex](#)

Parallel Sysplex is a cluster of IBM mainframes that act together as a single system image with z/OS. Used for disaster recovery, Parallel Sysplex combines data sharing and parallel computing to allow a cluster of up to 32 systems to share a workload for high performance and high availability.

Learn more

[ZD&T product page](#)

[Enterprise DevOps for z Systems cloud-based trial](#)

[ZD&T product page on IBM developerWorks](#)

[Liberate mainframe development teams with ZD&T](#)

[ZD&T videos on IBM developerWorks](#)

Connect with experts

[developerWorks forum for ZD&T](#)

[IBM Z Development and Testing Hub](#)

[ZD&T blogs on IBM developerWorks](#)

Find documentation

[ZD&T Library](#)

[ZD&T Fix List](#)

[ZD&T e-assemblies to download from Passport Advantage](#)

[Find PDF documentation](#)

[zPDT Guide and Reference](#)

[IBM zPDT 2017 Sysplex Extensions](#)

Find support

[IBM Support](#)

[IBM Software Support home page](#)

Overview

IBM Z Development and Test Environment (ZD&T) runs a 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 Z mainframe hardware. It enables z/OS, middleware, and other z/OS software to run on Intel and other compatible computers, and emulates z/Architecture with virtual I/O and devices.

Important: ZD&T cannot be used for production workloads of any kind, nor robust development workloads, production module builds, preproduction testing, stress testing, or performance testing.

ZD&T is offered in three different editions:

ZD&T Personal Edition

ZD&T Personal Edition enables a single user to run an IBM Z distribution on a personal computer. For more information about ZD&T Personal Edition, see [“Personal Edition” on page 7](#).

ZD&T Enterprise Edition

ZD&T Enterprise Edition enables enterprises to host an IBM Z distribution on low-cost Intel-based x86 machines. Enterprise Edition provides a web-based interface. You can extract, deploy, and manage the application images from an existing Z or ADCD packages. For more information about ZD&T Enterprise Edition, see [“Enterprise Edition” on page 30](#).

ZD&T Parallel Sysplex

ZD&T Parallel Sysplex can be used to enable a Sysplex environment that is running within z/VM[®]. For more information about ZD&T Parallel Sysplex, see [“Parallel Sysplex” on page 127](#).

The following table lists the differences of three ZD&T editions.

Feature	Personal Edition	Enterprise Edition	Parallel Sysplex
License type	USB	<ul style="list-style-type: none">• USB¹• Software-based license• Authorized User Single Session (AUSS)• Rational[®] Tokens	<ul style="list-style-type: none">• USB¹• Software-based license• Rational Tokens
Single user	Yes	Yes	Yes
Multiple users	No	Yes	Yes
Scalable ²	No	Yes	Yes
Cloud/VM support ³	No	Yes	Yes
Web-based interface	No	Yes	No
Native Linux interface	Yes	Yes for advanced Linux users	Yes
Dataset transfer support	No	Yes	No
Volume transfer support	No	Yes	No

Table 1. Differences in three ZD&T editions (continued)

Feature	Personal Edition	Enterprise Edition	Parallel Sysplex
Integration with IBM AD	No	Yes	No
OpenStack Cloud support	No	Yes	No

Note:

1. The USB licensing for ZD&T Enterprise Edition and Parallel Sysplex is end of market. If you did not purchase the USB license for Enterprise Edition and Parallel Sysplex, you cannot purchase this type of license now. If you purchased a USB license, you can continue to use the license until it is end of support.
2. You need to have enough Resource Value Units (RVUs) or Central Processors (CPs) that are available to scale your ZD&T environment. Also, your Linux hardware needs to support it.
3. ZD&T Enterprise Edition and Parallel Sysplex can work on cloud and VMs. You need to make sure that the underlying hardware is not changed by your cloud service provider. CPU or hard drive needs to be a static entity to ensure that all ZD&T editions work.

What's new in ZD&T V12.0.5

ZD&T V12.0.5 now supports z15 hardware, and ADCD is updated to z/OS 2.4. All installers for all editions are updated with the latest emulator version.

ZD&T Personal Edition

- A separate installer ZDT_Install_PE_V12.0.5.0.tgz is available to install Personal Edition. For more information, see [“Installing Personal Edition” on page 16](#).

ZD&T Enterprise Edition

- A separate installer ZDT_Install_EE_V12.0.5.0.tgz is available to install Enterprise Edition. For more information, see [“Installing Enterprise Edition” on page 61](#).
- A list of fixed defects, APARs, and RFEs is provided. For more information, see [Fix list for IBM Z Development and Test Environment](#).

ZD&T Parallel Sysplex

- A separate installer ZDT_Install_PS_V12.0.5.0.tgz is available to install Parallel Sysplex. For more information, see [“Installing Parallel Sysplex” on page 155](#).

Application Developers Controlled Distribution (ADCD)

Application Developers Controlled Distribution (ADCD) z/OS Version 2 Release 4 November Edition of 2019 is available with all ZD&T V12.0.5 editions.

The following products are added.

- IBM Migration Utility V4.1.0
- IBM Common Data Provider for z Systems® 2.1.0
- IBM Explorer for z/OS V3.2

The following products are updated with the latest available version.

- IBM Rational Team Concert® V6.0.6

- IBM UrbanCode® Deploy for z/OS V7.0.2
- IBM Z® Development and Test Environment Enterprise 12.0.4
- IBM Connect Direct for z/OS Standard Edition 6.0
- IBM Data Set Commander for z/OS 9.1
- IBM Z Workload Scheduler V9.5.0
- IBM OMEGAMON® Performance Management Suite for z/OS V5.5.1
- IBM Application Discovery for IBM Z V5.1.0
- IBM Operational Decision Manager 8.10

For more information about the new release, see [ADCD z/OS V2R4 November Edition of 2019 and Programs that are included in the IBM Z Development and Test Environment V12.0.5 offering](#).

For more information about the customization of ADCD for ZD&T, see [“Customizing ADCD for ZD&T” on page 179](#).

zPDT Emulator V1.10.55.04

The zPDT® driver that is used in all ZD&T V12.0.5 editions is updated to zPDT V1.10.55.04. The new emulator includes the following new functions and fixes.

- CFCC Prod Rel 24.00 Service Level 00.14.
- Crypto Express® 7s level, CCA version 7.0 from previous level CCA version 5.4.
 - The format of the master keys differs from the previous versions. zPDT will not convert the older form to the newer one. You must initialize the new crypto adapter level with the same master keys that were previously used if you need to access the previously encrypted data.
- z15 Architecture is supported.
 - MSA9 is not implemented.
 - The zPDT GA10 initial release has the following special considerations.
 - A normal IBM z15 system provides a DEFLATE function in two ways, as a simple instruction and as an asynchronous function.
 - A normal IBM z15 system does not support the zEDC adapter.
 - zPDT V1.10 does not provide either DEFLATE operation.
 - zPDT V1.10 supports the zEDC adapter, as was present in the zPDT GA9 release.
 - For your information, zPDT intends to support DEFLATE in a future fix pack.
- The command `sa_fenet_daemons_restart` was not scheduled in cron on Ubuntu.
- CKD locate record erase opcode is fixed.

For more information about the new release, see [IBM zPDT Guide and Reference](#).

Related information

[What's new in ZD&T V12.0.4](#)

[What's new in ZD&T V12.0.3](#)

[What's new in ZD&T V12.0.2](#)

[What's new in ZD&T V12.0.1](#)

[What's new in ZD&T V12.0.0.1](#)

[What's new in ZD&T V12.0.0](#)

Learning resources for ZD&T

Access web-based resources for learning more about ZD&T.

- [ZD&T product page](#)

- [IBM Z Trail program is powered by ZD&T](#)
- [developerWorks®: IBM Z Development and Testing Hub](#)
- [developerWorks forum for ZD&T](#)
- [zPDT Guide and Reference](#)
- [zPDT 2017 Sysplex Extensions](#)
- [Education Assistant](#)

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 10504-1785 US

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

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

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement 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 10504-1785 US

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 copyright notice as follows: © (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

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.

Personal Edition

ZD&T Personal Edition enables a single user to run an IBM Z distribution on a personal computer.

ZD&T Personal Edition can create an environment for mainframe application demonstration, development, testing, and education without Z mainframe hardware. It enables z/OS, middleware, and other z/OS software to run on Intel and other compatible computers.

To install ZD&T Personal Edition, you need to run the Personal Edition package on the 8086 Intel-based hardware with RHEL or Ubuntu operating system. This package can emulate z/Architecture® with virtual I/O and devices. For more information about hardware and software requirements, see [“Prerequisites” on page 11](#).

The following topology describes the various components of Personal Edition.



Figure 1. The topology and various components of Personal Edition

ZD&T Personal Edition provides the following features:

- Personalized application development and testing environment that can improve development infrastructure availability and flexibility
- Current levels of IBM z/OS software that can give access to new runtime capabilities for development and testing for a single user
- Mixed workload support for a single user that can help reduce development costs
- An approachable and portable environment for education on Z for a single user

License Key (USB)

ZD&T Personal Edition comes with a USB hardware device that contains the fingerprint that is used to identify the entitlement of user. After you install Personal Edition, you need to insert the USB hardware device to the USB port of the machine to enable the license. For more information, see [“Enabling a license key”](#) on page 18.

Application Developers Controlled Distribution (ADCD)

ZD&T Personal Edition comes with the Application Developers Controlled Distribution (ADCD) package, which is known as z/OS ADCD. A z/OS ADCD for ZD&T Personal Edition contains many of the products and subsystems that you might need to develop z/OS applications such as CICS®, DB2®, IMS, and z/OS compiler. Most of these products and subsystems can run without any customization, but some need customization or initialization.

For a complete list of z/OS products in ADCD, see [“ADCD z/OS V2R4 November Edition of 2019”](#) on page 178.

3270 terminal emulator

After Personal Edition and ADCD is installed successfully, you can IPL the z/OS distributions with the specified IPL parameters. Before the IPL process, you need to install a 3270 terminal emulator on your machine. The terminal emulator is not supplied with the package. You can download the x3270 Linux package on your machine, or purchase IBM Personal Communication Terminal Emulator to access Mainframe machine. You can run the emulator either on the Linux machine where you installed ZD&T

Personal Edition, or on your personal Windows, Mac devices to access the ZD&T z/OS distributions by the IP and port (3270) of the machine.

For the detailed steps to install, update, or uninstall ZD&T Personal Edition, see [“Checklist” on page 9](#).

Checklist

If you want to install, update, or uninstall Personal Edition, or update the license, you can follow the steps that are provided in the checklists.

To install ZD&T Personal Edition, complete the following steps:

Table 2. New installation checklist

Installing Personal Edition		Required/ Optional	Complete
1	“Prerequisites” on page 11 Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	<input type="checkbox"/>
2	Obtaining installation packages Describes the steps to download all required packages to install Personal Edition.	Required	<input type="checkbox"/>
3	“Installing Personal Edition” on page 16 Describes the steps to run the installation package.	Required	<input type="checkbox"/>
4	“Enabling a license key” on page 18 Describes the steps to obtain a license key, and apply the key to the machine.	Required	<input type="checkbox"/>
5	“Configuring Z instances” on page 20 Describes the steps to set up the ADCD package and create the device map to start Z instances.	Required	<input type="checkbox"/>
6	“Starting Z instances” on page 21 Describes the steps to start Personal Edition and Z instances.	Required	<input type="checkbox"/>
7	“Stopping Z instances” on page 22 Describes the steps to stop Personal Edition and z/OS images.	Optional	<input type="checkbox"/>
8	“Configuring the network” on page 23 Describes various ways to configure the network.	Optional	<input type="checkbox"/>

To update ZD&T Personal Edition, complete the following steps:

Table 3. Update checklist

Updating Personal Edition		Required/ Optional	Complete
1	<p>“Returning an existing license key” on page 20</p> <p>Describes the steps to return an existing hardware-based license key.</p>	Required	<input type="checkbox"/>
2	<p>“Prerequisites” on page 11</p> <p>Describes the software, hardware requirements, and the steps to set up the basic Linux system for newer version of ZD&T.</p>	Required	<input type="checkbox"/>
3	<p>Obtaining installation packages</p> <p>Describes the steps to download all required packages to install Personal Edition.</p>	Required	<input type="checkbox"/>
4	<ul style="list-style-type: none"> • If you installed ZD&T V11.0 or later versions, complete the following step: <ul style="list-style-type: none"> – “Updating Personal Edition to the latest version” on page 28 • If you installed zD&T V10.0.1 or earlier versions, complete the following steps: <ul style="list-style-type: none"> – “Uninstalling the earlier versions of ZD&T ” on page 27 – “Installing Personal Edition” on page 16 <p>Describes the steps to update Personal Edition.</p>	Required	<input type="checkbox"/>
5	<p>“Enabling a license key” on page 18</p> <p>Describes the steps to obtain a license key, and apply the key to the machine.</p>	Required	<input type="checkbox"/>
6	<p>“Configuring Z instances” on page 20</p> <p>Describes the steps to set up the ADCD package and create the device map to start Z instances.</p>	Required	<input type="checkbox"/>

To update the license, complete the following steps:

Table 4. License update checklist

Updating the license		Required/ Optional	Complete
1	<p>“Returning an existing license key” on page 20</p> <p>Describes the steps to return an existing hardware-based license key.</p>	Required	<input type="checkbox"/>
2	<p>“Enabling a license key” on page 18</p> <p>Describes the steps to obtain a license key, and apply the key to the machine.</p>	Required	<input type="checkbox"/>

To uninstall ZD&T Personal Edition, complete the following steps:

Table 5. Uninstallation checklist

Uninstalling Personal Edition		Required/ Optional	Complete
1	“Returning an existing license key” on page 20 Describes the steps to return an existing hardware-based license key.	Required	<input type="checkbox"/>
2	“Uninstalling Personal Edition” on page 29 Describes the steps to uninstall Personal Edition.	Required	<input type="checkbox"/>

Planning

Before you start with ZD&T Personal Edition, learn about the information about system requirements and installation packages, and the steps to install, update, or uninstall Personal Edition.

Prerequisites

Before you install or update ZD&T Personal Edition, you need to check the hardware and software requirements, and set up the base Linux system.

Hardware and software requirements

For a complete list of ZD&T hardware and software requirements, you can generate the report from [Software Product Compatibility Reports](#).

Setting up the base Linux system

Before you access the Linux Software repository to download the required software, you need to ensure the following prerequisites.

- You have access to internet on your machine.
- You have root or sudo access to the Linux operating system.
- Your Linux system is configured to download the required Linux packages. You can use `yum install` or `apt-get` commands to install the package that might be required during the installation procedure.
- Acquiring Linux knowledge

To set up the base Linux system, you need to have some Linux knowledge to ensure that you can go through some basic concepts and commands from internet. The following commands might be required to use when you install the package.

- Obtaining access to software repository to run YUM or `apt-get` commands

Make sure that you have access to software repository to run YUM commands on RHEL machine, or your Ubuntu machine have access to run `apt-get` commands. ZD&T Installer will install all required packages. However, if you don't want ZD&T installer to install the required packages that are listed below, you need to install the packages before you start ZD&T Installer.

- YUM commands on RHEL operating system

```
yum -y install iptables
yum -y install ftp
yum -y install libstdc++.i686
yum -y install perl
yum -y install zip
yum -y install unzip
yum -y install gzip
yum -y install bc
```

- apt-get commands on UBUNTU operating system

```
apt-get -y install iptables
dpkg --add-architecture i386
apt-get -y update
apt-get -y install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 lib32stdc++6
apt-get -y install ftp
apt-get -y install perl
apt-get -y install zip
apt-get -y install unzip
apt-get -y install gzip
apt-get -y install bc
apt-get -y install libasound2
apt-get -f install
```

- Setting up the system time

Set your hardware clock to Coordinated Universal Time (UTC) to avoid problems when Daylight Saving Time starts and stops. The License Server 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 the standard time.

- Installing a TN3270e client for the MVS™ console

The following TN3270e clients are examples of clients that can be used with the recent ZD&T offerings:

- x3270 (recent versions)
- Recent Personal Communications releases that run on the 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 ZD&T. 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
```

- Managing the firewall and other security functions

You must manage the firewall and other security functions that you install with your Linux. To simplify the configuration and operation, disable any firewall when you first work with zPDT. 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)
```

Note: Installer scripts changes iptables. You need to make sure that your Linux distribution is authorized by your network administrator to make the changes. For more information about what ports are opened during installation, see [“Installing Personal Edition” on page 16](#) and [“Configuring the network” on page 23](#).

Installation packages

Download the required installation packages from Passport Advantage before you install or update ZD&T Personal Edition.

To download the package, complete the following steps:

1. Log on to [Passport Advantage](#).
2. Select **Software Downloads** and **Media Access**.
3. Select **Program offering and agreement number**, and click **Continue**.
4. Enter the part description or part number, and click **Finder**.
5. Optionally, you can click the alphabetical order list to display and view the product by name.

6. Select **All Operating Systems** in the **Operating system** field, and **All Languages** in the **Languages** field. Then, click **Go**.
7. If you want to download the individual media from the list, click **Select individual files** to expand the list.
8. Verify the e-assemblies that you want to download with the list that is mentioned below in the table.

Note: Verify the integrity of downloaded ADCD packages by using the MD5SUM that is in the adcd.md5 and pe.md5 files.

	Name	Part No.	Package Type	File Name
1	IBM Z Development and Test Environment Personal Edition V12.0.5 Installation Multilingual	CC5FVML	ZD&T Personal Edition Installer	ZDT_Install_PE_V12.0.5.0.tgz
2	IBM Z Development and Test Environment Personal Edition V12.0.5 Installer checksum Multilingual	CC5FWML	checksum (md5 checksum)	pe.md5
3	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	CC59MML	ADCD November 2019 Edition RSU level 1909	A4RES1.ZPD
4	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	CC59NML	ADCD November 2019 Edition RSU level 1909	A4RES2.gz
5	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 3 of 19 - System volume Multilingual	CC59PML	ADCD November 2019 Edition RSU level 1909	A4SYS1.gz
6	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	CC59QML	ADCD November 2019 Edition RSU level 1909	A4CFG1.gz
7	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	CC59RML	ADCD November 2019 Edition RSU level 1909	A4USS1.gz
8	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	CC59SML	ADCD November 2019 Edition RSU level 1909	A4USS2.gz
9	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	CC59TML	ADCD November 2019 Edition RSU level 1909	A4PRD1.gz
10	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	CC59UML	ADCD November 2019 Edition RSU level 1909	A4PRD2.gz

	Name	Part No.	Package Type	File Name
11	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	CC59VML	ADCD November 2019 Edition RSU level 1909	A4PRD3.gz
12	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	CC59WML	ADCD November 2019 Edition RSU level 1909	A4PRD4.gz
13	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	CC59XML	ADCD November 2019 Edition RSU level 1909	A4DIS1.gz
14	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	CC5YML	ADCD November 2019 Edition RSU level 1909	A4DIS2.gz
15	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	CC59ZML	ADCD November 2019 Edition RSU level 1909	A4DIS3.gz
16	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	CC5A0ML	ADCD November 2019 Edition RSU level 1909	A4PAGA.gz
17	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	CC5A1ML	ADCD November 2019 Edition RSU level 1909	A4PAGB.gz
18	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	CC5A2ML	ADCD November 2019 Edition RSU level 1909	A4PAGC.gz
19	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	CC5A3ML	ADCD November 2019 Edition RSU level 1909	A4USR1.gz
20	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	CC5A4ML	ADCD November 2019 Edition RSU level 1909	SARES1.ZPD
21	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 19 of 19 - zCX Container volume	CC5A5ML	ADCD November 2019 Edition RSU level 1909	A4ZCX1.gz

	Name	Part No.	Package Type	File Name
22	IBM Z Development and Test Environment V12.0.5 with Rational Team Concert volume 6.0.6 Multilingual	CC5A6ML	ADCD November 2019 Edition RSU level 1909	A4BLZ1.gz
23	IBM Z Development and Test Environment V12.0.5 with DB2 V12 Part 1 of 2 Multilingual	CC5ABML	ADCD November 2019 Edition RSU level 1909	A4DBC1.gz
24	IBM Z Development and Test Environment V12.0.5 with DB2 V12 Part 2 of 2 Multilingual	CC5ACML	ADCD November 2019 Edition RSU level 1909	A4DBC2.gz
25	IBM Z Development and Test Environment V12.0.5 with DB2 V11 Part 1 of 2 Multilingual	CC5ADML	ADCD November 2019 Edition RSU level 1909	A4DBB1.gz
26	IBM Z Development and Test Environment V12.0.5 with DB2 V11 Part 2 of 2 Multilingual	CC5AEML	ADCD November 2019 Edition RSU level 1909	A4DBB2.gz
27	IBM Z Development and Test Environment V12.0.5 with CICS 5.4 Multilingual	CC5AFML	ADCD November 2019 Edition RSU level 1909	A4C541.gz
28	IBM Z Development and Test Environment V12.0.5 with CICS 5.5 Multilingual	CC5AGML	ADCD November 2019 Edition RSU level 1909	A4C551.gz
29	IBM Z Development and Test Environment V12.0.5 with IMS 14.1 Multilingual	CC5AHML	ADCD November 2019 Edition RSU level 1909	A4IME1.gz
30	IBM Z Development and Test Environment V12.0.5 with IMS 15.1 Multilingual	CC5AIML	ADCD November 2019 Edition RSU level 1909	A4IMF1.gz
31	IBM Z Development and Test Environment V12.0.5 with IBM Installation Manager 1.8.9 Multilingual	CC5AJML	ADCD November 2019 Edition RSU level 1909	A4INM1.gz
32	IBM Z Development and Test Environment V12.0.5 with DB2 Archive Logs Multilingual	CC5AKML	ADCD November 2019 Edition RSU level 1909	A4DBAR.gz
33	IBM Z Development and Test Environment V12.0.5 with Tivoli® Omegamon XE products 5.5.1 Multilingual	CC5ALML	ADCD November 2019 Edition RSU level 1909	A4KAN1.gz
34	IBM Z Development and Test Environment V12.0.5 with WAS V9.0 Part 1 of 2 Multilingual	CC5AMML	ADCD November 2019 Edition RSU level 1909	A4W901.gz
35	IBM Z Development and Test Environment V12.0.5 with WAS V9.0 Part 2 of 2 Multilingual	CC5ANML	ADCD November 2019 Edition RSU level 1909	A4W902.gz

	Name	Part No.	Package Type	File Name
36	IBM Z Development and Test Environment V12.0.5 Rocket Open Source Languages and Tools for z/OS	CC5APML	ADCD November 2019 Edition RSU level 1909	ZDTRKT.gz
37	IBM Z Development and Test Environment V12.0.5 ADCD Checksum Multilingual	CC5ARML	ADCD package MD5 checksum	nov2019_adcd_md5.txt

Installing Personal Edition

Learn about the steps to install ZD&T Personal Edition.

To install ZD&T Personal Edition, run the following steps from the root user ID.

1. Open the directory that contains the installer ZDT_Install_PE_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PE_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PE_V12.0.5.0.x86_64
```

6. Select **1** to install.
7. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
8. Optional: Enter y to install automatic network configuration. Or, enter n to decline the network configuration.

By entering y, your z/OS starts by 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. Before you enter y, you need to consider that the network configuration changes your Linux and networking functions, allows the Linux firewall to reroute TCP/IP packets to z/OS, and updates the TCP/IP specifications in the z/OS disks. For more information about network configuration, see [“Configuring the network”](#) on page 23.

9. Optional: Enter y to install all needed dependencies or enter n to decline.

By entering y, all required dependencies will be installed. The list of dependencies are mentioned in the [“Prerequisites”](#) on page 11. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

10. After the installation completes, run the following command to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep z1091
```

- UBUNTU operating system

```
dpkg -l | grep z1091
```

If the package is installed successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Important: The installer creates the `ibmsys1` as the non-root user ID to make ZD&T work, you need to switch to `ibmsys1` and change the password. If you like to create the non-root user ID, use the following commands for reference. Make sure that the non-root user ID is a part of group `zpdtd`.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpdtd -m ibmsys2
```

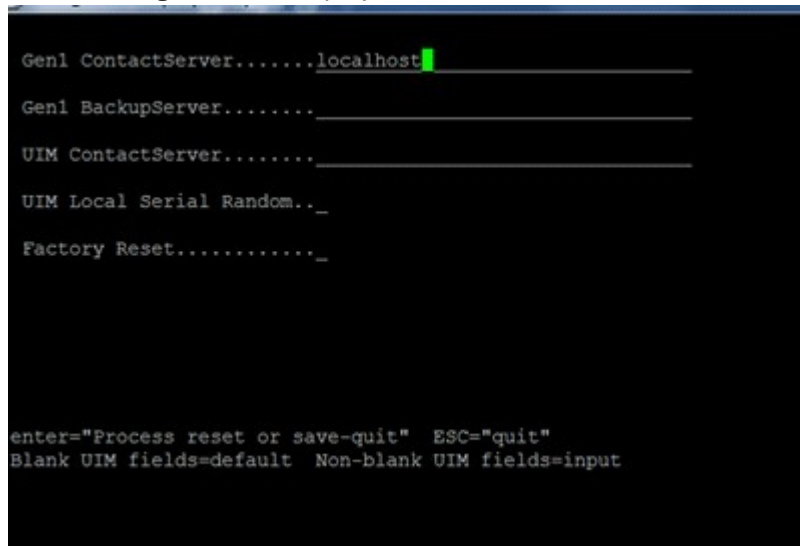
To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

11. Optional: Enter the following command from `/usr/z1090/bin` by using the root user ID.

```
./clientconfig
```

The following screen is displayed.



Note: The installer changes the iptables and Firewall. If you choose to install Personal Edition without network configuration, ports 1947, 3270, 9451 and 9450 are opened. You need to review the requirement with your security administration.

If you choose to install Personal Edition with network configuration, see [“Configuring the network”](#) on page 23 for more information about the ports that are opened.

Alternatively, after you complete steps 1 - 4, you can run the following commands to install ZD&T Personal Edition silently.

- To install Personal Edition without automated network configuration, run the following command:

```
./ZDT_Install_PE_V12.0.5.0.x86_64 --install
```

- To install Personal Edition with automated network configuration, run the following command:

```
./ZDT_Install_PE_V12.0.5.0.x86_64 --install --net
```

- To install the dependencies, run the following command.

```
./ZDT_Install_PE_V12.0.5.0.x86_64 --install --dep
```

Note:

- Silent installation does not install any required dependencies. You need to install the dependencies before you start the installer silently. For the list of dependencies, see “Prerequisites” on page 11.
- Silent installation does not display the option to review the license. The installation process assumes that you have reviewed the license before the installation. Read the license that is described in the installation media, or use the command line option to install Personal Edition interactively.

Enabling a license key

ZD&T Personal Edition 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. You must activate the USB hardware device with a license key file before you can use ZD&T Personal Edition or any of its components. To do this, obtain the license key file from the Rational License Key Center, and apply it to the USB hardware device to activate it. This grants you access to use ZD&T Personal Edition. 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.

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.

Figure 2. USB hardware device



Obtaining an update file from Rational License Key Center

Learn about the steps to obtain an initial update file from the Rational License Key Center.

1. Log in to [Rational License Key Center \(RLKC\)](#), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
2. On the left side of the screen, click **Get Keys**.
3. Select the product line for **IBM Rational Developer for System z Unit test**.
4. Select the license type that you want to apply to the key. If you purchased different license types, select **IBM Z SYSTEMS DEVELOPMENT AND TEST ENVIRONMENT PERSONAL EDITION (AU)**.
5. One or more boxes are displayed to show your orders that you can apply to one or more USB hardware devices.
6. Check the box next to the orders from which you would like to use licenses, and click **Next**.
7. Complete the table that is displayed. You must enter the USB hardware device serial numbers that you want to activate, enter 1 in both the **Number of Server Instances** and **Number of Licenses** fields. For more help, click the column headings in the table.
8. Click **Generate** at the bottom of the page. Then, a screen is displayed to download the update files.
9. Click **Download** for each update file that is generated to save the update files. Keep this file in the Linux machine where you installed ZD&T Personal Edition. 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: Use the binary transfer mode, but not a text mode such as ASCII. Also, spaces are not allowed in the path on the Linux machine.

Applying the update file to the ZD&T Personal Edition machine

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.

Note:

1. Never set the system clock to a future date or time when the USB hardware device is plugged in. Otherwise, the hardware device is damaged.
2. If the USB hardware device is attached to a computer with a clock that is set to a future date, the device remembers the future date. When the time is set back to the current time, the USB device does not recognize this change, and will not work until the clock reaches the previously set future time. If the clock was set to a future time or date, which causes ZD&T is unable to authenticate with the USB hardware device, 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).

To apply the update file to the ZD&T Personal Edition machine, complete the following steps:

1. Connect the USB hardware device to the computer that is running ZD&T Personal Edition.

Note: Connect only the USB hardware device that is being activated to your computer during this process. Otherwise, unpredictable results might occur.

2. Log in to the computer. If you are not running as a root user ID, enter `su` that is followed by the root password.
3. Change to the `/usr/z1090/bin` directory.

```
cd /usr/z1090/bin
```

4. Run the following command where `update-file.zip` is the full path and file name of the update file.

```
./Z1091_token_update -u update-file.zip
```

This command produces several messages that indicate that the update was successful.

5. After the update file is successfully applied, unplug the USB hardware key, and wait for at least 10 seconds. Then, reconnect the hardware key. It is now ready for routine ZD&T operation.
6. Restart the local SHK license server from a root user ID by using the command:

```
/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart
```

This step can ensure that the system discovers the newly applied licenses.

7. Optionally, verify that the licenses are now available on your USB hardware device by using the command:

```
./Z1091_token_update -status
```

8. Switch to non-root user ID, and enter the following command from `/usr/z1090/bin`

```
./uimcheck
```

If errors occur, enter the following command from `/usr/z1090/bin`.

```
./uimreset -l
```

Then enter the `./uimcheck` again to make sure that no error occurs.

Important: Even you have purchased license for more than one year. The license will expire after one year. You need to apply the update file again before the expiration date. After ZD&T is started, to verify

the effective Z serial number, and the expiration date of the license, enter the following token command from /usr/z1090/bin by using the non-root user ID (ibmsys1).

```
/home/ibmsys1/zVolumes  
./token
```

Returning an existing license key

For perpetual license entitlements, USB hardware device activations are set to expire one year from the date that 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 that 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 the 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. 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 way is to use the **View Keys by Host** link. You can also use the **Return Keys** link.

For ZD&T Personal Edition, 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 \(RLKC\)](#), 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 ID** column.
4. A table is displayed with data for the selected USB hardware device. 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 that you are working with, and click **Return**. A message is displayed to confirm that the license entitlements were successfully returned.

Configuring Z instances

You need to set up the ADCD package and create the device map to configure Z instances for development and test purposes.

Setting up the ADCD package

Before you configure Z instances, set up the ADCD package.

When you run the installer, a non-root user ID (ibmsys1) is created. The user ID is required to enable the z/OS images. You can also create other non-root user IDs.

To set up the ADCD package, complete the following steps:

1. Switch to the non-root user ID (ibmsys1).
2. Create a folder that is named `volumes` under `/home/ibmsys1`, and keep all ADCD volumes in the non-root user directory that you downloaded at step 2 of [“Checklist” on page 9](#).
3. Make sure that all volumes are owned by the non-root user ID. You can use the following command to change the owner and group.


```
chown ibmsys1 *
chgrp zpdt *
```

Note: Make sure that the folder that contains *.gz and .ZPD volumes are also owned by the non-root user ID.

4. Make sure that all volumes are executable. Otherwise, enter the following command:

```
chmod 755 *
```

5. Decompress all .gz volumes by using the Linux command.

```
gunzip *.gz
```

6. Decrypt .ZPD volumes by using the command from /usr/z1090/bin:

```
./Z1091_ADCD_install <path of volume>/A4RES1.ZPD <output path of volume>/A4RES1
```

For example,

```
./Z1091_ADCD_install
/home/ibmsys1/volumes/A4RES1.ZPD
/home/ibmsys1/volumes/A4RES1
```

For a complete list of z/OS products in ADCD, see [“ADCD z/OS V2R4 November Edition of 2019” on page 178](#).

Creating the device map

You can create the device map from a non-root user ID.

A sample program `create_devmap.pl` is available in the `/opt/ConfigGuideSample` directory where you have installed ZD&T. To create the device map from a non-root user ID, run the following command:

```
perl <path to command>/create_devmap.pl <path to disks> > generateddevmap
```

For example, `perl create_devmap.pl /home/ibmsys1/volumes > generateddevmap`

If you create the devmap manually, or edit the devmap that is created by the `create_devmap.pl` utility, you can run the following command from `/usr/z1090/bin` to verify it after you create the devmap manually.

```
awsckmap /home/ibmsys1/volumes/generateddevmap
```

For more information on the sections and syntax of device maps, see the "1090 Control Files" section of the [zPDT Guide and Reference](#).

Starting Z instances

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 ZD&T Personal Edition system with the `./runzpd` script that is provided in `/opt/ConfigGuideSample`.

To start Z instances, copy the `./runzpd` script from `/opt/ConfigGuideSample` to the directory where you have kept all ADCD volumes and devmap, and run the script from the non-root user ID (ibmsys1). For example, you copy the `./runzpd` script to the `/home/ibmsys1/volumes` directory, and run the script as shown below.

```
./runzpd -d <devmap name> -l <loadparm option>
```

For example, `./runzpdtd mygenerateddevmap -l cs`

Note: IPL the ZD&T z/OS with the loadparm CS for the first time.

If x3270 is not installed on your native Linux, and you use other terminal emulators, such as IBM Personal Communication, open the terminal emulator and configure it by specifying the IP (Linux machine IP) and port (3270) before you run the `./runzpdtd` command.

If you want to IPL the ZD&T manually, complete the following steps:

1. Switch to the non-root user ID (ibmsys1).
2. Change the path to `/home/ibmsys1/volumes` where you kept your devmap.
3. Enter the following command to start ZD&T Personal Edition.

```
awsstart <devmap> --clean
```

4. Open the terminal emulator, and provide the IP of your Linux machine. Then, set the port as 3270 if the automatic network configuration is not used, or 23/3270 if the automatic network configuration is used.

Then, the terminal emulator screen is displayed, and the device #0700 is listed. If you open another terminal emulator, the device #701 is listed.

5. Enter the IPL command:

```
ipl 0a80 parm 0a82<loadparm option>
```

For example, `ipl 0a80 parm 0a82CS`

6. Wait for a couple of seconds, the terminal emulator with #700 is switched to the master console. If the IPL process is successful, the terminal emulator with #701 is switched to the login screen.

For more information, see [ADCD z/OS V2R4 November Edition of 2019](#).

The `ipl` statement contains three pieces of information. 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 that holds IPL configuration files, and the LOADPARM. The suffix identifies the LOADxx member that is used to start z/OS.

In case the connection to master console is lost, or the master console is not displayed after you issue **awsstart** command, issue the following commands in the Linux machine where you issued the **awsstart** command by using the non-root user ID.

```
oprmsg 'VARY CN(*),ACTIVATE'  
oprmsg 'vary 0700,console'
```

Stopping Z instances

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 ZD&T Personal Edition contains sample startup and shutdown scripts for the systems that are available and active in that distribution.

You need to remember the IPL loadparm that is used to start ZD&T Personal Edition. To shut down ZD&T Personal Edition, use the respective shutdown command. All shutdown scripts are in the ADCD.Z24A.PARMLIB(SHUT**).

To shut down Z instances, complete the following steps:

1. Enter `S SHUTXX` in the master console.

XX is the last two characters of shutdown script that is generally equivalent to loadparms. To reply to any pending message, enter this command from the master console. If you do not have a master

console, enter /S SHUTXX in SDSF, or enter opxmsg "S SHUTXX" in the Linux machine. For the list of loadarm, see the loadparms section that is documented in the [Loadparms options](#).

2. During the shutdown process, you might need to respond to z/OS console messages, for example, when IMS™, TSO, or z/OS UNIX are stopped. To check which programs are still running, enter D J, L. Ensure that VTAM® and all subsystems are ended.
3. Enter \$P JES2 in the master console to close JES2.
4. Enter QUIESCE in the master console.
5. Enter awsstop by using the non-root user ID (ibmsys1) to completely shut down ZD&T Personal Edition.

Note: Do not make any change in the ADCD.Z24A.* library. If you want to edit any parm or proc, ADCD has USER.Z24A.* as a high-level concatenated library. You can create your own proc or parm in the library.

Configuring the network

You can either route all required traffic from ZD&T to Z image that is running on ZD&T, which is also known as automatic network configuration. Or you can assign the static IP to the Z Image.

Automatic network configuration

In this scenario, all the Linux ports are routed to ZD&T z/OS ports except a few ports that are required for the Linux machine.

If you decide to use this method, the network configuration can be installed automatically when you run the ZD&T Personal Edition installer. For more information, see [“Installing Personal Edition” on page 16](#).

If you do not decide to use the method when you run the installer, the network configuration still can be installed by running the scripts in the /opt/ConfigGuideSample. Run the following script.

```
./zdt_config_network10
```

Automatic network configuration makes the following changes to your Linux operating system.

```
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:9451
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:9450
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:1947
ACCEPT tcp -- 0.0.0.0/0 0.0.0.0/0 tcp dpt:3270

Chain FORWARD (policy ACCEPT)
target prot opt source destination
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:0:21
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:23:1946
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:1948:2021
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpt:2022
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:2023:3269
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:3271:5900
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:5903:9449
ACCEPT tcp -- 0.0.0.0/0 10.1.1.2 tcp dpts:9452:6553
```

Assigning a static IP to ZD&T z/OS

In this scenario, ZD&T z/OS was assigned a separate IP within the same subnet of the Linux machine where ZD&T Personal Edition is installed. In this way, you have 2 separate IP addresses. One is assigned for the Linux machine, and the other is assigned for z/OS images.

To configure the network, complete the following steps:

1. Obtain the static IP within the same subnet of your Linux machine.

- Run the **create_devmap** command from /opt/ConfigGuideSample with a host name. You can provide a host name that you want.

```
Perl create_devmap.pl <disk path> -h <host name> > (<devmap name>)
```

For example, Perl create_devmap.pl /home/ibmsys1/volumes -h zos-dev4 >generateddevmap.

- Verify whether the DEVMAP is displayed as shown below.

```
[manager] # tap0 define network adapter (OSA) for communication with Linux
name awsosa 0009 --path=A0 --pathtype=OSD --tunnel_intf=y # QDIO mode
device 400 osa osa --unitadd=0
device 401 osa osa --unitadd=1
device 402 osa osa --unitadd=2

[manager] # eno1 define network adapter (OSA) for communication with network
name awsosa 0019 --path=F0 --pathtype=OSD # QDIO mode
device 404 osa osa --unitadd=0
device 405 osa osa --unitadd=1
device 406 osa osa --unitadd=2
```

- IPL the ZD&T z/OS with the CS loadparm option.
- Copy GBLRESOL, TCPDATA, and PROF2 from ADCD.Z24A.TCPPARMS to USER.Z24A.TCPPARMS.
- Copy FTPD, NFSC, NFSS, PORTMAP, RESOLVER, and TCPIP from ADCD.Z24A.PROCLIB to USER.Z24A.PROCLIB.
- Edit FTPD, NFSS, NFSC, PORTMAP, and RESOLVER. After you make changes, verify whether the output is displayed as shown below.

```
USER.PROCLIB(FTPD)
//*SYSTCPD DD DISP=SHR,DSN=TCPIP.SEZAINST(TCPDATA)
//*SYSTCPD DD DISP=SHR,DSN=ADCD.&SYSVER..TCPPARMS(TCPDATA)
//SYSTCPD DD DISP=SHR,DSN=USER.TCPPARMS(TCPDATA)
USER.PROCLIB(NFSC)
//*SYSTCPD DD DISP=SHR,DSN=ADCD.&SYSVER..TCPPARMS(TCPDATA)
//SYSTCPD DD DISP=SHR,DSN=USER.TCPPARMS(TCPDATA)
USER.PROCLIB(NFSS)
//*SYSTCPD DD DISP=SHR,DSN=ADCD.&SYSVER..TCPPARMS(TCPDATA)
//SYSTCPD DD DISP=SHR,DSN=USER.TCPPARMS(TCPDATA)
USER.PROCLIB(PORTMAP)
//*SYSTCPD DD DSN=ADCD.&SYSVER..TCPPARMS(TCPDATA),DISP=SHR
//SYSTCPD DD DSN=USER.TCPPARMS(TCPDATA),DISP=SHR

USER.PROCLIB(RESOLVER)
//SETUP DD DISP=SHR,DSN=ADCD.&SYSVER..TCPPARMS(GBLRESOL),FREE=CLOSE
//SETUP DD DISP=SHR,DSN=USER.TCPPARMS(RESOLVER),FREE=CLOSE
//*
USER.PROCLIB(TCPIP)
//*
//PROFILE DD DISP=SHR,DSN=USER.TCPPARMS(PROFILE)
//*PROFILE DD DISP=SHR,DSN=FEU.&SYSVER..TCPPARMS(PROFILE)
//*PROFILE DD DISP=SHR,DSN=ADCD.&SYSVER..TCPPARMS(PROF1)
//*PROFILE DD DISP=SHR,DSN=ADCD.&SYSVER..TCPPARMS(PROF2)
//*PROFILE DD DISP=SHR,DSN=TCPIP.PROFILE.TCPIP

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

- Create the member IPNODES in USER.Z24A.TCPPARMS.

```
000001 X.XX.XXX.XXX ZOS-DEV4 ZOS-DEV4.ibm.com
000002 10.1.1.1 LINUX
000003 127.0.0.1 LOCALHOST
```

Replace X.XX.XXX.XXX with the static IP.

Replace ZOS-DEV4 with the host name that is used in step 2.

Replace ZOS-DEV4.ibm.com with the fully qualified host name.

9. Rename GBLRESOL to RESOLVER in USER.Z24A.TCPPARMS, and edit it as shown below.

```
***** ***** Top of Data *****
000001 DEFAULTTCPIPDATA('USER.TCPPARMS(TCPDATA)')
000002 GLOBALTCPIPDATA('USER.TCPPARMS(TCPDATA)')
000003 GLOBALIPNODES('USER.TCPPARMS(IPNODES)')
000004 DEFAULTIPNODES('USER.TCPPARMS(IPNODES)')
000005 COMMONSEARCH
000006 CACHE
000007 CACHESIZE(200M)
000008 MAXTTL(2147483647)
000009 UNRESPONSIVETHRESHOLD(25)
***** ***** Bottom of Data *****
```

10. Edit USER.Z24A.TCPPARMS(TCPDATA).

```
000072 ; No prefix is required if the TCPIP.DATA file is not being shared.
000073 ;
000074 S0W1:  HOSTNAME  XXXX
000075 ;
Replace XXXX with Hostname used in Step2

000085 ; host name.
000086 ;
000087 DOMAINORIGIN  XXXX
000088 ;
000089 ;
```

Replace XXXX with a domain name, for example, `ibm.com`. Do not append a host name with the domain name, for example, `ZDT-DEV4.ibm.com`.

```
000143 NSINTERADDR  Primary DNS
000144 NSINTERADDR  Secondary DNS
```

Repeat if a single NSINTERADDR is present in the existing source, and provide the primary and secondary DNS. You can find the primary and secondary DNS in the DEVMAP as shown below, or ask your network administrator for the DNS.

```
# ; Define name servers with NSINTERADDR x.x.x.x
# NSINTERADDR      9.26.32.5
# NSINTERADDR      9.26.33.5
# RESOLVEVIA       UDP
```

11. Rename PROF2 in USER.Z24A.TCPPARMS to PROFILE, and edit it.

```
000070 ;This device defines the tunnel
000071 DEVICE PORTA MPCIPA
000072 LINK TAP0 IPAQENET PORTA
000073 HOME 10.1.1.2 TAP0
000074 ; This second device is optional
000075 ;DEVICE PORTB MPCIPA
000076 ;LINK ETH0 IPAQENET PORTB
000077 ;HOME XX.XX.XX.X ETH0
000078 BEGINRoutes
000079 ; Destination SubnetMask FirstHop LinkName Size
000080 ROUTE 10.1.1.1 255.255.255.0 = TAP0 MTU 1492
000081 ROUTE XX.XX.XX.0 255.255.254.0 = ETH0 MTU 1492
000082 ROUTE DEFAULT XX.XX.XX.1 ETH0 MTU 1492
000083 ENDRoutes
```

Replace `XX.XX.XX` with the static IP, and check which protocol is running. If `eth0` is running, your devmap has a correct structure of PROFILE that you can copy and paste. Otherwise, you need to change the structure manually. To check which protocol is running, use `FIND_IO` to check the subnet mask and destination, and `IFCONFIG` to find the subnet mask and destination in Linux.

For example, run `FIND_IO` for `ibmsys1@csx101.XX.ibm.com` to check whether `TAP0` and `ETH0` are running on a machine.

IPv4	Interface	Current	MAC

Path Address	Name	State	Address	IPv6 Address
F0 9.28.160.193	eth0	UP, RUNNING	fa:16:3e:2f:c9:9c	fe80::f816:3eff:fe2f:c99c%eth0
* 192.168.122.1	virbr0	UP, NOT-RUNNING	52:54:00:77:c7:21	
* *	virbr0-nic	DOWN	52:54:00:77:c7:21	
A0 10.1.1.1	tap0	UP, RUNNING	12:3e:b7:80:ad:d1	fe80::103e:b7ff:fe80:ad1%tap0
* A1	tap1	DOWN	02:a1:a1:a1:a1:a1	
* A2	tap2	DOWN	02:a2:a2:a2:a2:a2	
* A3	tap3	DOWN	02:a3:a3:a3:a3:a3	
* A4	tap4	DOWN	02:a4:a4:a4:a4:a4	
* A5	tap5	DOWN	02:a5:a5:a5:a5:a5	
* A6	tap6	DOWN	02:a6:a6:a6:a6:a6	
* A7	tap7	DOWN	02:a7:a7:a7:a7:a7	

End of FIND_IO

12. Re-IPL the machine, or restart the TCP/IP and resolver by using console commands:

```
P TCPIP
P RESOLVER
S TCPIP
S RESOLVER, SUB=MSTR
```

13. Verify whether the static IP works.

- Open another terminal emulator, and connect with your static IP and port 23. Then, z/OS login screen is displayed.
- On the **Command** pane of ISPF, enter ping <ip of other machine>. Other machine responds.
- Enter the master console command D U, , , 404, 3. Four-zero-four status is turned into active (A-BSY).
- Enter the following console command, and OSA should be active.

```
v net, id=osatrl2, act
```

Updating Personal Edition

If you installed ZD&T V11.0 or later versions, you can update Personal Edition directly to a new version. If you installed ZD&T V10.0.1 or earlier versions, you need to uninstall the earlier version before you install the current Personal Edition.

For the detailed steps to update Personal Edition, see [“Checklist” on page 9](#).

Uninstalling the earlier versions of ZD&T

If you installed zD&T V10.0.1 or earlier versions, use the steps below to uninstall the previous versions.

To uninstall any supported version, complete the following steps.

Version 9.0 or above	<ul style="list-style-type: none">• Stop any currently active instance of RD&T.<ul style="list-style-type: none">– To stop a base instance of RD&T, see “Stopping Z instances ” on page 22.– If you are running a product license server:<ol style="list-style-type: none">1. Stop the UIM server by using the command from a non-root user ID.<pre data-bbox="951 531 1471 604">cd /usr/z1090/bin ./uimserverstop</pre>2. Enter these two commands to stop the product license server:<pre data-bbox="951 695 1471 814">cd /opt/safenet_sentinel/common_files/ sentinel_keys_server ./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 x 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 x is the version that is installed. • Go to the subdirectory entitled Uninstall_Rational Developer for z Unit Test V8.0.x. • Run the command ./"Uninstall_Rational Developer for z Unit Test V8.0.x". 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".
-------------	--

After the uninstallation completes, you can install the new version of ZD&T by completing the steps that are described in ["Installing Personal Edition"](#) on page 16.

Note: ZD&T Personal Edition doesn't support older version of ZD&T. So, to use ZD&T Personal Edition, you need to uninstall older versions of zD&T or RD&T, and install ZD&T V11.0.

Uninstalling older versions of zD&T or RD&T 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 ZD&T 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.

Updating Personal Edition to the latest version

Learn about the steps to update Personal Edition to the latest version interactively or silently.

Before you update ZD&T Personal Edition, make sure that the `awsstop` command was issued from a non-root user ID.

To update ZD&T Personal Edition, run the following steps from root user ID.

1. Open the directory that contains the installer `ZDT_Install_PE_V12.0.5.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PE_V12.0.5.0.tgz
```

4. Optional: Read the `README.txt` file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PE_V12.0.5.0.x86_64
```

6. Type **2** to update.

7. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
8. Optional: enter y to install automatic network configuration. Or, enter n to decline the installation.

By entering y, your z/OS starts 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. Before you enter y, you need to consider that the network configuration changes your Linux and networking functions, allows the Linux firewall to reroute TCP/IP packets to z/OS, and updates the TCP/IP specifications in the z/OS disks.

To learn more about the network configuration, see [“Configuring the network”](#) on page 23.

Important: Remember to download the volumes of ADCD November 2019 edition from passport advantage, and decompress the *.gz volume on the same machine where you install ZD&T V12.0.5.

9. After the update completes, run the following command to verify whether the update is successful.

For the RHEL operating system, run the command.

```
rpm -qa | grep z1091
```

For the UBUNTU operating system, run the command.

```
dpkg -l | grep z1091
```

If the package is updated successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Alternatively, after you complete steps 1- 3, you can run the following commands to update ZD&T Personal Edition silently.

- To update Personal Edition without automated network configuration, run the following command:

```
./ZDT_Install_PE_V12.0.5.0.x86_64 --update
```

- To update Personal Edition with automated network configuration, run the following command:

```
./ZDT_Install_PE_V12.0.5.0.x86_64 --update --net
```

After the update completes, you can follow the steps that are described in [“Enabling a license key”](#) on page 18 and [“Configuring Z instances”](#) on page 20.

Note: This process of update is only valid for ZD&T version 11 or above. For an earlier ZD&T or RD&T version, you need to follow the corresponding documentation of the version.

Uninstalling Personal Edition

Learn about the steps to uninstall Personal Edition interactively or silently.

Before you uninstall ZD&T Personal Edition, you must return an existing license key first. For more information, see [“Returning an existing license key”](#) on page 20.

To uninstall ZD&T Personal Edition, run the following steps from root user ID.

1. Stop any running Z instance. For detailed steps, see [“Stopping Z instances ”](#) on page 22.
2. Enter the following command from the root user ID.

```
cd /opt/safenet_sentinel/common_files/sentinel_keys_server  
./loadserv stop
```

3. Run the installer.

```
./ZDT_Install_PE_V12.0.5.0.x86_64
```

4. Type **3** to uninstall.

Alternatively, after you complete steps 1 - 3, you can run the following command to uninstall ZD&T Personal Edition silently.

```
ZDT_Install_PE_V12.0.5.0.x86_64 --uninstall
```

Note: This process of uninstallation is only valid for ZD&T version 11.0 or above. If you need to uninstall ZD&T v10.0.1 or earlier versions, see [“Uninstalling the earlier versions of ZD&T ” on page 27.](#)

Enterprise Edition

ZD&T Enterprise Edition enables enterprises to host an IBM Z distribution on low-cost Intel-based x86 machines. It creates an environment for mainframe application demonstration, development, testing, and education without Z mainframe hardware, enables z/OS, middleware, and other z/OS software to run on Intel and other compatible computers, and emulates z/Architecture with virtual I/O and devices. Enterprise Edition also provides a web-based interface. You can extract, deploy, and manage the application images from an existing Z or ADCD packages.

With a flexible licensing method, ZD&T Enterprise Edition can be used on cloud, VMs, or in-housed physical 8086 hardware. The Enterprise Edition also comes with a single user license that is known as Authorized User (AU) license, or with a multi-user license that is known as the Resource value Unit (RVU) license. If you use a Rational token license server to enable other IBM products, you can use the same licensing method for ZD&T.

ZD&T Enterprise Edition provides the following features.

- An application development and testing environment that can improve development infrastructure availability and flexibility
- Current levels of IBM z/OS software that give access to new runtime capabilities for development and testing for enterprises
- Mixed workload support for enterprises, which can help reduce the development costs
- An approachable and portable environment for education on Z for enterprises
- A web-based interface to extract, deploy and manage the application images from an existing Z or ADCD packages.
- Creating and managing 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 environments
- Integrating with IBM Application Discovery (AD) to deploy the data sets that are identified by AD to the target machine (ZD&T)

ZD&T Enterprise Edition web server

ZD&T Enterprise Edition web server enables users to use the browser to extract volumes or data sets, transfer the volumes or data sets to the image storage server, and deploy them to the target environment. You can use the web server to deploy the volumes or data sets to the target ZD&T machine step-by-step.

The Enterprise Edition web server provides the following features:

- Integrating with IBM Application Discovery (AD) to transfer the data sets that are identified in AD projects
- Integrating with company LDAP account to enable administrators to set up accounts with minimum efforts
- Scheduling the extraction or deployment for another date or time that you want

- Flexible functions to select required data sets or volumes from the source machine, and transfer the data sets or volumes to the target machine
- Providing Rest API support, which enables you to integrate and automate the functions of the web server in your own environment
- The deployment process can install ZD&T emulator, and make the system ready. You do not need to remember the ZD&T native commands.
- If IBM Application Discovery (AD) is integrated with a source Z, AD can be used to provision the CICS environment.
- Integrating with OpenStack cloud.

Source Z, ZD&T or ADCD

ZD&T Enterprise Edition can work with your existing Z machine, ZD&T machine or ADCD to extract and deploy the necessary volumes or data sets that are required for the target machine.

Image storage server

ZD&T Enterprise Edition stores extracted information on the intermediary storage machine, for example, FTP or SFTP server. Extracted information is never deleted from the image storage server until the information is manually deleted, which enables you to provision as many as machines that you want based on your license entitlement. You need to have enough storage on the image storage server to hold the required volumes or data sets.

Note: ZD&T Enterprise Edition web server, image storage server, and License Server (Software-based License Server or Hardware-based License Server) can be configured on the same machine. If you want to use one machine, you need to have large storage on the machine.

Software-based License Server

When you use the software-based licensing, the Software-based License Server provides a server for centralized management of license keys for one or more instances of ZD&T Enterprise Edition. With this method, a USB hardware device is not required.

To prepare the Software-based License Server, install the server and authenticated license key files. The licensee is not authorized to use Enterprise Edition or any of its components except when the Software-based License Server is activated with a license key file and is accessible by the program, as it provides the proof of license entitlement.

Each license key file is generated with a number of entitled emulated central processors, which is referred as Central Processors (CPs). A single instance of ZD&T 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 more information about the number of CPs that is used with an instance, see section 2.3.4 "Performance" in the [zPDT Guide and Reference](#).

The Enterprise Edition also comes with a single user license that is known as Authorized User Single Session (AUSS), or with a multi-user license that is known as the Resource Value Unit (RVU).

The following topologies describe the Software-based License Server and various components of Enterprise Edition.

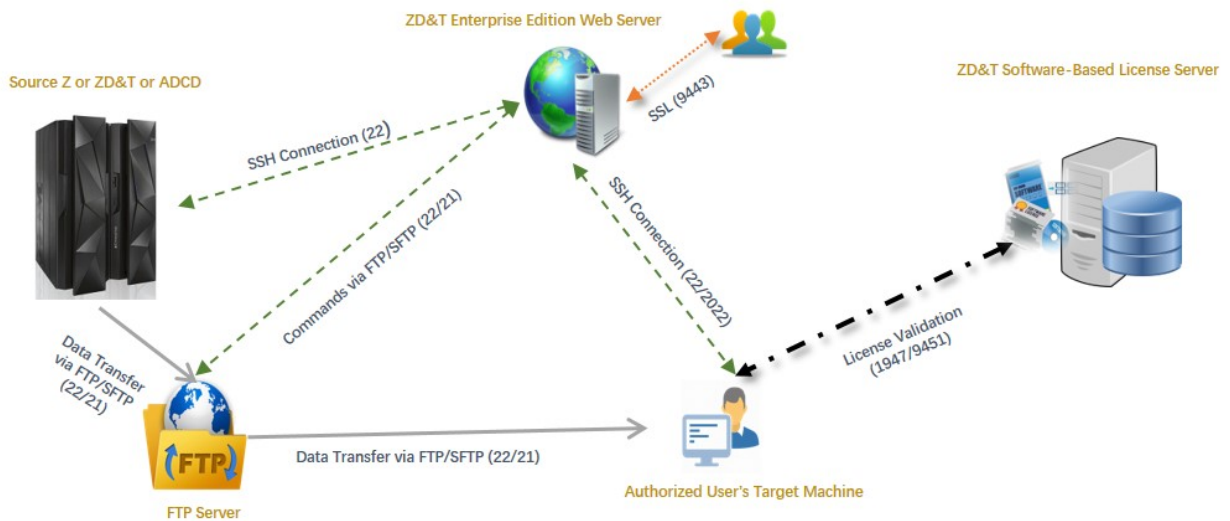


Figure 3. The AUSS and various components of Enterprise Edition

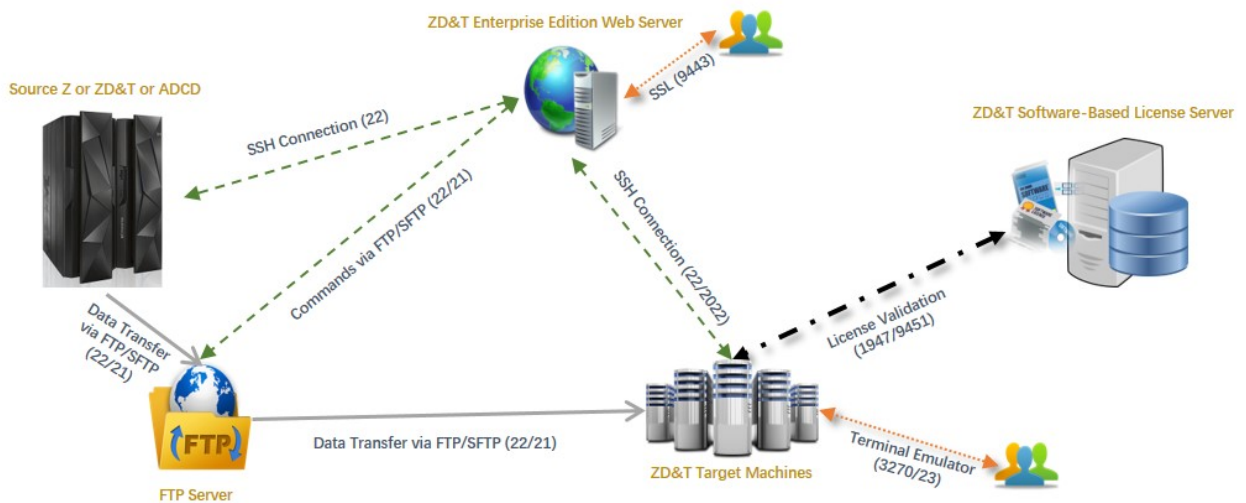


Figure 4. The RVU and various components of Enterprise Edition

Hardware-based License Server

The earlier versions of ZD&T Enterprise Edition require a USB hardware device that contains a license key to control 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. After you order a license key (called update file) from Rational License Key Center, you need to apply the license key to the USB hardware device. Then, connect the USB hardware device to the machine that is known as Hardware-based License Server.

Each update file is generated with a number of entitled emulated central processors, which is referred as Central Processors (CPs). A single instance of ZD&T requires at least 1 CP, and can run with a maximum of eight CPs. The number of CPs that is needed depends on the number of users and the types and amount of processing required. For more information about the number of CPs that is used with an instance, see section 2.3.4 "Performance" in the [zPDT Guide and Reference](#).

The following topology describes the Hardware-based License Server and various components of Enterprise Edition.

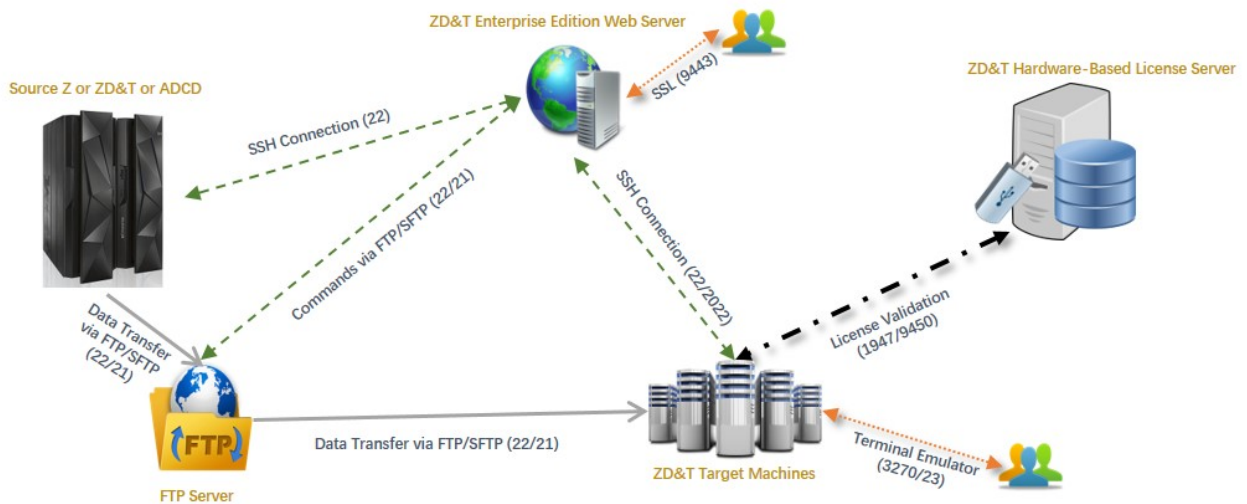


Figure 5. The Hardware-based License Server and various components of Enterprise Edition

Rational Tokens

Rational Software License Server (RLKS) uses Rational Tokens to enable IBM supplied products. If you also use other IBM products, and already set up RLKS, you can use Rational Tokens to enable ZD&T Enterprise Edition.

To enable Enterprise Edition by Rational Tokens, you need to decide to use Hardware-based License Server or Software-based License Server. In either case, the license key file indicates that Rational Tokens are required.

For ZD&T Enterprise Edition, each emulated central processor within an instance of a Z virtual machine requires a particular number of Rational Tokens, based on the token value for ZD&T Enterprise Edition. 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.

The use of Rational Tokens does not replace the requirement for a license key for ZD&T Enterprise Edition. Either a software-based license key file or a USB hardware device with a valid update file is still required.

The following topology describes the Software-based License Server and Software-based License Server with Rational Tokens and various components of Enterprise Edition.

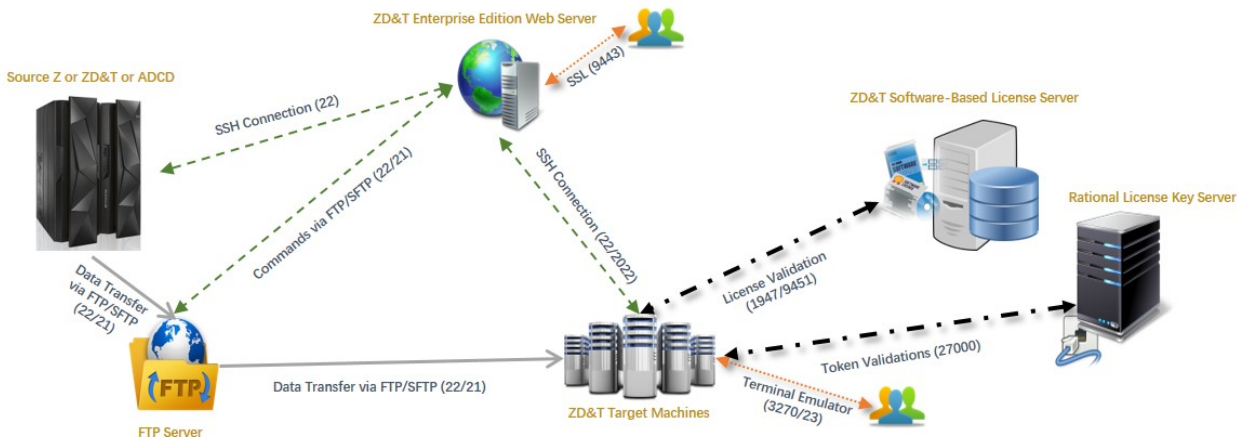


Figure 6. The Software-based License Server with Rational Tokens, and various components of Enterprise Edition

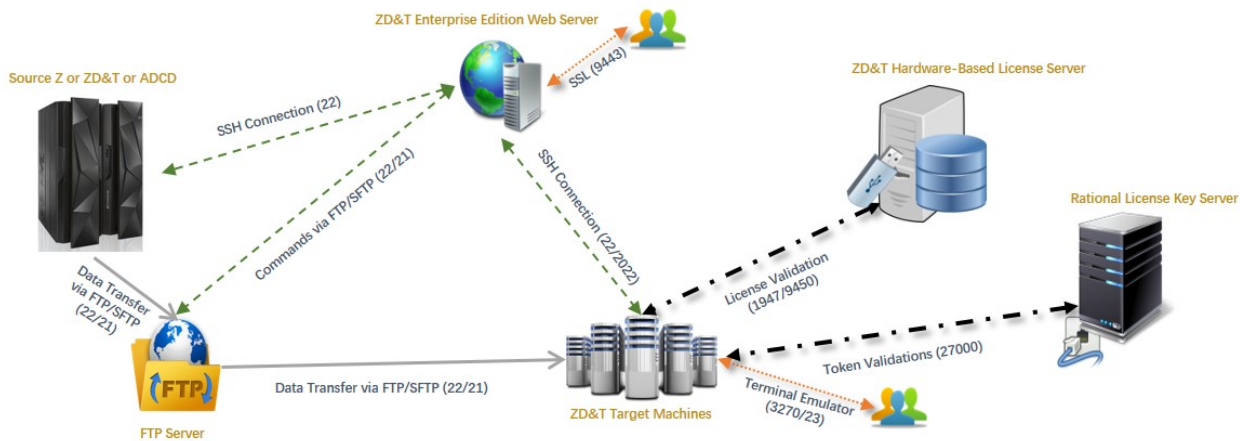


Figure 7. The Hardware-based License Server with Rational Tokens, and various components of Enterprise Edition

Application Developers Controlled Distribution (ADCD)

ZD&T Enterprise Edition also comes with the Application Developers Controlled Distribution (ADCD) package, which is known as z/OS ADCD. ADCD package enables the enterprise users to use and develop their applications with the newest set of mainframe software packages, such as CICS, Db2, IMS, z/OS 2.3, and JES2. You can select the software packages that you need, and deploy the packages out of ADCD to the target ZD&T images. Alternatively, you can use Z images from source Z.

For a complete list of z/OS products in ADCD, see “ADCD z/OS V2R4 November Edition of 2019” on page 178.

Terminal emulator

A terminal emulator, terminal application, or other terminologies that emulate a video terminal within some other display architecture. Though, the terminal emulator is typically synonymous with a shell or text terminal, the terminal emulator covers all remote terminals, including graphical interfaces. A graphical user interface that is in the terminal emulator is known as a terminal window.

Various terminal emulators are available in the market, and this software is not supplied with ZD&T. You can download the terminal emulator that you want. Or you can purchase IBM Personal Communication that is known as (IBM PCOMM), and download x3270 on your Linux machine if you want. Terminal emulator screen is accessed via port 3270/23.

Checklist

If you want to install, update, or uninstall Enterprise Edition, or if you want to update the license server or license, you can follow the steps that are provided in the checklist.

Table 6. New installation checklist for Enterprise Edition			
Installing Enterprise Edition		Required/Optional	Complete
1	“Prerequisites” on page 37 Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	<input type="checkbox"/>
2	Obtaining installation packages Describes the steps to download all required packages to install Enterprise Edition.	Required	<input type="checkbox"/>

Table 6. New installation checklist for Enterprise Edition (continued)

Installing Enterprise Edition		Required/ Optional	Complete
3	<p>“Enabling the license servers” on page 44</p> <p>Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.</p>	Required	<input type="checkbox"/>
4	<p>“Installing Enterprise Edition” on page 61</p> <p>Describes the steps to install the Enterprise Edition.</p>	Required	<input type="checkbox"/>
5	<p>“Optional: Installing the z/OS files” on page 63</p> <p>Describes the steps to install z/OS files if you need to create application images from IBM Z components.</p>	Optional	<input type="checkbox"/>
6	<p>Setting up the Enterprise Edition web server environment</p> <p>Describes the steps to configure the Enterprise Edition server environment before you start ZD&T Enterprise Edition.</p>	Required	<input type="checkbox"/>
7	<p>“Starting and accessing the Enterprise Edition web server” on page 64</p> <p>Describes the steps to start the Enterprise Edition web server.</p>	Required	<input type="checkbox"/>
8	<p>“User's Guide” on page 65</p> <ul style="list-style-type: none"> • “Enterprise Edition User's Guide” on page 65 • “REST APIs User's Guide” on page 97. <p>Describes the steps to create, monitor and deploy application images by using the Enterprise Edition web user interface or REST APIs.</p>	Required	<input type="checkbox"/>

Table 7. Update checklist for Enterprise Edition

Updating Enterprise Edition		Required/ Optional	Complete
1	<p>“Prerequisites” on page 37</p> <p>Describes the software and hardware requirements, and the steps to set up the basic Linux system.</p>	Required	<input type="checkbox"/>
2	<p>Obtaining installation packages</p> <p>Describes the steps to download all required packages to install Enterprise Edition.</p>	Required	<input type="checkbox"/>
3	<p>“Updating Enterprise Edition” on page 124</p> <p>Describes the steps to update Enterprise Edition.</p>	Required	<input type="checkbox"/>
4	<p>Setting up the Enterprise Edition web server environment</p> <p>Describes the steps to configure the Enterprise Edition server environment before you start ZD&T Enterprise Edition.</p>	Required	<input type="checkbox"/>

Table 7. Update checklist for Enterprise Edition (continued)

Updating Enterprise Edition		Required/ Optional	Complete
5	<p>“Starting and accessing the Enterprise Edition web server” on page 64</p> <p>Describes the steps to start the Enterprise Edition web server.</p>	Required	<input type="checkbox"/>
6	<p>“User's Guide” on page 65</p> <ul style="list-style-type: none"> • “Enterprise Edition User's Guide” on page 65 • “REST APIs User's Guide” on page 97. <p>Describes the steps to create, monitor and deploy application images by using the Enterprise Edition web user interface or REST APIs.</p>	Required	<input type="checkbox"/>

Table 8. Update checklist for license server

Updating license server		Required/ Optional	Complete
1	<p>“Returning a software-based license key” on page 50 or “Returning a hardware-based license key” on page 58</p> <p>Describes the steps to return a software-based or hardware-based license or Rational Tokens.</p>	Required	<input type="checkbox"/>
2	<p>“Updating the Software-based License Server” on page 53 or “Updating the Hardware-based License Server” on page 60</p> <p>Describes the steps to update the Software-based License Server or Hardware-based License Server.</p>	Required	<input type="checkbox"/>
3	<p>“Enabling the license servers” on page 44</p> <p>Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.</p>	Required	<input type="checkbox"/>

Table 9. Update checklist for license

Updating license		Required/ Optional	Complete
1	<p>“Returning a software-based license key” on page 50 or “Returning a hardware-based license key” on page 58</p> <p>Describes the steps to return a software-based or hardware-based license or Rational Tokens.</p>	Required	<input type="checkbox"/>
2	<p>“Enabling the license servers” on page 44</p> <p>Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.</p>	Required	<input type="checkbox"/>

Table 10. Uninstallation checklist for Enterprise Edition

Uninstalling Enterprise Edition		Required/ Optional	Complete
1	“Returning a software-based license key” on page 50 or “Returning a hardware-based license key” on page 58 Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	“Uninstalling the Software-based License Server” on page 53 or “Uninstalling the Hardware-based License Server” on page 60 Describes the steps to update the Software-based License Server or Hardware-based License Server	Required	<input type="checkbox"/>
3	“Uninstalling Enterprise Edition” on page 124 Describes the steps to uninstall Enterprise Edition.	Required	<input type="checkbox"/>

Planning

Learn about the planning information about how to define appropriate system requirements, installation packages, and steps based on the edition you purchased.

Prerequisites

Learn about hardware and software requirements for ZD&T Enterprise Edition.

For a complete list of ZD&T hardware and software requirements, you can generate the report from [Software Product Compatibility Reports](#). Hardware and software requirements are also documented in the [zPDT Guide and Reference](#).

z/OS system requirements

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

Required

- An SSH server must be running and accessible by the system to run Enterprise Edition.
- The FTP or SFTP client must be able to connect to the Enterprise Edition image storage server.
- To use SFTP, Java™ 1.6 or later versions must be installed, and the PATH needs to be pointed to the bin directory of the Java installation.
- Make sure to grant access to each volume or data set that is extracted. For more information, see [Creating application components from IBM Z mainframe volumes](#) and [Creating application components from IBM Z mainframe data sets](#).
- Make sure to grant READ access to DFDSS program ADRDSSU.

Optional

- Configure zEnterprise® Data Compression (zEDC) if it is available. Grant READ access to the resource FPZ.ACCELERATOR.COMPRESSION in SAF class FACILITY to the user ID that is used in the Enterprise Edition.
- Grant READ access to resource STGADMIN.ADR.DUMP.CNCURRNT in SAF class FACILITY.

Image storage server requirements

To install and run ZD&T Enterprise Edition, an image storage server to host the Enterprise Edition artifacts, such as z system volumes, data sets, Enterprise Edition metadata, must be set up. To transfer volumes images files from the image storage server or to the image storage server, you can choose FTP or SFTP as the transferring method.

- **Disk space**

- Sufficient space is needed to hold numerous and potentially large files for extracted IBM Z volumes.
- 150 GB of disk space is needed for ADCD z/OS V2.3 distribution.

- **Software requirements**

- A running FTP or SFTP server

- **FTP server**

- Enable passive mode by configuring `PASV_ENABLED=YES` in the `vsftpd.conf`.
- Set SELinux to Permissive or Disabled mode.
- Open firewall ports for FTP command and data channels.

- **SFTP server**

- Open the firewall port for SFTP command.

Target environments requirements

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

- **Linux environments**

- If you choose to install the required Linux packages during the deployment, the software repository needs to be available and accessible by the target environments.
 - A Red Hat software repository for 'yum' needs to be available and accessible by the target environments.
 - An Ubuntu software repository for 'apt-get' need to 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 Enterprise Edition.
- The command `ftp client` must reside on the target environments.
- 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 environment, as the `loadparm.txt` that is generated for a script to modify z/OS parameters might cause space problem.

Note: ZD&T Hardware-Based and Software-Based License Server installation options open the ports 1947, 3270, 9451 and 9450 to communicate between License server and Terminal emulator. You need to remove any restriction that might prevent the installer from opening the ports cannot be on your Linux machine. To deactivate any firewall that is active, or ask your network administrator to deactivate the firewall. For more information about the opened ports, see [“Firewall ports list” on page 69](#).

- **OpenStack Cloud environments**

- Make sure that the default security group of the OpenStack Cloud contains the rules that enable all the required ports of a target environment. For more information, see [“Firewall ports list” on page 69](#).

- **Access to software repository to run YUM or apt-get commands**

Make sure that you have access to software repository to run YUM commands on RHEL machine, or run `apt-get` commands on Ubuntu machine. ZD&T installer will install all required packages. However, if

you don't want ZD&T installer to install the required packages that are listed below, you need to install the packages before you start ZD&T installer.

– YUM commands on RHEL operating system

```
yum -y install iptables
yum -y install ftp
yum -y install libstdc++.i686
yum -y install perl
yum -y install zip
yum -y install unzip
yum -y install gzip
yum -y install bc
```

– apt-get commands on UBUNTU operating system

```
apt-get -y install iptables
dpkg --add-architecture i386
apt-get -y update
apt-get -y install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 lib32stdc++6
apt-get -y install ftp
apt-get -y install perl
apt-get -y install zip
apt-get -y install unzip
apt-get -y install gzip
apt-get -y install bc
apt-get -y install libasound2
apt-get -f install
```

Note: The 'nc' command is not available by default on the RHEL 7.4 and 7.5. As the 'nc' command is required to pass the validation before you start a deployment to the target environment, the missing command 'nc' might cause the failure of the connection. To install the command, run the following command.

```
yum -y install nc
```

• Users and group settings

Before you deploy application images, make sure to create a new group 'zpd' in the target environment if the group does not exist.

- If you use the root user ID to deploy application images, create a user ID 'ibmsys1' if the user ID does not exist, and assign the user ID 'ibmsys1' to the group 'zpd'.
- If you use a non-root user ID to deploy application images, assign the user ID to the group 'zpd'.

• Sudo access configuration

The term *sudo* stands for super user do. *Sudoers* is the configuration file with the corresponding operating system sudo settings. This file is typically at /etc/sudoers. For more information about the specific sudoers file format, see [Sudoers Manual](#).

The following code shows an example of a sudoers entry.

```
ibmsys1 ALL = (root) NOPASSWD: ALL
```

In the code example, user `ibmsys1` can access any shell script file on any host in any location as the root user without providing an identification password.

During the deployment, Enterprise Edition runs several scripts that require the root access. For the security reasons, Enterprise Edition also changes the ownership of the scripts to the root user ID. The user ID that is used for the deployment needs to have the permission to run the scripts and change the ownership of the scripts. The scripts list is shown as below.

```
[deployment directory]/zdt/zdtInstall/zdt_install_product_byRoot.sh
[deployment directory]/zdt/zdtInstall/zdt_modify_files_byRoot.sh
[deployment directory]/zdt/zdtInstall/zdt_install_dependencies_byRoot.sh (optional)
[deployment directory]/zdt/zdtInstall/zdt_config_user_byRoot.sh (optional)
[deployment directory]/zdt/zdtInstall/zdt_config_network_byRoot.sh (optional)
[deployment directory]/zdt/zdtInstall/zdt_cleanup_byRoot.sh (optional)
```

The deployment directory is an optional input value that can be specified from web user interface or REST API. By default, the deployment directory is /home/ibmsys1 if you log in as the root user, and /home/[userid] if you log in as a non-root user.

The following code shows an example of the sudoers entry. The user ID that is used is ibmsys1, and the deployment directory is /home/ibmsys1.

```
ibmsys1 ALL=(root) NOPASSWD: /bin/chown root /home/ibmsys1/zdt/zdtInstall/  
zdt_modify_files_byRoot.sh,  
/home/ibmsys1/zdt/zdtInstall/zdt_modify_files_byRoot.sh  
/home/ibmsys1/zdt/zdtInstall/zdt_install_dependencies_byRoot.sh,  
/home/ibmsys1/zdt/zdtInstall/zdt_config_user_byRoot.sh,  
/home/ibmsys1/zdt/zdtInstall/zdt_install_product_byRoot.sh,  
/home/ibmsys1/zdt/zdtInstall/zdt_config_network_byRoot.sh,  
/home/ibmsys1/zdt/zdtInstall/zdt_cleanup_byRoot.sh
```

If you use other privilege management tools other than sudo, you also need to do the configuration.

Network configuration

To make other systems communicate with your emulated z/OS, you need to configure the emulated environment with an IP of 10.1.1.2 to ensure that the emulated environment can be accessible. A user is always able to configure the routing, and the only requirement is to route the port 2022 on the Linux to port 22 on the emulated z/OS.

To configure the network, complete the following steps:

1. Back up the current iptables rules.
2. Run the following commands.

```
iptables --flush  
iptables --table nat --flush  
iptables --table nat --append POSTROUTING --out-interface eth1 -j MASQUERADE  
iptables --table filter --append FORWARD --in-interface tap0 -j ACCEPT  
iptables --table filter -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p tcp --dport 0:21 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 0:21 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p tcp --dport 23:2021 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 23:2021 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p tcp --dport 2022 -j DNAT --to 10.1.1.2:22  
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 2022 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p tcp --dport 2023:3269 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 2023:3269 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p tcp --dport 3271:9449 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 3271:9449 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p tcp --dport 9452:65535 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 9452:65535 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p udp --dport 111 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p udp -d 10.1.1.2 --dport 111 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p udp --dport 514 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p udp -d 10.1.1.2 --dport 514 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p udp --dport 1023 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p udp -d 10.1.1.2 --dport 1023 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p udp --dport 1044:1049 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p udp -d 10.1.1.2 --dport 1044:1049 -j ACCEPT  
iptables -A PREROUTING --table nat -i eth1 -p udp --dport 2049 -j DNAT --to 10.1.1.2  
iptables -A FORWARD -p udp -d 10.1.1.2 --dport 2049 -j ACCEPT  
echo 1 > /proc/sys/net/ipv4/ip_forward
```

eth1 is an example of the network interface name. To find available network interfaces, run the command `ifconfig`, `ip -o address show`, and so on.

Db2 extraction

Db2® table extraction uses the following standard Db2 utilities and functions.

- Db2 REXX Language Support (DSNREXX).
- Stored Procedure DSNWZP for using the Db2 Admin Tool.
- Stored Procedure DSNUTILU for running Db2 online utilities.

To extract data from a Db2 table, you must ensure that all these utilities are available on the source system. DSNUTILU and DSNWZP also require z/OS Workload Manager (WLM) application environments. To define these utilities, you can use the following Db2 installation jobs that are typically run during the Db2 installation.

DSNTIJTM

This job can be used to bind DSNREXX.

DSNTIJRT

This job can be used to define the Db2 routines DSNUTILU and DSNWZP.

DSNTIJRW

This job can be used to define and optionally activate Workload Manager application environments that are needed for DSNUTILU, DSNWZP, and other Db2 WLM environments.

• **User Access**

To use the user ID that is specified on the source system to extract Db2 data, you must ensure that the user ID has the following access.

- Read access to the Db2 catalog tables.
- Read access to the tables that are selected for an extraction.
- Unload access to the tables that are selected for an extraction.
- Authority to stop Db2 UNLOAD utilities.
- If you need to use the Db2 Admin Tool, the user ID that runs the extraction must have the Db2 or RACF access to run the DDL Generation Plan, for example, ADB2GEN.

• **System Libraries**

REXX.SEAGALT or REXX.SEAGLPA must be in the system search order, that is, Linklist or LPA.

• **Db2 Admin Tool**

To obtain the source database DDL, the Db2 Admin Tool must be installed and available. If the Db2 Admin Tool is not available, you must supply and verify all DDL source. The database DDL that is created on the target system must be compatible to Db2 supplied sample DSNTEP2.

Installation packages

Learn about how to download installation package that is required for ZD&T Enterprise Edition.

To download the package, complete the following steps:

1. Log on to [Passport Advantage](#).
2. Select **Software Downloads** and **Media Access**.
3. Select **Program offering and agreement number**, and click **Continue**.
4. Enter the part description or part number, and click **Finder**.
5. Optionally, you can click the alphabetical order list to display and view the product by name.
6. Select **All Operating Systems** in the **Operating system** field, and **All Languages** in the **Languages** field. Then, click **Go**.
7. If you want to download the individual media from the list, click **Select individual files** to expand the list.
8. Verify the e-assemblies that you want to download with the list that is mentioned below in the table.

Note: Verify the integrity of downloaded ADCD packages by using the MD5SUM that is in the adcd.md5 and ee.md5 files.

	Name	Part No.	Package Type	Name
1	IBM Z Development and Test Environment Enterprise Edition 12.0.5 Installation Multilingual	CC5ASML	ZD&T Enterprise Edition installer	ZDT_Install_EE_12.0.5.0.tgz

	Name	Part No.	Package Type	Name
2	IBM Z Development and Test Environment Enterprise Edition 12.0.5 Installer checksum Multilingual	CC5ATML	checksum (md5 checksum)	ee.md5
3	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	CC59MML	ADCD November 2019 Edition RSU level 1909	A4RES1.ZPD
4	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	CC59NML	ADCD November 2019 Edition RSU level 1909	A4RES2.gz
5	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 3 of 19 - System volume Multilingual	CC59PML	ADCD November 2019 Edition RSU level 1909	A4SYS1.gz
6	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	CC59QML	ADCD November 2019 Edition RSU level 1909	A4CFG1.gz
7	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	CC59RML	ADCD November 2019 Edition RSU level 1909	A4USS1.gz
8	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	CC59SML	ADCD November 2019 Edition RSU level 1909	A4USS2.gz
9	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	CC59TML	ADCD November 2019 Edition RSU level 1909	A4PRD1.gz
10	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	CC59UML	ADCD November 2019 Edition RSU level 1909	A4PRD2.gz
11	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	CC59VML	ADCD November 2019 Edition RSU level 1909	A4PRD3.gz
12	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	CC59WML	ADCD November 2019 Edition RSU level 1909	A4PRD4.gz
13	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	CC59XML	ADCD November 2019 Edition RSU level 1909	A4DIS1.gz
14	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	CC59YML	ADCD November 2019 Edition RSU level 1909	A4DIS2.gz

	Name	Part No.	Package Type	Name
15	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	CC59ZML	ADCD November 2019 Edition RSU level 1909	A4DIS3.gz
16	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	CC5A0ML	ADCD November 2019 Edition RSU level 1909	A4PAGA.gz
17	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	CC5A1ML	ADCD November 2019 Edition RSU level 1909	A4PAGB.gz
18	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	CC5A2ML	ADCD November 2019 Edition RSU level 1909	A4PAGC.gz
19	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	CC5A3ML	ADCD November 2019 Edition RSU level 1909	A4USR1.gz
20	IBM Z Development and Test Environment 12.0.5 with z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	CC5A4ML	ADCD November 2019 Edition RSU level 1909	SARES1.ZPD
21	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 19 of 19 – zCX Container volume	CC5A5ML	ADCD November 2019 Edition RSU level 1909	A4ZCX1.gz
22	IBM Z Development and Test Environment 12.0.5 with Rational Team Concert volume 6.0.6 Multilingual	CC5A6ML	ADCD November 2019 Edition RSU level 1909	A4BLZ1.gz
23	IBM Z Development and Test Environment 12.0.5 with DB2 12 Part 1 of 2 Multilingual	CC5ABML	ADCD November 2019 Edition RSU level 1909	A4DBC1.gz
24	IBM Z Development and Test Environment 12.0.5 with DB2 12 Part 2 of 2 Multilingual	CC5ACML	ADCD November 2019 Edition RSU level 1909	A4DBC2.gz
25	IBM Z Development and Test Environment 12.0.5 with DB2 V11 Part 1 of 2 Multilingual	CC5ADML	ADCD November 2019 Edition RSU level 1909	A4DBB1.gz
26	IBM Z Development and Test Environment 12.0.5 with DB2 V11 Part 2 of 2 Multilingual	CC5AEML	ADCD November 2019 Edition RSU level 1909	A4DBB2.gz
27	IBM Z Development and Test Environment 12.0.5 with CICS 5.4 Multilingual	CC5AFML	ADCD November 2019 Edition RSU level 1909	A4C541.gz
28	IBM Z Development and Test Environment 12.0.5 with CICS 5.5 Multilingual	CC5AGML	ADCD November 2019 Edition RSU level 1909	A4C551.gz

	Name	Part No.	Package Type	Name
29	IBM Z Development and Test Environment 12.0.5 with IMS 14.1 Multilingual	CC5AHML	ADCD November 2019 Edition RSU level 1909	A4IME1.gz
30	IBM Z Development and Test Environment 12.0.5 with IMS 15.1 Multilingual	CC5AIML	ADCD November 2019 Edition RSU level 1909	A4IMF1.gz
31	IBM Z Development and Test Environment 12.0.5 with IBM Installation Manager 1.8.9 Multilingual	CC5AJML	ADCD November 2019 Edition RSU level 1909	A4INM1.gz
32	IBM Z Development and Test Environment 12.0.5 with DB2 Archive Logs Multilingual	CC5AKML	ADCD November 2019 Edition RSU level 1909	A4DBAR.gz
33	IBM Z Development and Test Environment 12.0.5 with Tivoli Omegamon XE products 5.5.1 Multilingual	CC5ALML	ADCD November 2019 Edition RSU level 1909	A4KAN1.gz
34	IBM Z Development and Test Environment 12.0.5 with WAS V9.0 Part 1 of 2 Multilingual	CC5AMML	ADCD November 2019 Edition RSU level 1909	A4W901.gz
35	IBM Z Development and Test Environment 12.0.5 with WAS V9.0 Part 2 of 2 Multilingual	CC5ANML	ADCD November 2019 Edition RSU level 1909	A4W902.gz
36	IBM Z Development and Test Environment V12.0.5 Rocket Open Source Languages and Tools for z/OS	CC5APML	ADCD November 2019 Edition RSU level 1909	ZDTRKT.gz
37	IBM Z Development and Test Environment 12.0.5 with ADCD Metadata for Tools Multilingual	CC5AQML	ADCD November 2019 Edition RSU level 1909	ADCDTOOLS.XML
38	IBM Z Development and Test Environment 12.0.5 ADCD Checksum Multilingual	CC5ARML	ADCD package MD5 checksum	nov2019_adcd_md5.txt

Enabling the license servers

To operate ZD&T Enterprise Edition, and authorize the licensee to use the product or any of its components, you need to enable the License Server. You can follow the steps that are provided in the checklist.



Attention: All license key files or USB hardware device activations expire after one year from the date they were generated, or after less than a year if the 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 or the USB hardware device activation each year. Renewal activates the license key or USB hardware device for a subsequent year or until the end of the fixed term if less than a year.

Table 11. Enablement checklist for software-based licensing

Software-based license		Required/ Optional	Complete
1	“Installing Software-based License Server” on page 46 Describe the steps to install Software-based License Server	Required	<input type="checkbox"/>
2	Obtaining an update file Describes the steps to obtain an update file for Software-based License Server.	Required	<input type="checkbox"/>
3	Applying the update file to the Software-based License Server Describes the steps to apply the update file to Software-based License Server.	Required	<input type="checkbox"/>

Table 12. Enablement checklist for hardware-based licensing

Hardware-based license		Required/ Optional	Complete
1	“Installing the Hardware-based License Server” on page 55 Describes the steps to install the Hardware-based License Server.	Required	<input type="checkbox"/>
2	Obtaining an update file Describes the steps to obtain an update file for Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Applying the update file to the Hardware-based License Server” on page 58 Describes the steps to apply the update file to ZD&T Enterprise Edition.	Required	<input type="checkbox"/>

Table 13. Switching from Hardware-based License Server to Software-based License Server

Hardware-based license to Software-based license		Required/ Optional	Complete
1	“Returning a hardware-based license key” on page 58 Describes the steps to return a hardware-based license key.	Required	<input type="checkbox"/>
2	“Uninstalling the Hardware-based License Server” on page 60 Describes the steps to uninstall Hardware-based License Server.	Required	<input type="checkbox"/>

Table 13. Switching from Hardware-based License Server to Software-based License Server (continued)

Hardware-based license to Software-based license		Required/ Optional	Complete
3	<p>“Installing Software-based License Server” on page 46</p> <p>Describes the steps to install Software-based License Server.</p>	Required	<input type="checkbox"/>
4	<p>Obtaining an update file for Software-based License Server</p> <p>Describes the steps to obtain an update file for Software-based License Server.</p>	Required	<input type="checkbox"/>
5	<p>Applying the update file to the Software-based License Server</p> <p>Describes the steps to apply the update file to Software-based License Server.</p>	Required	<input type="checkbox"/>

Software-based licensing

When you use the software-based licensing, the Software-based License Server provides a server for centralized management of license keys for one or more instances of ZD&T.

Before you enable the software-based licensing, you need to know the terms that are used in the enablement process.

Host ID and host name

Each license key file is uniquely identified in the Rational License Key Center with the host name and host ID of the Software-based License Server 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 Software-based License Server. 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 Software-based License Server for which it was generated.

Installing Software-based License Server

To install the Software-based License Server, run the installer from root user ID.

1. Open the directory that contains the installer ZDT_Install_EE_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **2** for Software-based License Server.

7. Select **1** to install.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. Read the warning message, and accept or decline by entering y or n.
10. Optional: Enter y to install all needed dependencies or enter n to decline.

By entering y, all required dependencies will be installed. The list of dependencies is mentioned in the “Prerequisites” on page 37. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

11. After the installation completes, run the following commands to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep zpdtdk
rpm -qa | grep zpdtdm
```

- UBUNTU operating system

```
dpkg -l | grep zpdtdk
dpkg -l | grep zpdtdm
```

If the package is installed successfully, the following output is displayed.

```
vim 1.10.55.04
```

Important: The installer creates the `ibmsys1` as the non-root user ID to make ZD&T work, you need to switch to `ibmsys1` and change the password. If you like to create the non-root user ID, use the following commands for reference. Make sure that the non-root user ID is a part of group `zpdtd`.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpdtd -m ibmsys2
```

To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

Alternatively, after you complete steps 1 - 4, you can install the Software-based License Server silently.

- To install the Software-based License Server on the virtual machine, run the following command:

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdtswlicense --ackVM
```

- To install the dependencies, run the following command:

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdtswlicense --ackVM --dep
```

Obtaining an update file

To enable the Software-based License Server, you must first obtain a unique update file, and then apply the update file to the Software-based License Server.

For Authorized User Single Session license

Learn about the steps to obtain an update file for Authorized User Single Session license.

1. Log in to the machine where you installed the Software-based License Server.
2. Go to `/opt/IBM/LDK`, and run the following command from root user ID. This step is to create a file that is known as the request file in the root home directory.

```
./request_license
```

3. Log in to the [Rational License Key Center \(RLKC\)](#), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
4. On the left side of the screen, click **Get Keys**.
5. Select the product line for **IBM Z Development and Test Environment**.
6. Select the license type that you want to apply to the key. If you purchased different license types, enter IBM Z DEVELOPMENT AND TEST ENVIRONMENT ENTERPRISE EDITION AUTHORIZED USER SINGLE SESSION. Then, one or more boxes are displayed to show your orders.
7. Check the boxes next to the orders from which you would like to use licenses, and click **Next**.
8. A screen is displayed to show a table that you must complete. Provide the request file that you generated at step 2, and enter a number in the **Emulated Central Processor** field. For more help, click the column headings in the table.
9. Click **Generate** at the bottom of the page. Then, a screen is displayed to download the update files.
10. Click **Download** for each update file that is generated to save the update file. Keep this file in the Linux machine where you have installed the Software-based License Server. 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: Use the binary transfer mode, but not a text mode such as ASCII. Also, spaces are not allowed in the path on the Linux machine.

For Software-based License Server

Learn about the steps to obtain an update file for Software-based License Server.

1. Log in to the machine where you installed the Software-based License Server.
2. Go to /opt/IBM/LDK, and run the following command from root user ID. This step is to create a file that is known as the request file in the root home directory.

```
./request_license
```

3. Log in to the [Rational License Key Center \(RLKC\)](#), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
4. On the left side of the screen, click **Get Keys**.
5. Select the product line for **IBM Z Development and Test Environment**
6. Select the license type that you want to apply to the key. If you purchased different license types, enter IBM Z DEVELOPMENT AND TEST ENVIRONMENT Enterprise Edition Resource Value Unit. Then, one or more boxes are displayed to show your orders.
7. Check the boxes next to the orders from which you would like to use licenses, and click **Next**.
8. A screen is displayed to show a table that you must complete. Provide the request file that you generated at step 2, and enter a number in the **Emulated Central Processor** field. For more help, click the column headings in the table.
9. Click **Generate** at the bottom of the page. Then, a screen is displayed to download the update files.
10. Click **Download** for each update file that is generated to save the update file. Keep this file in the Linux machine where you have installed the Software-based License Server. 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: Use the binary transfer mode, but not a text mode such as ASCII. Also, spaces are not allowed in the path on the Linux machine.

For Software-based License Server with Rational Tokens

Learn about the steps to obtain an update file for Software-based License Server with Rational Tokens.

1. From the root user ID on the Software-based License Server, run the following 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 Software-based License Server. You can use the file 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 Rational License Key Center, and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
4. On the left side of the screen, click **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 contains IBM Z Development and Test Environment. Make a note of the part number for the ZD&T 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.
8. Click **Generate** to generate the Rational Token license file. You are presented with a screen that shows 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 update file for your Software-based License Server.
11. A **Required Information** page opens and displays a table that you must complete. Provide the request file for the Software-based License Server for which the license key file is 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 ZD&T and ZD&T 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 software-based license server update file. For more help, click the column headings in the table.
12. Click **Generate** at the bottom of the page. A window opens for you to download the update file.
13. To save the file, click **Download** for each file that is 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.

Applying the update file to the Software-based License Server machine

After you obtain the update file, you need to apply it to the Software-based License Server to enable the license server.

To apply the update file to the Software-based License Server machine, complete the following steps:

1. Log in to the computer. If you are not running as a root user ID, enter `su` that is followed by the root password.
2. Change to the `/opt/IBM/LDK` directory:

```
cd /opt/IBM/LDK
```

3. Run the following command:

```
./update_license <path of the updated file>
```

This file must have downloaded in your machine in the steps of [Obtaining an update file for Software-based License Server](#). This command produces several messages that indicate that the update is successful.

4. Use root user ID to issue the following command from /opt/IBM/LDK.

```
./query_license
```

Then, you can see the number of available CPs. For example, the output is shown as below.

The following key is available:

```
HASP-SL key_id=432975633343422885 feature(s):
FID Feature Name Expiration Logins MaxLogins
333- CPU License Tue Oct 30,2018 19:59:59 39 100
334- ADCD License Tue Oct 30,2018 19:59:59 0 1
```

Host Information: zdt-lic-mgr localhost

These are the currently active sessions:

KeyID	FID	FeatureName	Address	User	Machine	LoginTime
432975633343422885	333	CPU License	9.26.158.161	ibmsys1	zdt-dev4	Mon Oct 30, 17:25:40

5. Go to /usr/z1090/bin, and run the following command from the non-root user ID (ibmsys1).

```
./uimserverstart
```

Returning a software-based license key

All license key files expire after one year from the date they were generated, or after less than a year if the 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. Also, if you need to move, update or uninstall the Software-based License Server, you must first return the activated license key.

Returning the license key less than 31 days before expiration

To generate a new license key file for a Software-based License Server, 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 Software-based License Server. You do not need to return the physical software-based license server 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.

To return a software-based license less than 31 days, complete the following steps:

1. From the root user ID on the Software-based License Server, run the following 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 Software-based License Server, 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](#), 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 Software-based License Server 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 the license key more than 30 days before expiration

To generate a new license key file for a Software-based License Server, 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 Software-based License Server. You do not need to return the physical software-based license server 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 Software-based License Server as part of the return process. This process consists of these general steps:

- Create a software-based license server request file from the Software-based License Server 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 Software-based License Server.
- 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. You can also use the **Return Keys** link.

To return a software-based license less than 30 days, complete the following steps:

1. Create a software-based license server request file from the Software-based License Server where the license key file is installed:
 - a. From the root user ID on the Software-based License Server, 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 Software-based License Server, and is used to generate the software-based license key file.
 - b. If needed, transmit the `Hostname_XXXXXXXXXX.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](#) , 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 Software-based License Server you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXXX.zip` file that you just created.
 - f. Click **Initiate Return**.
 - g. A window opens, prompting you to download the software-based license server update file with the file to invalidate the license on the Software-based License Server.
 - 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 software-based license server update file that invalidates the license key to the Software-based License Server.

3. Install the license key file that invalidates the current license key on the Software-based License Server.

- a. From the root user ID on the Software-based License Server, unzip the `Hostname_XXXXXXXXXX_update.zip` file, run the following command:

```
/opt/IBM/LDK/update_license Hostname_XXXXXXXXXX_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 the following command from the root user ID.

```
systemctl restart aksusbd.service
```

The Software-based License Server 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 Software-based License Server, run the following 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 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_XXXXXXXXXX.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](#), 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 Software-based License Server 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 Software-based License Server you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXXX.zip` file that you just created.
- f. Click **Complete Return**.
- g. A message is displayed to confirm that the license entitlements were successfully returned.

Returning Rational Tokens for Software-based License Server

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 Software-based License Server. You do **not** need to return the physical update file to the Rational License Key Center.

1. Log in to [Rational License Key Center](#) and select your account.
2. On the left side of the screen, select **Return Keys**.
3. Select **IBM Rational Tokens**.
4. A list with license entitlements that are assigned to them from the same Order Line is displayed. Locate the host description you are working with, and click **Return**. A message is displayed to confirm that the license entitlements were successfully returned.

Updating the Software-based License Server

To update the Software-based License Server, you need to return the existing license first. Then, run the installer from the root user ID to update the Software-based License Server, and enable the license server again.

Before you update the Software-based License Server, you need to return the existing license key. For the steps to update the Software-based License Server, see [“Returning a software-based license key” on page 50](#).

To update the Software-based License Server, complete the following steps:

1. Open the directory that contains the installer `ZDT_Install_EE_V12.0.5.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the `README.txt` file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **2** for Software-based License Server.
7. Select **2** to update.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. After the update completes, run the following command to verify whether the update is successful.

- RHEL operating system

```
rpm -qa | grep zpdtdk  
rpm -qa | grep zpdtuim
```

- UBUNTU operating system

```
dpkg -l | grep zpdtdk  
dpkg -l | grep zpdtuim
```

If the package is installed successfully, the following output is displayed.

```
uim 1.10.55.04
```

Alternatively, after you complete steps 1 - 4, you can update Software-based License Server silently by using the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --update --zdtswlicense
```

After the Software-based License Server is updated, you need to enable the license server by running the steps that are described in [“Enabling the license servers” on page 44](#).

Uninstalling the Software-based License Server

To uninstall the Software-based License Server, you need to return the license key, stop the Software-based License Server and UIM server first. Then, run the installer from the root user ID.

Before you uninstall the Software-based License Server, you need to complete the following steps first.

1. Return the existing license key by running the steps that are described in [“Returning a software-based license key” on page 50](#).

2. Stop the Software-based License Server by using the following command from the root user ID

```
service aksusbd stop
```

3. Stop the UIM server by using the command from a non-root user ID.

```
cd /usr/z1090/bin  
./uimserverstop
```

To uninstall the Software-based License Server, complete the following steps:

1. Run the installer from the root user ID.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

2. Select **2** for Software-based License Server.

3. Select **3** to uninstall.

4. After the uninstallation completes, run the following command to verify whether the uninstallation is successful.

- RHEL operating system

```
rpm -qa | grep zpdtuim
```

- UBUNTU operating system

```
dpkg -l | grep zpdtuim
```

If the package is uninstalled successfully, no package is listed.

Alternatively, after you complete steps 1 - 4, you can uninstall the Software-based License Server silently by using the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --uninstall --zdtswlicense
```

Hardware-based licensing

If you purchased hardware-based license or hardware-based license with Rational Tokens, you need to enable a hardware license server to operate ZD&T Enterprise Edition and authorize the licensee to use the product or any of its components. To enable the hardware license server, you can follow the steps that are provided in the checklist.

Before you enable the hardware-based licensing, you need to know the terms that are used in the enablement process.

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.

Figure 8. USB hardware device



In figure 1, the entire serial number is 02-00222.

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 ZD&T 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.

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 [Obtaining an update file for Hardware-based License Server with Rational Tokens](#) and [“Setting up Rational Tokens”](#) on page 185.

Installing the Hardware-based License Server

To install the Hardware-based License Server, run the installer from root user ID.

1. Open the directory that contains the installer ZDT_Install_EE_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **3** for the Hardware-based License Server.
7. Select **1** to install.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. Optional: Enter y to install all needed dependencies or enter n to decline.

By entering y, all required dependencies will be installed. The list of dependencies is mentioned in the [“Prerequisites”](#) on page 37. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

10. After the installation completes, run the following command to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep z1091
```

- UBUNTU operating system

```
dpkg -l | grep z1091
```

If the package is installed successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Important: The installer creates the `ibmsys1` as the non-root user ID to make ZD&T work, you need to switch to `ibmsys1` and change the password. If you like to create the non-root user ID, use the following commands for reference. Make sure that the non-root user ID is a part of group `zpdtd`.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpdtd -m ibmsys2
```

To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

Alternatively, after you complete steps 1 - 4, you can install the Hardware-based License Server silently by using the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdthwlicense
```

To install the dependencies silently, run the following command:

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdthwlicense --dep
```

Obtaining an update file

To enable a Hardware-based License Server, you must first obtain a unique update file for the Hardware-based License Server.

For Hardware-based License Server

Learn about the steps to obtain an initial update file for the Hardware-based License Server.

1. Log in to the [Rational License Key Center](#) (RLKC), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
2. On the left side of the screen, click **Get Keys**.
3. Select the product line for **IBM Rational Developer for System z Unit Test**.
4. Select the license type that you want to apply to the key. If you purchased different license types, enter `IBM RATIONAL DEVELOPMENT AND TEST ENVIRONMENT FOR SYSTEM Z PER RVU FOR SYSTEM Z LIC`. Then, one or more boxes are displayed to show your 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 ZD&T 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.

5. Check the boxes next to the orders from which you would like to use licenses, and click **Next**.
6. Complete the table that is displayed. You must enter the USB hardware device serial numbers that you want to activate, enter a number in the **Emulated Central Processor** field. For more help, click the column headings in the table.
7. Click **Generate** at the bottom of the page. Then, a screen is displayed to download the update files.

8. Click **Download** for each update file that is generated to save the update file. Keep this file in the Linux machine where you have installed the ZD&T Hardware License Server. 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: Use the binary transfer mode, but not a text mode such as ASCII. Also, spaces are not allowed in the path on the Linux machine.

For Hardware-based License Server with Rational Tokens

Learn about the steps to obtain an update file for Hardware-based License Server with Rational Tokens.

1. Log in to the [Rational License Key Center](#) (RLKC), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
2. On the left side of the screen, click **Get Keys**.
3. Select a product line for **IBM Rational Tokens**.
4. On the **Select License Keys** screen, one or more boxes for orders are displayed. Find the order that contains IBM Z Development and Test Environment. Make a note of the part number for the ZD&T 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.
6. Click **Generate** to generate the Rational Token license file. A screen with the generated license information is displayed.
7. Click **Download License Keys** to download the Rational Token license file to your computer. This token license file 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 to show a table that you must complete with the USB hardware device serial numbers and associated ZD&T 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 ZD&T Enterprise Edition. 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. A screen to download the update files is displayed.
11. Click **Download** for each update file to save the update file. 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, 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.

Applying the update file to the Hardware-based License Server

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.

Note:

1. Never set the system clock to a future date or time when the USB hardware device is plugged in. Otherwise, the hardware device is damaged.
2. If the USB hardware device is attached to a computer with a clock that is set to a future date, the device remembers the future date. When the time is set back to the current time, the USB device does not recognize this change, and will not work until the clock reaches the previously set future time. If the clock was set to a future time or date, which causes ZD&T is unable to authenticate with the USB hardware device, 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).

To apply the update file to the ZD&T Enterprise Edition machine, complete the following steps:

1. Connect the USB hardware device to the computer that is running ZD&T Hardware-based License Server.

Note: Connect only the USB hardware device that is being activated to your computer during this process. Otherwise, unpredictable results might occur.

2. Log in to the computer. If you are not running as the root user ID, enter `su` that is followed by the root password.
3. Change to the `/usr/z1090/bin` directory:

```
cd /usr/z1090/bin
```

4. Run the following command where `update-file.zip` is the full path and file name of the update file.

```
./Z1091_token_update -u update-file.zip
```

This command produces several messages that indicate that the update is successful.

5. After the update file is successfully applied, unplug the USB hardware key, and wait for at least 10 seconds. Then, reconnect the hardware key. It is now ready for routine ZD&T operation.
6. Restart the local SHK license server from a root user ID by using the command:

```
/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart
```

This step can ensure that the system discovers the newly applied licenses.

7. Optionally, verify that the licenses are now available on your USB hardware device by using the command:

```
./Z1091_token_update -status
```

Tip: After ZD&T is started, you can verify the effective Z serial number, and the expiration date of the license.

8. Go to `/usr/z1090/bin`, and run the following command from the non-root user ID (ibmsys1).

```
./uimserverstart
```

Returning a hardware-based license key

USB hardware device activations expire after one year from the date they were generated, or after less than a year if the 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 return a hardware-based license key or Rational Tokens for Hardware-based License Server, and renew the USB hardware device activation each year.

Returning an existing hardware-based license key

For perpetual license entitlements, the hardware-based license are set to expire one year from the date that 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 that 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 the term end. After the 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. 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 way is to use the **View Keys by Host** link. You can also use the **Return Keys** link.

For ZD&T, 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 \(RLKC\)](#), 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 ID** column.
4. A table is displayed with data for the selected USB hardware device. 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 that you are working with, and click **Return**. A message is displayed to confirm that the license entitlements were successfully returned.

Returning Rational Tokens for Hardware-based License Server

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.

1. Log in to [Rational License Key Center \(RLKC\)](#), and select your account.
2. On the left side of the screen, select **Return Keys**.
3. Select **IBM Rational Tokens**.
4. A list of devices with license entitlements that are assigned to them from the same Order Line is displayed. Locate the 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.

Updating the Hardware-based License Server

To update the Hardware-based License Server, you need to return the existing license first. Then, run the installer from root user ID.

Before you update the Hardware-based License Server, you need to return the existing license key. For the steps to update the Hardware-based License Server, see [“Returning a hardware-based license key” on page 58](#).

1. Open the directory that contains the installer `ZDT_Install_EE_V12.0.5.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the `README.txt` file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **3** for Hardware-based License Server.
7. Select **2** to update.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.

If the package is updated successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Alternatively, after you complete steps 1 - 4, you can update the Hardware-based License Server silently by using the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --update --zdthwlicense
```

After the Hardware-based License Server is updated, you need to enable the license server by running the steps that are described in [“Enabling the license servers” on page 44](#).

Uninstalling the Hardware-based License Server

To uninstall the Hardware-based License Server, you need to return the license key, stop the UIM server and Hardware-based License Server first. Then, run the installer from root user ID.

Before you uninstall the Hardware-based License Server, you need to complete the following steps first.

1. Return the existing license key by running the steps that are described in [“Returning a hardware-based license key” on page 58](#).
2. Stop the UIM server by using the command from a non-root user ID.

```
cd /usr/z1090/bin  
./uimserverstop
```

3. Stop the Hardware-based License Server by using the following command.

```
service aksusbd stop
```

```
cd /opt/safenet_sentinel/common_files/sentinel_keys_server  
./loadserv stop
```

To uninstall the Hardware-based License Server, complete the following steps:

1. Run the installer from the root user ID.


```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

2. Select **3** for Hardware-based License Server.
3. Select **3** to uninstall.

Alternatively, after you complete steps 1 - 4, you can run the following command to uninstall the Hardware-based License Server silently.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --uninstall --zdthwlicense
```

Installing Enterprise Edition

To install ZD&T Enterprise Edition, run the installer from the root user ID.

1. Open the directory that contains the installer ZDT_Install_EE_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **1** for ZD&T Enterprise Edition.
7. Select **1** to install.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. Enter y if you want a user ID 'zdt' that is automatically created on the system to run the Enterprise Edition web server process. If you enter n, you will be promoted to specify the existing user ID that you want to use to run the web server process.
10. Enter y if you want to install ZD&T Enterprise Edition in the default directory /opt/ibm. If you enter n, you will be promoted to specify the directory that you want to install ZD&T Enterprise Edition.
11. Enter y if you want to use the default port to access ZD&T Enterprise Edition server. If you enter n, specify your installation port.
12. Enter y if you want to change the default password for the user ID to log in ZD&T Enterprise Edition server. Then, specify and verify your password.
13. After the installation completes, run the following command to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep zdtapp
```

- UBUNTU operating system

```
dpkg -l | grep zdtapp
```

If the package is installed successfully, the output contains the string zdtapp 12.0.5.0.

Alternatively, after you complete steps 1 - 4, you can run the following commands to install ZD&T Enterprise Edition silently.

- To install Enterprise Edition without specifying the user ID and installation path.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdtee --zdtee_port=9443
```

Note: If you install Enterprise Edition without specifying the user ID and installation path, the default user ID will be 'zdt', and the default installation path will be /opt/ibm.

- To install Enterprise Edition with specifying the user ID and installation path.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdtee --zdtee_port=9443 --zdtee_user=<username>  
--zdtee_path=<installationpath>
```

Note: Silent installation does not install any required dependencies. You need to install the dependencies before you start the installer silently. For the list of dependencies, see [“Prerequisites” on page 37](#).

Default authentication and authorization

ZD&T Enterprise Edition comes with a default user ID and password. You can also change the default password.

The default user ID 'zdtadmin' of Enterprise Edition is defined in the file <install directory>/Liberty/usr/servers/zDTServer/basicauth-conf.xml, and the default password of the user ID is 'password'. This password is authorized as an administrator account in Enterprise Edition. To change the default password, complete the following steps.

1. Find the encryption key that is specified by wlp.password.encrypted.key in the /opt/ibm/zDT/Liberty/usr/servers/zDTServer/bootstrap.properties.
2. If the Java path is not set up, you need to set up the path before you run the next command.
3. Run the following command where you installed ZD&T Enterprise Edition.

```
/opt/ibm/zDT/Liberty/bin/securityUtility encode --encoding=aes --  
key=<encryption_key_found_above> <new password value>
```

Copy the following encoded string into the file <install directory>/Liberty/usr/servers/zDTServer/basicauth-conf.xml. For example, replace the text in bold with the new encoded password value.

```
<!-- user authentication -->  
<basicRegistry id="zDT" realm="zDT">  
<user name="zdtadmin"  
password="*{aes}AM1LZsnwLRNsVtYAiwhqVD09/RL+NgYthDZXZhQgARtB*" />  
</basicRegistry>
```

For more information about authentication, see [“Starting and accessing the Enterprise Edition web server” on page 64](#).

For more information about authorization, see [“Configuring authorization” on page 72](#).

Installing ZD&T emulator

To install ZD&T emulator, run the installer from the root user ID.

1. Open the directory that contains the installer ZDT_Install_EE_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xvzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **4** for ZD&T emulator.
7. Select **1** to install.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. Optional: Enter y to install all needed dependencies or enter n to decline.

By entering y, all required dependencies will be installed. The list of dependencies is mentioned in the “Prerequisites” on page 37. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

10. Read the warning message, and accept or decline by entering y or n.
11. After the installation completes, run the following command to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep z1091
```

- UBUNTU operating system

```
dpkg -l | grep z1091
```

Alternatively, after you complete steps 1 - 4, you can install ZD&T emulator silently.

- To install ZD&T emulator, run the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdem
```

- To ZD&T emulator with all dependencies, run the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdem --dep
```

- To ZD&T emulator with network configuration, run the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --install --zdem --net
```

Note: ZD&T emulator cannot be installed on the same system as Software-based License Server, Hardware-based License Server, and ZD&T Enterprise Edition.

Optional: Installing the z/OS files

This step is only required when you need to transfer application images from z/OS, and you need to install the mainframe portion of ZD&T Enterprise Edition.

To install ZD&T Enterprise Edition host files, complete the following steps:

1. After the installer ZDT_Install_EE_V12.0.5.0.tgz run on a Linux machine to install the web user interface, the directory /opt/ibm/zDT/zSystem contains the IBM Z components, the program directory file HALMC05.pdf, and other files, including the jar file zDTMainframeSFTP.jar.
2. For the installation instructions of IBM Z component, refer to the details in HALMC05.pdf, specifically in section 6.0 "Installation Instructions".
3. Transfer /opt/ibm/zDT/zSystem/zDTMainframeSFTP.jar to the USS folder /usr/lpp/IBM/zdt that is located at the source z/OS.

Note: The data set hlq.IBM.HALMC05.F2 obtained at step 6.1.3 in HALMC05.pdf is a usable program library. If you want to install the Enterprise Edition without SMP/E, complete the following steps:

1. Choose a value for `dsnprefix`. Then, rename `hlq.IBM.HALMC05.F2` to `dsnprefix.SFEUAUTH`, and rename `hlq.IBM.HALMC05.F3` to `dsnprefix.SFEUEXEC`.
2. Go to step 6.1.11 by skipping steps from 6.1.4 to 6.1.10.

Starting and accessing the Enterprise Edition web server

The script to start the Enterprise Edition web server ensures that the server process runs under the user ID that is specified during the Enterprise Edition installation. The user ID can be changed by modifying `ZDTEE_USER` in the `/opt/ibm/zDT/bin/zdtee_env.sh` script file. You also need to change the file ownership of all the files in the directory `/opt/ibm/zDT/Liberty/usr/servers/zDTServer/` to the user ID, for example, `chmod -R newuser:root /opt/ibm/zDT/Liberty/usr/servers/zDTServer/`. If the web server is already started, you can run the `/opt/ibm/zDT/bin/stopServer.sh` command before you restart the server by using the newly specified user ID.

To start the ZD&T Enterprise Edition web server, you can run the command by using any user ID. However, the user ID must have the permission to switch to the user ID that is specified during the installation. By default, you can use the `sudo` command to switch to another user ID. If your Linux environment uses other privilege management tools than `sudo`, you can change the command by modifying `PRIVILEGED_ZDTEE_CMD` in the `/opt/ibm/zDT/bin/zdtee_env.sh` script file.

To start the ZD&T Enterprise Edition web server, run the following command.

```
/opt/ibm/zDT/bin/startServer.sh
```

Open the browser, and enter the URL that is contained in the output. Then, enter the default user ID (`zdtadmin`) and Password (`password`) to log in to the web server.

Authentication for the Enterprise Edition application

Before you start the ZD&T Enterprise Edition application server, you can modify how authentication is accomplished by the Enterprise Edition.

Basic Authentication

By default, ZD&T Enterprise Edition uses a basic registry as the user registry for authentication and provides a default user ID `'zdtadmin'`. Also, you can add more users to this registry.

If you need to change the default authentication before you modify the basic authentication, see [“Default authentication and authorization”](#) on page 62.

To add more users to the basic registry, complete the following steps before you start the server.

1. Find the encryption key that is specified by `wlp.password.encryption.key` in the `/opt/ibm/zDT/Liberty/usr/servers/zDTServer/bootstrap.properties`.
2. If the Java path is not set up, you need to set up the path before you run the next command.
3. Run the following command where you installed ZD&T Enterprise Edition.

```
/opt/ibm/zDT/Liberty/bin/securityUtility encode --encoding=aes --  
key=<encryption_key_found_above> <new password value>
```

Copy the following encoded string into the file `<install directory>/Liberty/usr/servers/zDTServer/basicauth-conf.xml`. For example, replace the text in bold with the new encoded password value.

```
<basicRegistry id="zDT" realm="zDT">  
  <user name="zdtadmin" password="{aes}AM1LZsnwLRNsVtYAiwqhVD09/RL+NgYthDZXzhQgARtB" />  
  <user name="new user" password="<new user password>" />  
  <user name="another new user" password="<another new user password>" />  
</basicRegistry>
```

This step only allows the user to authenticate to the ZD&T Enterprise Edition web server. For more information about authorization, see [“Configuring authorization”](#) on page 72.

Lightweight Directory Access Protocol (LDAP)

Learn about the steps to enable LDAP authentication for ZD&T Enterprise Edition.

The Lightweight Directory Access Protocol (LDAP) settings for the server are defined in the <install directory>/Liberty/usr/servers/zDTServer/ldap-conf.xml file. To enable the support, refer to [Configuring LDAP user registries in Liberty](#) to modify the file and define your LDAP configuration before you start the server. The appSecurity-2.0, ldapRegistry-3.0, and transportSecurity-1.0 Liberty features are already included in the configuration of the server, and these steps are unnecessary. The truststore of the server is located in the <install directory>/Liberty/usr/servers/zDTServer/resources/security/zdttrust.p12. The password for the zdttrust.p12 file is 'changeme', and the client needs to know the information to add their certificate to the file. If you are communicating with an SSL-enabled LDAP server, you need to add your signer certificate for the LDAP server to this keystore file.

This step only allows the user to authenticate to the ZD&T Enterprise Edition web server. For more information about authorization, see [“Configuring authorization” on page 72](#).

User's Guide

A set of User's Guide that describes the usage of ZD&T Enterprise Edition are available to help you use Enterprise Edition. Each of these User's Guide has specific target functions.

- *Enterprise Edition User's Guide* describes all detailed steps to create, monitor, and deploy application images by using the web user interface.
- *REST APIs User's Guide* describes all detailed steps to access the features of Enterprise Edition web user interface by calling REST APIs.

Enterprise Edition User's Guide

This document describes the steps to create, monitor and deploy application images by using the Enterprise Edition web user interface.

Overview of Enterprise Edition

ZD&T Enterprise Edition provides a web user interface that can create, monitor, and deploy application images to a target environment more quickly with the security and control you need.

By using Enterprise Edition, you can create, monitor, and deploy application from the following resources:

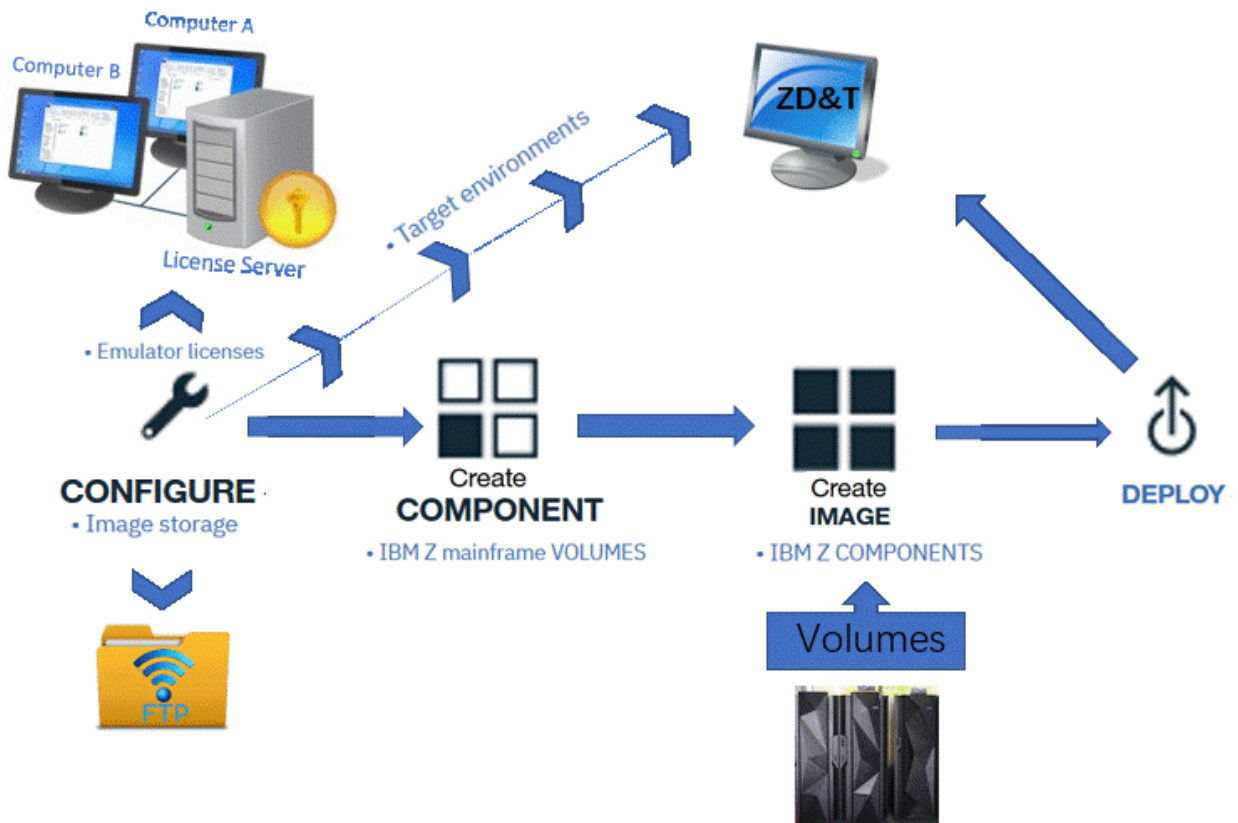
- IBM Z components
- Application Developers Controlled Distributions (ADCD)
- Import exiting content

Application images creation from IBM Z components

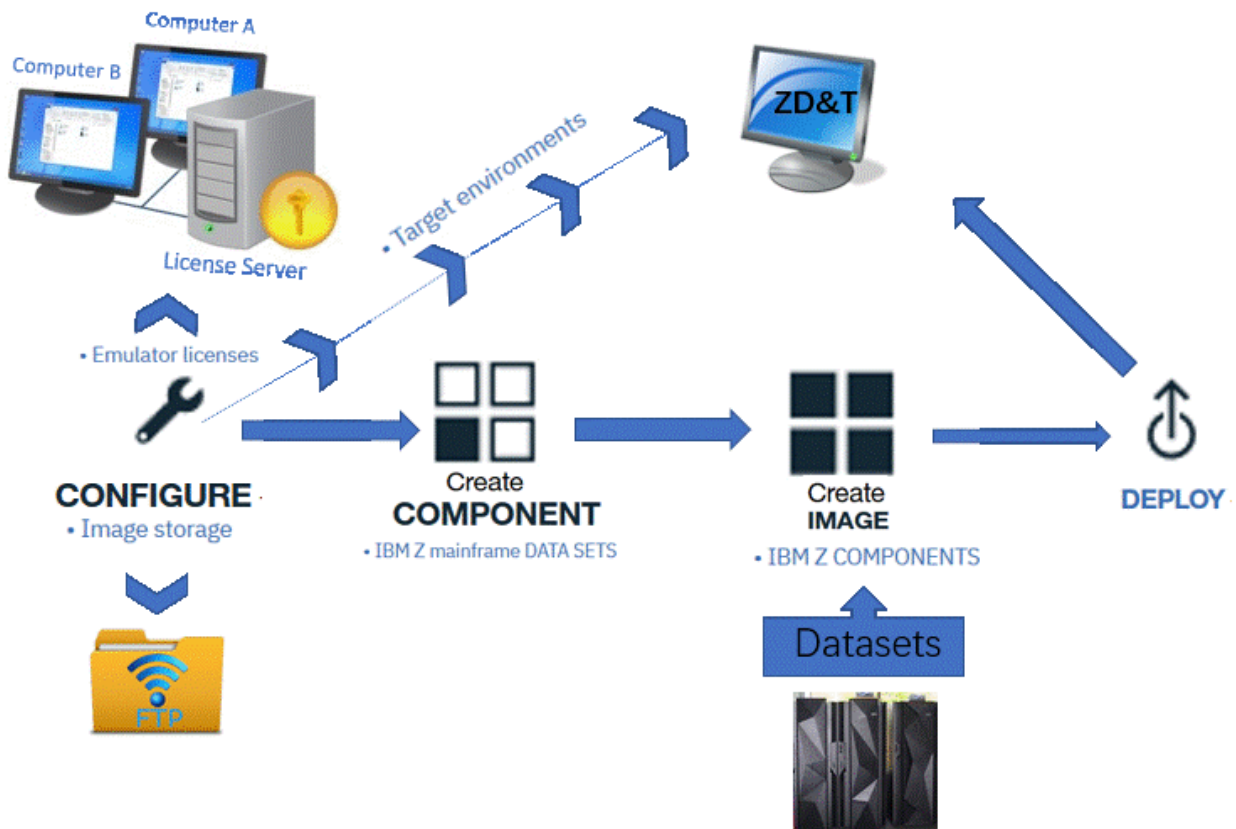
Before you create application images from IBM Z components, you can create application components from the following resources:

- IBM Z mainframe data sets
- IBM Z mainframe volumes
- IBM Z mainframe CICS
- IBM Application Discovery projects

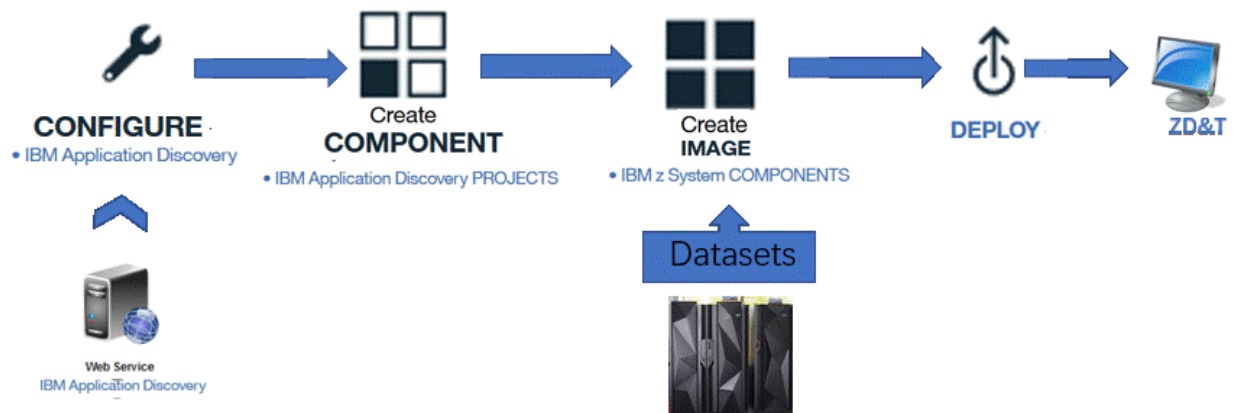
Before you provision ZD&T from z/OS or transfer single or multi volumes, you need to configure the FTP server that is your intermediate image storage, License Server that provides the license to ZD&T emulators, and target machine that provides the environment for the development and testing. Then, create application components by selecting the volumes from source z/OS, create the application images from the created components, and deploy the images to the target machine.



To transfer data sets either from z/OS or existing ZD&T machine, optionally, you need to validate the configuration of FTP server, License Server, and target machine that you must have configured when you provision the base target machine. Then, create application components by selecting the data sets from source z/OS, create the application images from the created components, and deploy the images to the target machine.

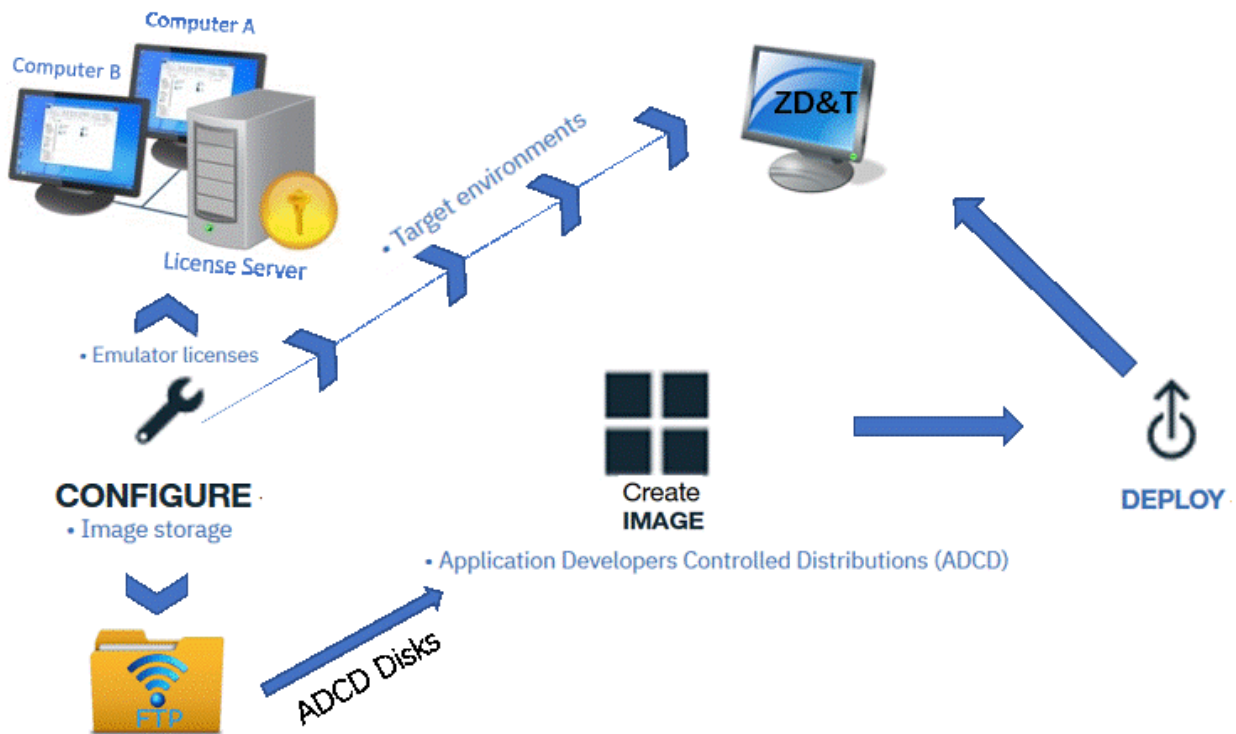


ZD&T Enterprise Edition can be integrated with IBM Application Discovery. If you have built your project in Application Discovery, and want to transfer artifacts such as data sets from source Z machine to the provision ZD&T machine, you need to configure connection with your IBM Application Discovery web services. Then, create application images from the created components, and deploy the images to the target machine.



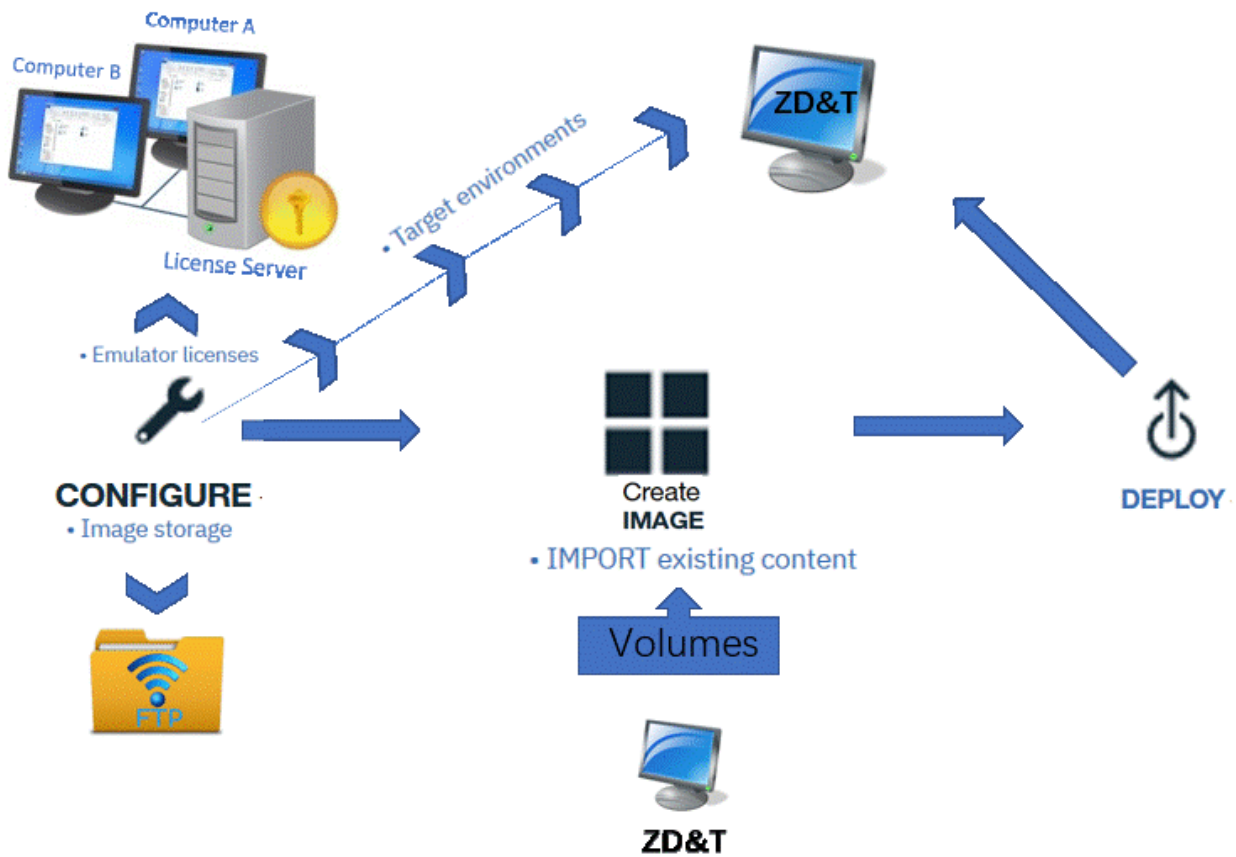
Application images creation from ADCD

Before you provision ZD&T from ADCD, you need to configure the FTP server that is your intermediate image storage, License Server that provides the license to ZD&T emulators, and target machine that provides the environment for the development and testing. Then, download all ADCD disks to FTP server, create application images by selecting the subsystems, such as CICS, IMS, DB2 and so on, and the required versions of the subsystems, and deploy the images to the target machine.



Application images creation from existing ZD&T

Before you provision applications from existing ZD&T, you need to configure the FTP server that is your intermediate image storage, License Server that provides the license to ZD&T emulators, and target machine that provides the environment for the development and testing. Then, import existing content by providing the ZD&T connection strings and devmap, and deploy the images to the target machine.



Firewall ports list

Before you install and use ZD&T Enterprise Edition, make sure that you are aware of the following firewall ports that Enterprise Edition installer opens.

Table 14. Mainframe

PORT	DIRECTION	USAGE	CONFIGURABLE
22	BOTH	SSH For the connection from Enterprise Edition web server	YES This port can be configured on the Enterprise Edition web user interface

Table 15. Enterprise Edition web server

PORT	DIRECTION	USAGE	CONFIGURABLE
22	BOTH	SSH For the installation and maintenance of the Enterprise Edition web server	No
9443	BOTH	SSL	No

Table 16. License Server

PORT	DIRECTION	USAGE	CONFIGURABLE
22	BOTH	SSH For the installation and maintenance of the Enterprise Edition web server	No
1947	BOTH	Licensing	No
9450	BOTH	Licensing	No
9451	BOTH	Licensing	No

Table 17. Target Environment (emulator)

PORT	DIRECTION	USAGE	CONFIGURABLE
22	BOTH	SSH For the installation and maintenance of the Enterprise Edition web server	YES From Enterprise Edition web server
1947	BOTH	Licensing	No
2022	BOTH	SSH For the Z instance that is running on emulator	No
3270	BOTH	3270 terminals	No
9450	BOTH	Licensing	No
9451	BOTH	Licensing	No

Note: During the deployment to the target environment, the process automatically configures the ports and firewall rules that are on target environment. Only port 22 is an exception, the port needs to be enabled before the deployment process starts; otherwise, an error message occurs when you deploy an application image. All other ports are forwarded to the emulator Z instance.

Table 18. FTP server

PORT	DIRECTION	USAGE	CONFIGURABLE
20	BOTH	FTP	No
21	BOTH	FTP	No

Table 19. SFTP server

PORT	DIRECTION	USAGE	CONFIGURABLE
22	BOTH	SFTP	No

Note: From ZD&T version 11.0.2.1, Enterprise Edition web server uses passive mode for all FTP connections, and port 20 is not required.

Limitations

Before you use ZD&T Enterprise Edition, make sure that you are aware of the following known limitations.

Application Developer Controlled Distribution (ADCD)

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

Creation support

- ZD&T Enterprise Edition supports only the creation of monoplex-emulated Z environments now.
- The CICS components and data set components can be included only in the ADCD image of z/OS 2.3 May 2018 Edition or later versions.

On-going extracted volumes

When the creation of application images is in progress, if the WebSphere® Liberty server that hosts the Enterprise Edition is shut down, the permissions to access IBM Z will be lost, which will cause a failure to extract the remaining volumes. Then, you need to rerun the creation process to extract 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 data set is referred in the SYSIN card, IBM Application Discovery might not be able to identify the data sets, and cannot pick data sets 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 data set 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 the 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” on page 126](#).
- 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 the 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.

Db2 extraction

Db2 table data is unloaded by using the standard Db2 UNLOAD utility with the `FORMAT INTERNAL` option. Only the data that is compatible with the option can be extracted. For more information about the `FORMAT INTERNAL` option, see [Db2 online utilities](#).

The Db2 tables that have a restricted status or incomplete definition will not be displayed or available for an extraction.

The Db2 tables that have LOB or XML columns are not supported because these columns are not compatible with the `FORMAT INTERNAL` option of the UNLOAD utility.

The Db2 tables that have edit, field, or validation procedures are not supported.

The Db2 components can be deployed only to ADCD-based target environments.

Docker container support

zPDT is supported in the Docker container with the following requirements.

- The sum of memory that is used by each Docker container that is needed to run the base Linux system cannot exceed the Linux system memory.
- The packages that are required by zPDT need to be installed in the Docker container. For example, rpms, Debian.
- The Docker container must run in the Privileged mode.
- The Docker container must have the read access to Linux cgroups.
- You need to understand the security ramifications of setting up your network configuration. Only the required traffic is allowed.
- You must manage any OSA performance issues.
- You must expose any zPDT core image from the Docker container to the Linux host file system.
- You must make sure the following requirements for debug purposes.
 - Each zPDT container must expose the zPDT logs directory to the native Linux host file system. For example, on the Linux host, the directory /zPDTlogs/container1 needs to contain the /home/<user>/z1090/logs of the container1, and the directory /zPDTlogs/container2 needs to contain the /home/<user>/z1090/logs of container2.
 - The Linux window that starts zPDT needs to be available.
 - The z/OS system console needs to be available.
 - Linux c debugger package gbx needs to be installed in the container.

Configuring Enterprise Edition

Learn how to configure ZD&T Enterprise Edition.

After the Enterprise Edition server environment is configured, you need to start the Enterprise Edition 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 Enterprise Edition server, see [“Starting and accessing the Enterprise Edition web server”](#) on page 64.

Configuring authorization

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 before you configure the users.

Administrator configuration

To log in to the Enterprise Edition, use the default User ID **zdtadmin** that has a default password of **password**.

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

1. Click **Users** on the **QUICK START** page, or click the left-top button to navigate to **Settings > Users**.
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 Enterprise Edition.
5. Click **Add User**.
6. Sign out of Enterprise Edition.
7. Sign in to Enterprise Edition with your LDAP user ID and password to ensure all the configuration is proper.
8. To open the **Users** page, click the **Settings > Users**, 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 Enterprise Edition.

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

1. Click **Users** on the **QUICK START** page, or click the left-top button to navigate to the **Settings > Users**.
2. Add the users who need to have access to the Enterprise Edition, 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

The target environment page is available for all users with different roles. You can provide the list of Linux or OpenStack cloud target environments that the application images will be deployed to.

Configuring a Linux target environment

Learn about the steps to configure, or delete a Linux target environment.

To configure a Linux target environment, complete the following steps:

1. Click **Target environments** on the **QUICK START** page, or click the left-top button to go to **Settings > Target environments**.
2. Click **Add Target Environment**.
3. Select **Linux**.
4. Enter the label of the target environment, the qualified host name, and Secure Socket Shell (SSH) port number of the environment.

Note: This descriptive target environment label is used to help recognize the target environment within this application. This label must be unique and use a maximum of 255 characters.

5. Add the target environment in one of the following ways:
 - Clear **Adjust advanced settings**, and click **Add Target environment** to save the target environment to the Enterprise Edition repository.
 - Select **Adjust advanced settings**, and click **Next**. Then, complete the following optional steps:
 - a. Select the sudo option for the privilege management. Or, add the required privilege management command and validation command.

For more information about configuring sudo access, see [Sudo access configuration](#).
 - b. Modify the number of volumes that you want to extract and transfer simultaneously.
 - c. Select if you want to install the required packages.

If you clear the option, refer to [Access to software repository to run YUM or apt-get commands](#) to check the package list, and make sure that the image you select installed all the required packages.
 - d. Select if you want to configure the user settings during the deployment. Then, click **Next**.

If you clear the option, make sure that you manually configure the users and group settings before you deploy application images. For more information, see [Users and group settings](#).
 - e. Change the default emulated z/OS IP address 10.1.1.2 and the default emulated z/OS Gateway IP address 10.1.1.1.

Note: The customization of the IP address is supported only for the application images that are created from IBM Z mainframe and ADCD.

- f. Modify the Linux operating system routing rules to the emulated z/OS by changing the default TCP and UDP port forwarding and redirection rules.

Note: To change the ports, follow these port routing rules:

- For the ports that are forwarded from Linux to z/OS, enter a range of ports that are routed to the same ports to z/OS, and ensure that the first number must be smaller than the second number. For example, `23:2021`. However, the ports 22, 9450, and 9451 cannot be included in the range because these ports are used. The port 22 is required for the TCP port rules of the redirected ports, 9450 is used by the license server, and 9451 is used by the UIM server.
- For the ports that are redirected from Linux to z/OS, enter a port that is routed to a different port to z/OS. For example, `2022>22`. Also, the rule `2022>22` is required for TCP port rules.

If you clear the option, make sure that you manually configure the network before you deploy application images. For more information, see [“Network configuration” on page 40](#).

- g. Click **Add Target environment** to save the target environment to the Enterprise Edition repository.

6. Specify the name of the Linux network interface that can be used for routing the emulated z/OS network traffic.

Deleting a target Linux environment

To delete a target environment, click **Delete**. If you deployed application images to a Linux target environment, when you click **Delete**, the following two options will be displayed:

- Remove ZD&T instances from the target system and revert network configurations.
- Remove the application images (volume files).

The options are selected by default. You can select to remove ZD&T instances and revert network configurations, but keep the application images in the deployment directory. Or, you can clear the option to remove ZD&T instances and revert network configuration, and the application images also cannot be removed.

Configuring an OpenStack cloud target environment

Learn about the steps to configure an OpenStack cloud target environment.

To configure an OpenStack cloud target environment, complete the following steps:

1. Click **Target environments** on the **QUICK START** page, or click the left-top button to go to **Settings > Target environments**.
2. Click **Add Target Environment**.
3. Select **Cloud**.
4. Enter the label of the target environment, and choose the level of your authentication, **Domain** or **Project**. Then, enter the credential of your cloud account, and click **Next**.

Note: This descriptive target environment label is used to help recognize the target environment within this application. The label must be unique and use a maximum of 255 characters.

5. Enter the keystone URL, domain name, project name. Then, click **Next**.
6. Select an availability zone from the drop-down list to provision an instance on the cloud platform.
7. Select a flavor, a server image from the drop-down list, and enter the default user ID of the image with a maximum 8 characters. Then, click **Next**.

Note: When you select a flavor, 3 CPU and 4-GB RAM are needed for a better performance.

8. Select a network and a floating IP from the drop-down list, and specify the name of the Linux network interface that will be used for routing the emulated z/OS network traffic. Then, click **Next**.

Note: The **Requires floating IP** is optional. If you select to assign a floating IP, check the box, and provide the name of the network that is used to request the floating IP.

9. Add the target environment in one of the following ways.

- Clear **Adjust advanced settings**, and click **Add Target environment** to save the target environment to the Enterprise Edition repository.
- Select **Adjust advanced settings**, and click **Next**. Then, complete the following optional steps:
 - a. Modify the number of volumes that you want to extract and transfer simultaneously.
 - b. Select the sudo option for the privilege management. Or, add the required privilege management command and validation command.
 - c. Select if a snapshot of an application image volume is allowed, which can significantly improve the efficiency of subsequent deployments.
 - d. Select if you want to install the required Linux packages during the deployment.

If you select the option, you must ensure the following prerequisites:

- A Red Hat software repository for 'yum' must be available and accessible by a Red Hat server image.
- An Ubuntu software repository for 'apt-get' must be available and accessible by an Ubuntu server image.

If you clear the option, refer to [Access to software repository to run YUM or apt-get commands](#) to check the package list, and make sure that the image you select installed all the required packages.

- e. Select if you want to configure the user settings during the deployment.

If you clear the option, make sure that you manually configure the users and group settings before you deploy application images. For more information, see [Users and group settings](#).

- f. Change the default emulated z/OS IP address 10.1.1.2 and the default emulated z/OS Gateway IP address 10.1.1.1.

Note: The customization of the IP address is supported only for the application images that are created from IBM Z mainframe and ADCD.

- g. Modify the Linux operating system routing rules to the emulated z/OS by changing the default TCP and UDP port forwarding and redirection rules.

Note: To change the ports, follow these port routing rules:

- For the ports that are forwarded from Linux to z/OS, enter a range of ports that are routed to the same ports to z/OS, and ensure that the first number must be smaller than the second number. For example, 23:2021. However, the ports 22, 9450, and 9451 cannot be included in the range because these ports are used. The port 22 is required for the TCP port rules of the redirected ports, 9450 is used by the license server, and 9451 is used by the UIM server.
- For the ports that are redirected from Linux to z/OS, enter a port that is routed to a different port to z/OS. For example, 2022>22. Also, the rule 2022>22 is required for TCP port rules.

If you clear the option, make sure that you manually configure the network before you deploy application images. For more information, see [“Network configuration”](#) on page 40.

- h. Click **Add Target environment** to save the target environment to the Enterprise Edition repository.

Note: The configured OpenStack cloud target environment cannot be deleted if some systems were provisioned to the environment.

Configuring Image storage

Learn how to configure the image storage.

The **Image storage** is a crucial function of the Enterprise Edition. You can transfer and store all of the created images on the Image storage by using SFTP (SSH File Transfer Protocol) and FTP (File Transfer Protocol).

Before you configure the image storage, you need to install an FTP or an SFTP server first.

Installing an FTP server

To install an FTP server, complete the following steps:

1. Run the following command to install an FTP server.

```
yum install vsftpd ftp
```

2. Enable the FTP service.

```
Systemctl enable vstpd.service
```

3. Edit the `vsftpd.conf` file, and make sure that `local_enable=YES`, `write_enable=YES`, `anonymous_enable=YES` are mentioned and uncommented. The file is presented in the `/etc/vsftpd/vsftpd.conf`.

4. Create an FTP user ID and password.

```
useradd userid  
passwd userid
```

5. Enter your new password, and confirm the new password again.

6. Start the FTP server.

```
systemctl start vsftpd.service
```

7. Check whether the port 21 is enabled. If not, run the following command to enable the port 21.

```
iptables -I INPUT -p tcp --dport 21 -j ACCEPT
```

Installing an SFTP server

To install an SFTP server, complete the following steps:

1. Run the following command to check whether SSH is installed.

```
rpm -qa|grep ssh
```

2. If SSH is not installed, run the following command.

```
yum install openssh-server
```

3. Open the port 22.

```
iptables -I INPUT -p tcp --dport 22 -j ACCEPT
```

4. Create a directory and change the authority.

```
mkdir -p /data/sftp  
chmod 701 /data
```

5. Add a group, and a user for SFTP. Then, provide your new password.

```
groupadd sftpusers  
useradd -g sftpusers -d /upload -s /sbin/nologin sftpuser  
passwd sftpuser
```

6. Create a directory for the file transfer. Then, make the changes that you need.

```
mkdir -p /data/sftpuser/upload  
chown -R root:sftpusers /data/sftpuser  
chown -R sftpuser:sftpusers /data/sftpuser/upload
```

7. Add the following lines in the `/etc/ssh/sshd_config`.

```
Match Group sftpuser  
ForceCommand internal-sftp
```


8. Restart the SFTP service.

```
service sshd restart
```

Connecting the FTP or SFTP server

After you install an FTP or an SFTP server, you can go to the Enterprise Edition web user interface, and complete the following steps:

1. Click **Image storage** on the **QUICK START** page, or click the left-top button to navigate to **Settings > Image storage**.
2. Select **SFTP** or **FTP** you need to use.
3. Enter the fully qualified host name of the storage server along with the port, directory, user ID, and password.
4. Click the **Test Connection** button to ensure that the Enterprise Edition system is able to communicate with the Image service.
5. Click the **Save** button.

Configuring 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 **Settings > 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 125 for reference.

Related information

[IBM AD Build User Guide](#)

Configuring IBM Z mainframe systems

Learn how to configure, define, or delete IBM Z mainframe systems with CICS regions and Db2 subsystems.

IBM Z mainframe runs z/OS that is used for extracting data to be used in IBM Z Development and Test Environment instances. All commands to these systems will be run by using an SSH connection to the z/OS UNIX System Services.

To configure **IBM Z mainframe systems**, complete the following steps:

1. Click **IBM Z mainframe systems** on the **QUICK START** page, or click the left-top button to navigate to **Settings > IBM Z mainframe systems**.
2. Click **Add new mainframe**.
3. Enter the specified host and SSH port of the IBM Z mainframe system that is used to extract z/OS data.
4. Click **Add**.

Note: The configured mainframe system cannot be deleted if some components were created from the mainframe system.

After you add an IBM Z mainframe system, you can also configure CICS regions and Db2 subsystems on the mainframe system.

Configuring CICS regions

If you want to create CICS components, complete the following steps:

1. Expand the IBM Z mainframe that you added, and click **Define a CICS region**.
2. Specify the CICS region details that are required.
3. Add a maximum of four active lists that you need in the region.
4. Click **Add**.

Configuring Db2 subsystems

If you want to create Db2 components, complete the following steps:

1. Expand the IBM Z mainframe that you added, and click **Define a Db2 subsystem**.
2. Specify the Db2 subsystem details that are required.
3. Add the Db2 subsystem in one of the following two ways.
 - Clear **Use IBM Db2 Administration Tool for z/OS**, and click **Add**.
 - Select **Use IBM Db2 Administration Tool for z/OS**. Enter the plan name of the IBM Db2 Administration Tool for z/OS DDL Generation, for example, ADB2GEN, and enter the IBM Db2 Administration Tool for z/OS load library, for example, SADBLLIB. Then, click **Add**.

After you configure a CICS region or a Db2 subsystem, you can also edit or delete the CICS region and Db2 subsystem at any time.

Note: To configure a CICS region or a Db2 subsystem, you need to have some basic knowledge of CICS regions or Db2 subsystems on the mainframe system.

Configuring 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 versions of ZD&T 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 Enterprise Edition, for example, **adcd/may2017**.
4. Transfer the ADCD Enterprise Edition.XML file to the directory specified in step 3. The file describes the metadata for the Enterprise Edition.
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 Enterprise Edition will automatically locate any ADCD provided with version 11.0.0 or later versions of ZD&T on the **IMAGE from ADCD** page.

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

Configuring Emulator licenses

Learn how to configure the emulator licenses.

The emulator licenses page also requires administrator privileges.

The license servers used by each target ZD&T instance needs to be configured within the Enterprise Edition 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 **Settings > Emulator licenses**.
2. In the **Primary** field, enter the fully qualified host name of the system to run the primary Software-based License Server.
3. Optional: In the **Secondary** field, enter the fully qualified host name of the system to run the secondary Software-based License Server.
4. Optional: In the Token Servers table, enter the ports and fully qualified host names of the systems to serve Rational tokens.

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 and packaged together as a single deliverable for a target environment. ZD&T Enterprise Edition allows you to create application images from various sources:

- Existing IBM Z.
- Application Developers Controlled Distributions (ADCD). The z/OS Software distribution is bundled with the product.
- Existing ZD&T instances.

Creating an application image from an existing IBM Z

Learn how to create an application image from an existing IBM Z.

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

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

Creating application components from an existing IBM Z

Learn how to create an application component from an IBM Z.

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

You can create application components from four resources:

- IBM Z mainframe data sets
- IBM Z mainframe volumes
- IBM Z mainframe CICS
- IBM Z mainframe Db2
- IBM Application Discovery projects

Creating application components from IBM Z mainframe data sets

Learn how to create an application component from IBM Z mainframe data sets.

If you create application components from IBM Z mainframe data sets, DASDVOL READ access is sufficient. If the DASDVOL READ access is not granted, the access in SAF class DATASET to individual data sets and catalogs is checked. Retrieval of a data set requires READ access to that data set. If the master catalog must be retrieved, the ALTER access is needed.

To create a component from IBM Z mainframe data sets, you can input a filter string that is passed to z/OS via the REST API. Then, you can get a list of cataloged data sets that match the filter.

The filter string that is used to search data set 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 Enterprise Edition needs wildcards at the end of the filter string to indicate that the filter string will match more characters. For example, 'IBMUSER' in ISPF means 'IBMUSER.**' in Enterprise Edition.

For the wildcards definition, see the following table:

Wildcards	Definition
%	Any single character
*	Any number of characters within a single qualifier. You 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 data sets unconsciously, which will be slow. But if you really want to retrieve all data sets, you can use '**.*.**'.
- The restriction '*' is set to help you avoid getting the data sets with only one qualifier when you try to retrieve all data sets. But if you really want to retrieve data sets with only one qualifier, you can use '*.'.

Before you create application components from an existing IBM Z, you need to grant READ access to the SAF profile in class DASDVOL for the volumes that hold the data sets to be extracted to the user ID that runs Enterprise Edition, or grant read access to the data sets to be extracted to the user ID that runs Enterprise Edition.

To create application components from IBM Z data sets, complete the following steps:

1. Click **IBM Z mainframe DATA SETS** on the **QUICK START** page, or click the left-top button to navigate to **Create - Component > Data set - z mainframe**.
2. Select an IBM Z mainframe that you configured, or add a mainframe system.
3. Enter the specified credentials.

Note: Contact your administrator if you do not have the information of the mainframe system.

4. Optional: Select **Remember my credential for this session** if you want.
5. Enter a component name.
6. Optional: Enter comments that can help you identify your components.
7. Enter a qualifier, and then select all data sets that are needed for the component.

- Optional: Select **Adjust advanced settings**, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.

Note: Concurrent extractions can extract multiple data sets in parallel, and improve the efficiency of creating application components with multiple data sets. However, if the number is too large, the system might be overloaded, and the overall process might be slow.

- Click **Create Component** directly. Or click **Schedule Component Extraction** to specify the date and time that you want to create the component.

When you create an application component, the information about IBM Z and the selected data sets will be saved and extracted to the image store system to be used later in an application image.

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

Creating application components from IBM Z mainframe volumes

Learn how to create an application component from IBM Z mainframe volumes.

If you create application components from IBM Z mainframe volumes, grant READ access to the SAF profile in class DASDVOL for the volumes that are extracted to the user ID that is used in the Enterprise Edition. For more information about DASD volume migration, see section 15.3.2 "RACF® requirements" in the [zPDT Guide and Reference](#).

To create application components from IBM Z mainframe volumes, follow these steps:

- Click **IBM Z mainframe VOLUMES** on the **QUICK START** page, or click the left-top button to navigate to **Create - Component > Volume - Z mainframe**.
- Select an IBM Z mainframe that you configured, or add a new mainframe system.
- Enter the specified credentials.

Note: Contact your administrator if you do not have the information of the mainframe system.

- Optional: Select **Remember my credential for this session** if you want.
- Enter a component name.
- Select if you want to contain system residence volumes.
- Optional: Enter comments that can help you identify your components.
- Select all volumes you need for the component. Click **Selected Volumes** to check the volumes that are selected for the component.
- Select **Adjust advanced settings**, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.

Note: Concurrent extractions can extract multiple volumes in parallel, and improve the efficiency of creating application components with multiple volumes. However, if the number is too large, the system might be overloaded, and the overall process might be slow.

- Click **Create Component** directly. Or click **Schedule Component Extraction** to specify the date and time that you want to create the component.

When you create an application component, the information about IBM Z 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 the step.

Creating an application component from an IBM Z CICS region

Learn how to create an application component from an IBM Z mainframe CICS region.

After you configured an IBM Z mainframe system with CICS regions, you can create application components from the CICS region.

To create an application component from an IBM Z CICS region, complete the following steps:

- Click **IBM Z mainframe CICS** on the **QUICK START** page, or click the left-top button to navigate to **Create - Component > CICS - Z mainframe**.

2. Select an IBM Z mainframe that you configured, or add a mainframe system.
3. Enter the specified credentials.

Note: Contact your administrator if you do not have the information of the mainframe system.
4. Optional: Select **Remember my credential for this session** if you want.
5. Select a configured CICS region or add a CICS region.
6. Enter a component name.
7. Optional: Enter comments that can help you identify your components.
8. Select the CICS groups from your IBM Z mainframe for the component.
9. Optional: Enter a qualifier, and select the data sets that contain the load and source modules that needs to be included in the component.
10. Optional: Select **Adjust advanced settings**, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.

Note: Concurrent extractions can extract multiple artifacts in parallel, and improve the efficiency of creating application components with multiple artifacts. However, if the number is too large, the system might be overloaded, and the overall process might be slow.
11. Click **Create Component**. Or click **Schedule Component Extraction** to specify the date and time when you want to create the component.

After you create an application component from IBM Z mainframe CICS, all extracted artifacts will be stored on the system you specified on the image storage.

Creating an application component from an IBM Z Db2 subsystem


Learn how to create an application component from an IBM Z Db2 subsystem.

Before you create application components from an IBM Z Db2 subsystem, you must ensure that you are aware of the prerequisites and known limitations for creating application components from an IBM Z Db2 subsystem. For more information, see [“Prerequisites” on page 37](#) and [“Limitations” on page 70](#).

To create an application component from an IBM Z Db2 subsystem, complete the following steps:

1. Click **IBM Z mainframe Db2** on the **QUICK START** page, or click the left-top button to navigate to **Create - Component > Db2 - Z mainframe**.
2. Select an IBM Z mainframe that you configured, or add a mainframe system.
3. Enter the specified credentials to access the mainframe system. Then, select **Remember my credential for this session** if you want.

Note: Contact your administrator if you do not have the information of the mainframe system.

4. Select a configured Db2 subsystem, or add a subsystem.
5. Enter a component name.
6. Optional: Enter comments that can help you identify your component.
7. Select the artifacts from the schemas that are displayed for the component. To show the artifacts with different types, such as tables, view or unsupported type, click the filter icon .

Note: If you select the check box of a schema, all the artifacts are automatically selected. If you want to select some artifacts of a schema, click the schema name, and select the artifacts that are needed. After you select the artifacts that are needed from the displayed schemas, you can click the **Selected artifacts** tab to verify. The tab can show all the artifacts that you selected from the schemas, and the artifacts names will be displayed after the schemas names with a dot, for example, DSNACC.UTLIST.

8. If you clear the option to use the IBM Db2 Administration Tool for z/OS when you configured a Db2 subsystem, you need to enter the required DDL file path.
9. Optional: Select **Adjust advanced settings**, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.

Note: Concurrent extractions can extract multiple artifacts in parallel, and improve the efficiency of creating application components with multiple artifacts. However, if the number is too large, the system might be overloaded, and the overall process might be slow.

10. Click **Create Component**. Or click **Schedule Component Extraction** to specify the date and time that you want to create the component.

Note: The option that automatically includes all tables with referential constraints is selected by default. This option can ensure that all the tables with referential constraints will be automatically validated and included when the data is extracted. If you clear the option, referential constraints will not be validated or included automatically, and the data extraction might not be completed. This option does not apply to view selection.

After you create an application component from an IBM Z Db2 subsystem, all the extracted artifacts will be stored on the system you specified on the image storage.

Note:

All database objects such as tables, indexes, and views are created on the target system based on the definitions that are obtained from the source system. Even if you select only one table for extraction, all the tables that are contained in the database will be created on the target system. Sequences are also created on the target system for each schema that is involved.

Db2 data is unloaded or loaded by using pipes (FIFO files in z/OS UNIX System Services), therefore any Db2 unloads or loads that failed cannot be restarted. If the failures occur during the Db2 unload or load process, the Db2 extraction programs will attempt to automatically clean up and stop the online utilities. If the user ID that runs the extraction does not have the authority to clean up these utilities, the manual cleanup action is required by the user ID that has the authority. The further extractions or deployments might fail until the cleanup is complete.

Due to the method that is used to extract and create Db2 objects, the following conditions prevent the creation of more than one Db2 component in a single ZD&T image.

- The components that contain an overlap of Db2 for z/OS tables.
- The components that have an overlap of Db2 for z/OS databases.

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:

- DASDVOL READ access is granted. If the DASDVOL READ access is not granted, the access in SAF class DATA SET to individual data sets and catalogs is checked. Retrieval of a data set requires READ access to that data set. If the master catalog must be retrieved, the ALTER access is needed.
- You are aware of all known limitations for creating application components from IBM Application Discovery projects. For more information, see [“Limitations”](#) on page 70.
- The connection is set up and tested through the IBM Application Discovery Configuration page. For more information, see [“Configuring IBM Application Discovery Server”](#) on page 77.

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 where all retrieved data sets are present.
4. Enter the port number to connect to Z.

Note: Currently, ZD&T Enterprise Edition only uses the SSH port to connect to Z. 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 data sets you need to transfer from your z System machine to the target ZD&T machine. Alternatively, you can search the terms you want.

Note: Currently, ZD&T Enterprise Edition only supports data sets whose host names are entered. The data sets 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**.

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 125 for reference.

Creating an application image from IBM Z components

Learn how to create an application image from IBM Z components.

An application image consists of a set of application components from the same IBM Z. After you create 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 might 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 COMPONENTS** on the **QUICK START** page, or click the left-top button to navigate to **Create - Image > Image - Z components**.
2. Log in the specified IBM Z.

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

3. Optional: Select **Remember my credential for this session**.
4. Enter an image name.
5. Optional: Enter comments that can help you identify your images.
6. Select the components and versions that are included in the application images.
7. Optional: Select **Adjust advanced settings**, and enter an integer in the box of **Number of cryptographic coprocessor** and **Domain**.

Note: If you want to run a zPDT that is configured to require a cryptographic coprocessor, enable this function. You need to ensure that the number of cryptographic coprocessors and domain that is defined in the image matches the configuration of your z/OS system. For more information, see the chapter 17 of [zPDT Guide and Reference](#).

8. Click **Create Image** to extract the volumes.

Creating an application image from ADCD package

Learn how to create an application image from ADCD.

ZD&T Enterprise Edition supports creating an application image by selecting the 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 or V12, and the required z/OS 2.3 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 > Image -ADCD**.
2. Select an ADCD version that is configured.
3. Enter an image name.
4. Optional: Enter comments that can help you identify your images.
5. Select all the components from ADCD and the extracted IBM Z components that you want to include in the application image.

Note:

- To include a CICS or Db2 component that was extracted from other z/OS system, the ADCD image you create must include one CICS runtime or Db2 runtime component from the selected ADCD package.
- The CICS components and data set components can be only included in the ADCD image of z/OS 2.3 May 2018 Edition or later versions.
- The Db2 components can be only included in the ADCD image of z/OS 2.3 May 2019 Edition or later versions.

Restriction: Only one Db2 runtime can be included in an ADCD image that contains a Db2 component. If you select more than one Db2 runtime in the ADCD packages first, the check box of the extracted Db2 components will be disabled. If select more than one Db2 component first, only one Db2 runtime in the ADCD packages can be selected.

6. Optional: Select **Adjust advanced settings**, and enter an integer in the box of **Number of cryptographic coprocessor**.

Note: If you want to run a zPDT that is configured to require a cryptographic coprocessor, enable this function. You need to ensure that the number of cryptographic coprocessors that is defined in the image matches the configuration of your z/OS system. For more information, see the chapter 17 of [zPDT Guide and Reference](#).

7. Click **Create Image**.

Creating an application image from an existing ZD&T

Learn how to create an application image from an existing ZD&T.

ZD&T Enterprise Edition also allows you to import an application image from an existing ZD&T.

To import an application image from an existing ZD&T, 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 the host name of ZD&T system to import application image, and login with the credential or certificate. Then, click **Next**.
3. Specify the location of the devmap file.
4. Select if you want to compress the contents that is extracted from the source system. If you select **Allow compression**, specify the location on the system that has adequate space to store the compressed contents. Then, click **Complete**.
5. Enter an image name.
6. Specify the required default IPL command for the z/OS that is included in the application image.
7. Click **Import Image**.

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

After you create application images, ZD&T Enterprise Edition will automatically set up an entire ZD&T for development and testing when it is needed.

The deployment options in the Enterprise Edition will install and configure the IBM ZD&T emulator on the selected target environment, which is followed by the initial program load of z/OS that is contained in the selected application image.

Note: If you create an ADCD application image, after the deployment is successfully complete, you need to log in z/OS, and change the expired password of the user ID 'IBMUSER'. If the ADCD application image contains extracted IBM Z data set components, and the process to restore the data sets fails during the deployment, set the password of the user ID 'IBMUSER' to be 'SYS1'.

Pre-deployment validation

Before you deploy application images to a Linux target environment, you must ensure that the following items are valid. Otherwise, when you deploy application images, the pre-deployment validation function will automatically check the items, and some error messages may occur.

Note: The pre-deployment validation is only applicable for the Linux target environments.

1. Linux distribution

Only RHEL and Ubuntu operating systems are officially supported. CentOS, SLES operating systems are unofficially supported.

2. CPU architecture

Only x86_64 is supported.

3. CPs and RAM

The number of central processors (CPs) must be not longer than 8 characters. And at least 2G random access memory (RAM) size is required.

4. Emulator status

The emulator cannot be running.

5. License server connectivity

- If you use a Software-based License Server, the port number must be 1947.
- If you use a Hardware-based License Server, the port number must be 9450.

6. Image storage server connectivity

- If you use FTP as the image storage server, the port number must be 21.
- If you use SFTP as the image storage server, the port number must be 22.

7. User and group configurations

If you did not select to automatically configure user and group when you configured the Linux target environments, the pre-deployment validation function will check the users and group configurations by using `zdt_config_user_byRoot.sh`.

• Group

The function will check whether the group 'zpdt' exists.

• Users

- If you deploy application images by using the root user ID, this function will check whether the user ID 'ibmsys1' exists, and whether 'ibmsys1' belongs to the group 'zpdt'.
- If you deploy application images by using a non-root user ID, this function will check whether the user ID is assigned to the group 'zpdt'.

8. Dependent libraries

- If you did not select to install all dependencies when you ran the Enterprise Edition installer, this function will check the following dependent libraries by using `zdt_install_dependencies_byRoot.sh`.

- REHL operating system

```
iptables
ftp
libstdc++.i686
perl
zip
unzip
gzip
```

- Ubuntu operating system

```
iptables
libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 lib32stdc++6
ftp
perl
zip
unzip
gzip
bc
libasound2
```

9. Permission on the deployment directory

Check whether the user ID has the write access to the deployment directory that was specified.

10. Available disk space on the deployment directory

Check whether the available disk space of the deployment directory is adequate. To deploy application images successfully, a full application size and an extra space of the maximum uncompressed volume size are needed.

11. Privilege commands permission

If you use a non-root user ID to deploy application images, this function will check whether the user ID has the permission to do the following actions.

a. Change the ownership of the files:

```
[deployment directory]/zdt/zdtInstall/zdt_install_product_byRoot.sh
If allow install dependencies
[deployment directory]/zdt/zdtInstall/zdt_install_dependencies_byRoot.sh (optional)
if allow configure user
[deployment directory]/zdt/zdtInstall/zdt_config_user_byRoot.sh (optional)
if allow configure network
[deployment directory]/zdt/zdtInstall/zdt_config_network_byRoot.sh (optional)
[deployment directory]/zdt/zdtInstall/zdt_config_tcpports (optional)
[deployment directory]/zdt/zdtInstall/zdt_config_udpports (optional)
if allow clean up when withdraw an image
[deployment directory]/zdt/zdtInstall/zdt_cleanup_byRoot.sh (optional)
```

b. Run the following scripts

```
[deployment directory]/zdt/zdtInstall/zdt_install_product_byRoot.sh
If allow install dependencies
[deployment directory]/zdt/zdtInstall/zdt_install_dependencies_byRoot.sh (optional)
if allow configure user
[deployment directory]/zdt/zdtInstall/zdt_config_user_byRoot.sh (optional)
if allow configure network
[deployment directory]/zdt/zdtInstall/zdt_config_network_byRoot.sh (optional)
[deployment directory]/zdt/zdtInstall/zdt_config_tcpports (optional)
[deployment directory]/zdt/zdtInstall/zdt_config_udpports (optional)
if allow clean up when withdraw an image
[deployment directory]/zdt/zdtInstall/zdt_cleanup_byRoot.sh (optional)
```

Manually issue IPL commands

When you deploy the created application images, if you clear the option to issue the IPL command manually, complete the following step after the deployment.

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 account that you used to do the deployment.

Note: If you choose a Linux target environment, and deploy application images by using the root user ID, login with the user ID **ibmsys1**. If you choose a OpenStack cloud environment, login with your default cloud-init user ID. The private key and passphrase can be downloaded from monitor page.

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/zdt/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.

Deploying application images to a Linux target environment

Learn about the steps to deploy application images to a Linux target environment.

To deploy an application image to a Linux target environment, complete the following steps:

1. Click **DEPLOY** on the **QUICK START** page, or click the left-top button to navigate to **Deploy**.
2. Select an application image that you want to deploy.
3. Select a target environment that the application image is deployed to. Then, click **Deploy Image**.
4. Select a version of the application image that you want to deploy.
5. Provide the required credentials of the target environment in one of the following ways.

- Click **Credentials**, and supply the user credentials that can access the target environment.

Note: If you deploy application images by using a non-root user ID, make sure that the user ID has sudo access. For more information, see [Sudo access configuration](#).

- Click **Certificate**, and supply the user ID along with your OpenSSH private key certificate file. If your key file requires a passphrase, enter it.

Note: You need to add your public key to the `authorized_keys` file of the user ID in the target environment: `/[home directory]/.ssh/authorized_keys`.

6. Provide the deployment directory.

The deployment directory is a directory where ZD&T Enterprise Edition stores installation files and volume files during the deployment process. For root users, the default deployment directory is `/home/ibmsys1`; for non-root users, the default deployment directory is `/home/[user ID]`.

Note: ZD&T Enterprise Edition will create a subdirectory that is named 'zdt' inside the deployment directory, and a couple of subdirectories inside 'zdt'. Make sure that the current user ID has the permission to do that.

After you click **Next**, the pre-deployment validation step will automatically check whether the target environment meets the requirements of a deployment. For more information about the items that will be checked, see items 1 - 8 in [“Pre-deployment validation”](#) on page 86.

7. Click **Next** if you deploy an application image that contains data set components. Then, enter the credentials to access the IBM Z mainframe.

The credentials are needed because a more flexible and efficient way is used to restore data sets. ZD&T Enterprise Edition needs to log in to the target environment to run the job or program.

Note: If you deploy a z/OS application image that requires the credentials, you must enter the initial password of the target environment. However, if you deploy an ADCD application image that requires the credentials, you can set an initial password instead of providing the default password.

8. Configure the deployment in one of the following ways.

- Clear **Adjust advanced default settings**, and click **Complete** to start the deployment.
- Select **Adjust advanced default settings**, and click **Next**. Then, complete the following optional steps:
 - a. Configure the central processors (CPs), integrated information processors (zIIPs), and available system memory size that are allocated to the instance.

Note:

 - The total quantity of CPs and zIIPs cannot be more than 8, and the quantity of zIIPs cannot be greater than the quantity of CPs.
 - To ensure the emulator performs well, the total quantity of CPs and zIIPs should be less than the number of real cores on the target environment.
 - b. Override the IPL command if you want. Specify the attributes of the device address, IODF address, and LOADxx suffix. Then, click **Complete**.

After you click **Complete**, the pre-deployment validation will automatically check the capability of the deployment directory, and whether the user ID has the permission to the deployment directory and privilege command. For more information, see items 9 - 11 in [“Pre-deployment validation” on page 86](#).

Deploying application images to an OpenStack cloud

Learn about the steps to deploy application images to an OpenStack cloud, or resume and update a deployment. The cloud target environments can be your own on-premises physical machines with your own personal cloud, or the ones from an IBM-managed cloud.

To deploy application images to an OpenStack cloud, which includes provision a new instance and deploy application images to the instance, 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. Then, click **Deploy Image**.
4. Select a version of the application that you want to deploy, and enter a descriptive deployment label that is used to help recognize the target environment within this application image.

Note: The deployment label must be unique, and use a maximum of 255 characters.

5. Choose the level of your authentication, Domain or Project, and enter the credential of your cloud account.
6. Click **Next** if you deploy an application image that contains data set components. Then, enter the credentials to access the IBM Z mainframe.

The credentials are needed because a more flexible and efficient way is used to restore data sets. The ZD&T Enterprise Edition needs to log in to the target environment to run the job or program.

Note: If you deploy a z/OS application image that requires the credentials, you must enter the initial password of the target environment. However, if you deploy an ADCD application image that requires the credentials, you can set the initial password instead of providing the default password.

7. Configure the deployment in one of the following ways.

- Clear **Adjust advanced default settings**, and click **Complete** to start the deployment.
- Select **Adjust advanced default settings**, and click **Next**. Then, complete the following optional steps:

- a. Configure the central processors (CPs), integrated information processors (zIIPs), and available system memory size that are allocated to the instance.

Note:

- The total quantity of CPs and zIIPs cannot be more than 8, and the quantity of zIIPs cannot be greater than the quantity of CPs.
 - To ensure the emulator performs well, the total quantity of CPs and zIIPs should be less than the number of real cores on the target environment.
- b. Override the IPL command if you want. Specify the attributes of the device address, IODF address, and LOADxx suffix. Then, click **Complete**.

After you click **Complete**, the pre-deployment validation will automatically check the capability of the deployment directory, and whether the user ID has the permission to the deployment directory and privilege command. For more information, see the items 9 - 11 in [“Pre-deployment validation”](#) on page 86.

Resuming a failed deployment

If a deployment fails, you can resume the deployment from the **Deploy** or **Monitor and Manage** page.

Resuming a failed deployment from the Deploy page

To resume a deployment that fails from the **Deploy** page, complete the following steps:

1. Click **Resume deployment** tab.
2. Click **Retrieve logs** to check the reason that causes the failure.
3. Click **Resume image deploy**, and enter all required credentials. Then, click **Next**.
4. Configure the central processors (CPs) and system memory (GB). Then, click **Next**.
5. Configure the automatic IPL command, and click **Complete**.

Resuming a failed deployment from the Monitor and Manage page

To resume a deployment that fails from the **Monitor and Manage** page, complete the following steps:

1. Click the **Actions** icon, and select **Resume image deployment**.
2. Provide the required credentials.
3. Configure the central processors (CPs) and system memory (GB). Then, click **Next**.
4. Configure the automatic IPL command, and click **Complete**

Updating a deployment to a Linux target environment

After you successfully deploy an application image that has multiple versions to a Linux target environment, you can update the deployment from **Deploy** or **Monitor and Manage** page.

Updating a successful deployment from the Deploy page

To update a deployment from the **Deploy** page, complete the following steps:

1. Click **Update deployment** tab.
2. Click the **Update deployment** icon () next to the asset that you want to update.
3. Select a version of the application image that is contained in the deployment.

Note: The **Update deployment** supports updating the target environment with any other version of the application image that had been deployed to the target environment.

4. Complete steps 5-8 in [“Deploying application images to a Linux target environment”](#) on page 88.

Updating a successful deployment from the Monitor and Manage page

To update a deployment from the **Monitor and Manage** page, complete the following steps:

1. Click the **Actions** icon, and select **Update deployment**.
2. Select a version of the application image that is contained in the deployment.
3. Complete steps 5 - 8 in [“Deploying application images to a Linux target environment”](#) on page 88.

Note: To update an image that was deployed by deleting a data set, you need to have the 'tsocmd' access and RACF Alter access to the data set or profile.

Monitoring and managing all created assets and target environments

ZD&T Enterprise Edition provides a monitor page that you can monitor and manage all of the created assets and target environments on the page. Also, you can take some actions directly for the created assets and target environments.

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

Monitoring and managing all created application components

After you created an application component, you can use the monitor page to check the status of the extraction, and manage the created application components. Also, you can retrieve logs to check the details of the application components.

Editing created application components

After you created an application component, you can edit the contents of the component by completing the following steps:

1. Click the **Actions** icon, and select **Edit**.
2. Modify the volumes or data sets that you selected. Or, reselect the volumes or data sets for the application component.
3. Click **Update Component**. Or, click **Schedule Component Extraction** to specify the date and time when you want to update the component.

Note:

- An application component that is being extracted, scheduled, or deployed cannot be edited. However, you can create a new version or duplicate the application component to edit and work with it.
- Only the contents of application components can be edited. All other fields will be unavailable and cannot be edited, such as the source mainframe, component name and so on.
- To edit the name of an application component, click the **Rename** icon beside the component name, and change the name. Then, click **Save** button.

Deleting created application components

To delete an application component, click the **Actions** icon. Then, select **Delete Component**.

Note:

- Only the application components that were created from IBM Z can be deleted separately.
- An application component that is being created cannot be deleted.
- An application component that is used in an application image cannot be deleted. To delete an application component, you must delete all of the application images that use the components first.
- If you delete an application component and image, the volume files that are located in the image storage server will be deleted automatically.

Resuming a failed extraction of an application component

If the extraction of an application component failed, you can resume the extraction by completing the following steps:

1. Click the **Actions** icon, and select **Resume extraction**.
2. Enter the required credentials, and click **Next**.
3. Click **Complete**.

Canceling a scheduled extraction

If you want to cancel the extraction of an application component that is scheduled to be created, complete the following steps:

1. Click the **Actions** icon, and select **Cancel scheduled extraction**.
2. Click **Submit** to confirm cancellation.

Stopping the extraction of an application component

To stop an application component that is being extracted, complete the following steps:

1. Click the **Actions** icon, and select **Stop extraction**.
2. Click **Terminate** to confirm it.

Creating new versions of an application component

When you create an application component, the version number always starts at 1, and increases by 1 when you create a new version. If you want to add new contents or refresh contents but do not want to modify the existing component, you can create a new version by completing the following steps:

1. Click the **Actions** icon, and select **New version**.
2. Log in to the specified IBM Z mainframe.
3. Add new contents, or refresh the contents you selected for the new version.
4. Click **Create component**.

Note:

- When you create a new version of an application component, you can add, remove, or reselect the volumes or data sets for the new version.
- If you need to reselect data sets for a new version, click **All data sets**. Then, enter the data set qualifier.
- The **New version** button is available only for the newest version of an application component.

Duplicating a created application component

If you want to copy a created application component with a separate lifecycle that starts from version 1, complete the following steps:

1. Click the **Actions** icon, and select **Duplicate**.
2. Log in to the specified IBM Z mainframe.
3. Create a new component name.
4. Duplicate the same contents, or reselect the volumes or add data sets that you want for the duplicated application component.
5. Click **Create component**.

Migrating application components to the current version

If a data set component that were created from a previous version contains mini volumes, the migration icon will be displayed. You need to re-extract the contents of the component, otherwise the components cannot be deployed.

To migrate an application component, complete the following steps:

1. Click the **Actions** icon, and select **Perform extraction**.

2. Enter the credentials of the IBM Z mainframe that you create application component from.
3. Click **Complete**.

In a migration, the contents of an application component might be changed if the original contents that came from the mainframe system changed.

Retrieve logs

To check the details of the created application components, click the **Actions** icon. Then, select **Retrieve logs**.

Note: Only the application components that were created from IBM Z volumes or data sets supports editing, creating new versions, or duplicating functions. For the components that were created from IBM Z CICS regions and Db2 subsystems, these functions are not supported.

Monitoring and managing all created application images

After you create an application image, you can stop, resume, or delete the application image. Also, you can retrieve logs to check the details of the application image.

Editing created application images

After you created an application image, you can edit the components you selected for the image by completing the following steps:

1. Click the **Actions** icon, and select **Edit**.
2. Modify the components that you selected. Or, reselect the components for the application image.
3. Click **Update Image**.

Note:

- Only the application images that were created from IBM Z mainframe and ADCD can be edited.
- An application image that is being extracted, scheduled, or deployed cannot be edited. However, you can create a new version or duplicate the application image to edit and work with it.
- Only the components of application image can be edited. All other fields will be unavailable and cannot be edited, such as the source mainframe, image name and so on.
- To edit the name of an application image, click the **Rename** icon beside the image name, and change the name. Then, click **Save** button.

Deleting created application images

To delete an application image that was created, complete the following step:

1. Click the **Actions** icon. Then, select **Delete application image**.
2. Click **Delete**.

Note:

- An application image that is being extracted, scheduled, or deployed cannot be deleted. To delete an application image that has been deployed, you must remove the image from all target environments first.
- If you delete an application image that was created from IBM Z components or ADCD, the components that are used in the application image will not be deleted. However, if you delete an application image that was created from an exiting ZD&T, the components that are used in the application image will also be deleted.

Creating new versions of an application image

When you create an application image, the version number always starts at 1, and increases by 1 when you create a new version. If you want to add new components or refresh the components you selected

but do not want to modify the existing image, you can create a new version by completing the following steps:

1. Click the **Actions** icon, and select **New version**.
2. Log in to the specified IBM Z mainframe.
3. Add new components, or refresh the components you selected for the new version.
4. Click **Create image**.

Note:

- The **New version** button is available only for the newest version of an application image.
- When you create a new version of an application image, you can add, remove, or reselect the components for the new version.

Duplicating a created application image

When you want to copy a created application image with a separate lifecycle that starts from version 1, you can complete the following steps:

1. Click the **Actions** icon, and select **Duplicate**.
2. Log in to the specified IBM Z mainframe.
3. Create a new image name.
4. Duplicate the same components, or reselect the components that you want for the duplicated application image.
5. Click **Create image**.

Migrating application images to the current version

To ensure the application images that were created from IBM Z mainframe systems in a previous version can be used in the current version, a migration might be needed. You need to go to the **Monitor and Manage** page to verify if an application image shows the migration icon. If the icon exists, migrate the application images, otherwise the images cannot be deployed.

To migrate application images that can be used in the current version, complete the following steps:

1. Click the **Actions** icon, and select **Migrate image to current version**.
2. Log in to the specified IBM Z mainframe that you create application images from.
3. Click **Complete**.

The migration only collects the information about a list of offline direct access storage device (DASD) that we need from the mainframe systems, which will not change the contents of application images.

Retrieving logs

To check the details of the created application images, click the **Actions** icon. Then, select **Retrieve logs**.

Note:

- Only the application images that were created from IBM Z components and ADCD supports editing, creating new versions, or duplicating functions.
- When you edit an image, create a new version, or duplicate an image, you must select at least one component that contains system residence volume.

Monitoring the deployment to target environments

After you deployed an application image to a target environment, you can use the monitor page to check the status of the deployment. Also, you can remove or resume application images, retrieve logs or the private key, and start or stop the z/OS emulator.

Checking the status of the deployment process

After you deployed an application image, the monitor page shows the overall percentage of the deployment process. If you want to check the details of the deployment process, click the percentage. Then, a pop-up window is displayed to show the deployment status that includes the following information:

- Overall progress of the deployment with a specific percentage
- Current[®] operation
- Deployment detail that shows each particular step in the deployment with a specific percentage

Updating the emulator to the latest version

After an application image is deployed, the emulator version number displays under the deployment. If the emulator that was added in the previous release cannot be determined, the **Unknown version** displays. If an unknown or earlier version of emulator is installed, the link to update the emulator to the latest version will be available next to the emulator.

To update the emulator to the latest version, complete the following steps:

1. Click the link next to the earlier version number or **Unknown version**.
2. Check the new functions or improvements of the latest emulator.
3. Click **Update**.

After you click **Update**, the current emulator will be stopped and uninstalled. Then, the newest emulator will be installed and restarted with an IPL.

Removing deployed application images from cloud environments

To remove application images from cloud environments, complete the following steps:

1. Click the **Actions** icon, and select **De-provision system**.
2. Enter the required credentials, and click **De-provision system**.

Removing deployed application images from Linux environments

To remove application images from Linux environments, complete the following steps:

1. Click the **Actions** icon, and select **Remove image**.
2. Enter the required credentials, and click **Next**.
3. Select or clear the following options.
 - Remove ZD&T instances from the target system and revert network configurations.
 - Remove application images (volume files)

The options are selected by default. You can select to remove ZD&T instances and revert network configuration, but keep the application images in the deployment directory. Or, you can clear the option to remove ZD&T instances and revert network configuration, and the application images also cannot be removed.

4. Enter the required credentials, and click **Remove**.

Resuming a failed deployment

To resume a failed deployment of an application image, complete the following steps:

1. Click the **Actions** icon, and select **Resume image deployment**.
2. Provide the required credentials.
3. Configure the central processors (CPs) and system memory (GB). Then, click **Next**.
4. Configure the automatic IPL command, and click **Complete**

Updating a successful deployment

If a successful deployment contains an application image that has multiple versions, you can update the deployment by completing the following steps:

1. Click the **Actions** icon, and select **Update deployment**.
2. Select a version of the application image that is contained in the deployment.
3. Complete steps 5 - 8 in [“Deploying application images to a Linux target environment”](#) on page 88.

Note: When you update a deployed ADCD image that requires z/OS credentials, the check box on the **z/OS Information** page will confirm whether you logged in to the z/OS instance and changed the default IBMUSER password. If you select the check box, you need to provide the changed credentials. If you clear the check box, the back-end API will use the default password to authenticate, and you must specify the password to authenticate into the system after the update deployment is completed.

Starting the emulated z/OS

If the emulated z/OS is offline or stopped, start the emulated z/OS by completing the following steps:


1. Click the **Actions** icon, and select **Start emulated z/OS**.
2. Enter the required credentials, and start the emulated z/OS in one of the following steps.
 - Clear **Adjust advanced settings**, and click **Complete**.
 - Select **Adjust advanced settings**, and click **Next**. Then, enter the additional IPL parameters, and click **Complete**.

Stopping the emulated z/OS

To stop a running emulated z/OS, complete the following steps:

1. Click the **Actions** icon, and select **Stop emulated z/OS**.
2. Enter the required credentials, and click **Complete**.

Note:

- An application image that is being deployed cannot be removed or resumed. Also, you cannot start or stop the z/OS emulator currently.
- If the icon  shows next to a deployment, it indicates that some issues exist on the deployment, you need to expand the deployment for details.

Scheduling the images creation

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

ZD&T Enterprise Edition supports scheduling and specifying a date and time to create an application image from Z. When creating application images from an IBM Z, 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 that 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.

REST APIs User's Guide

ZD&T Enterprise Edition supports REST APIs usage. You can access all functions of Enterprise Edition without the web user interface, including creating, monitoring, and deploying application images.

Getting started with REST APIs

To get started with REST APIs, you need to find the specification of all REST APIs that are available in the ZD&T Enterprise Edition, and know the restrictions of REST APIs usage.

To find the complete specification of all the available REST APIs provided by ZD&T Enterprise Edition after you start your Enterprise Edition server, go to the following URL:

`https://<your-tools-server-host-name>:9443/zdttools/apis/explorer/`

Note: This document only provides the restrictions and some examples on how to use the REST APIs.

Error response

For the error codes that are returned from the response body, refer to [“ZD&T response error codes” on page 110](#) to get the corresponding error messages. The parameters returned from the response body are used to complete the arguments in the error messages.

Restrictions

When you use REST APIs, the following restrictions also exist.

- All REST APIs require the basic authentication.
- Any account that can authenticate with the web user interface can be used to call any of the REST APIs. Authorization to the REST APIs is determined based on the role of the user.
- The initial password of 'zdtadmin' is 'password'. To change the password, complete the following steps:
 1. Find the encryption key that is specified by `wlp.password.encrypted.key` in the `/opt/ibm/zDT/Liberty/usr/servers/zDTServer/bootstrap.properties`.
 2. Run the following command where you installed ZD&T Enterprise Edition.

```
/opt/ibm/zDT/Liberty/bin/securityUtility encode --encoding=aes --  
key=<encryption_key_found_above> <new password value>
```

After you run the command, an encoded string is created.

3. Put the encoded string into the `server.xml` file that is at `/opt/ibm/zDT/Liberty/usr/servers/zDTServer/server.xml`. Then, replace the text in bold with the new encoded password value.

```
<!-- user authentication -->  
<basicRegistry id="zDT" realm="zDT">  
<user name="zdtadmin"  
password="{aes}AM1LZsnwLRNsVtYAIwqhVD09/RL+NgYthDZXZhQgARtB" />  
</basicRegistry>
```

Adding REST API through Zowe API Gateway

ZD&T Enterprise Edition installer contains a YAML file that can be used to add ZD&T REST APIs through Zowe API Gateway.

To add ZD&T REST APIs through Zowe API Gateway, complete the following steps.

1. Run the installer `ZDT_Install_EE_V12.0.4.0.tgz` to install ZD&T Enterprise Edition, and download the YAML file that is located at the directory `/opt/ibm/zDT/zowe`.
2. Obtain the IP and port of your IBM Z Development and Test Environment web server.
3. Update the YAML file to replace the pattern `{zdt tools machine IP};{zdt tools machine port}`.
4. Copy your YAML file to the `/${zoweRuntime}/api-mediation/api-defs` directory.
5. Restart Zowe if it is running.

For more information, see [Add a definition in the API Mediation Layer in the Zowe runtime](#).

REST API typical examples

To use REST API to access all functions of Enterprise Edition, use your own setup to replace the variables.

For more descriptions and available values of each field, refer to the REST API specification at <https://your-tools-server-host-name:9443/zdttools/apis/explorer/>.

Initial configuration

To do the initial configuration, you can use 'curl' to call REST APIs.

About this task

To do the initial configuration by using 'curl' to call REST APIs, and you need to use your own setup variables to replace the following *italic* text. For more information on the descriptions and available values of each field, refer to the REST API specification at the following URL:

<https://<your-tools-server-host-name>:9443/zdttools/apis/explorer/>

Procedure

1. Configure the connection to the image storage server.

```
curl -k -u ztdadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_ftp_storage_server_host_name", "storageSystemType": "FTP", "baseDirectory": "/data/zdttools", "port": 21, "username": "zdttools", "password": "zdttoolspw" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/imageStoreServices/FTP'
```

2. Configure the connection to the target environments if you want to deploy images to Linux machines.

```
ccurl -k -u ztdadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_target_system_host_name", "sshPort": 22, "label": "zdtForTestingCics", "installOSPackages": true, "concurrentVolumeTransferNumber": 3, "configNetwork": true, "configUser": true, "syntax": "sudo ${command}", "syntaxForValidation": "sudo -n -l", "networkInterface": "eth1"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystem'
```

3. Configure connection to the cloud platform if you want to deploy images to an OpenStack cloud.

```
curl -k -u ztdadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "endpoint": "mytarget.domain.com:5000", "domain": "ibm", "project": "myProject", "floatingip": true, "snapshot": true, "type": "OpenStack", "network": "your-network-uuid-in-cloud", "flavor": "your-flavor-id-in-cloud", "serverImage": "your-serverimage-uuid-in-cloud", "label": "My Cloud", "installOSPackages": true, "cloudUser": "your-cloud-user-name", "concurrentVolumeTransferNumber": 3, "configNetwork": true, "configUser": true, "syntax": "sudo ${command}", "syntaxForValidation": "sudo -n -l", "networkInterface": "eth1", "availabilityZone": "nova", "floatingIpPoolUuid": "your-cloud-floating-ip-pool-uuid"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/cloudPlatform'
```

4. Configure the connection to the emulator licenses. The following examples show the configuration of the Software-based License Server and Rational Tokens Server.

```
curl -k -u ztdadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "softwarePrimary":{"hostname": "your_license_server_host_name"}, "softwareSecondary":{"hostname": ""}}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/licenseServices/licensing/software'
```

```
curl -k -u ztdadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_rational_token_server_host_name", "port": 27000}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/licenseServices/licensing/token'
```

5. Configure the connection to z/OS.

This step is required only when you want to extract the data from z/OS systems.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/sourceEnvServices/sourceSystem'
```

6. Configure the connection to IBM Application Discovery server.

This step is required only when you want to import the data set information from the projects that are defined in IBM Application Discovery.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_AD_host_name", "port": 8090, "username": "aduser", "password": "adpw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/adServices/AD'
```

Creating application components

If you want to extract data from z/OS Systems, you must create application components before you create application images.

Creating volumes components from z/OS system

To create volumes components from z/OS system, you need to call the REST APIs to authenticate the SSH connection to the z/OS system, obtain the information on all volumes for the z/OS system, and create the components by including the volumes that are returned from the call.

Procedure

1. Call the REST API to authenticate the SSH connection to the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22, "username": "zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

2. Call the REST APIs to get the information on all volumes for a specific z/OS system. Put the token from above call into the authorization header.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "VVNFUjM1NTE4MTkwMzg4MzI=", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/zServices/listZosVolumes?hostName=YOUR_Z_OS_HOST_NAME&sshPort=22'
```

The following response is displayed.

```
[{"id":0, "name": "A27AAA", "extractionTime": "1960-01-01 23:03:20", "deviceNumber": "048C", "deviceType": "3390", "progress": 0, "compressed": 0, "size": 1892010960}, {"id":0, "name": "A27CCC", "extractionTime": "1960-01-01 23:03:20", "deviceNumber": "04BE", "deviceType": "3390", "progress": 0, "compressed": 0, "size": 2838016440}, {"id":0, "name": "A45SYS", "extractionTime": "1960-01-01 23:03:20", "deviceNumber": "0492", "deviceType": "3390", "progress": 0, "compressed": 0, "size": 8514049320}, ...]
```

3. Create the component by including some of the volumes that are returned from above call.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "name": "testComp-fromLinux", "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22 }, "systemType": "COMP_SYSTEM_TYPE_ZSYSTEM_ENV", "description": "comp_desc", "volumes": [ { "name": "A27AAA" }, { "name": "MVS220" } ], "schedules": { "originalScheduleTime": "2018-01-06 13:01:00.0", "originalScheduleTimeZoneId": "CTT" }, "concurrentExtractionNumber": 3, "ftpSystem": { "hostname": "your.image.storage.com" } }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/v2/zosVolumeComponent'
```

Creating data set components from z/OS system

Learn about the steps to create application components from IBM Z data set by using REST API.

Procedure

1. Call the REST API to authenticate the SSH connection to the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22, "username": "zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVNFUJM1NTE40DczMjY3NjU=","port":22}
```

2. Call the REST API to get the information on data sets that match a specific filter pattern from a specific z/OS system. Put the token from above call into the authorization header.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "cm9vdC0xMzZM5NzAzMjQ2", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/zServices/listZosDatasets?hostName=YOUR_Z_OS_HOST_NAME&filter=SYS1.*&sshPort=22'
```

The following response is displayed.

```
[{"id":0,"name":"SYS1.A39AAA.LPALIB","zosType":"Non-VSAM"}, {"id":0,"name":"SYS1.A45.LINKLIB","zosType":"Non-VSAM"}, {"id":0,"name":"SYS1.A45AAA.LINKLIB","zosType":"Non-VSAM"}, {"id":0,"name":"SYS1.ADFMAC1","zosType":"Non-VSAM"},...]
```

3. Create the component by including some of the data sets that are returned from the above call. Put the token from the first call into the authorization header.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "dXNlcjM1NTg5OTg4OTgwNA==", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' -d '{ "name": "testDsComp-fromLinux", "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22 }, "systemType": "COMP_SYSTEM_TYPE_ZSYSTEM_ENV", "description": "ds_comp", "dataSets": [ { "name": "SYS1.A39AAA.LPALIB" }, { "name": "SYS1.AFOMMOD1" }, { "name": "SYS1.A45AAA.LINKLIB" } ], "schedules": { "originalScheduleTime": "2018-01-06 13:01:00.0", "originalScheduleTimezoneId": "CTT" }, "concurrentExtractionNumber": 3, "ftpSystem": { "hostname": "your.image.storage.com" } }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/zosDatasetComponent'
```

Creating CICS components from z/OS system

Learn about the steps to create CICS components from z/OS system by using REST API.

Procedure

1. Configure the CICS to add the information of any CICS regions that you want to create components from by using Enterprise Edition web UI. For more information, see [“Configuring IBM Z mainframe systems”](#) on page 78.
2. Call the REST API to authenticate the SSH connection to the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22, "username": "zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVNFUJM1NTE40DczMjY3NjU=","port":22}
```


3. Create the component by specifying the GROUPS from the CICS CSD, the user load library data sets, and the source module data sets that will be included in the component. Put the token from the first call into the authorization header.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "dXNlcjM1NTg5OTg4OTgwNA==", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' -d '{ "name": "testCicsComp", "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22 }, "version": 1, "systemType": "COMP_SYSTEM_TYPE_ZSYSTEM_USER", "description": "ds_comp", "schedules": { "originalScheduleTime": "2019-01-06 13:01:00.0", "originalScheduleTimezoneId": "CTT" }, "concurrentExtractionNumber": 1, "ftpSystem": { "hostname": "your.image.storage.com" }, "regionName": "CICSTS54", "selectedGroups": [ { "groupName": "GENASAF", "listName": "GENALIST" }, { "groupName": "GENASAD", "listName": "GENALIST" } ], "loadModuleDataSets": [ { "name": "YOUR.APP1.LOADLIB" } ], "sourceModuleDataSets": [ { "name": "YOUR.APP1.SOURCE1.C" } ] }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/cicsComponent'
```

Creating Db2 components from z/OS system

Learn about the steps to create Db2 components from z/OS system by using the REST API.

Procedure

1. Call the REST API to authenticate the SSH connection to the target environment.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_target_system_host_name", "port": 22, "username": "root", "password": "rootpw" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"root","password":"doc1","token":"cm9vdC02NjQ3ODgxMTY=", "port":22}
```

2. Obtain the list of Db2 subsystems from the z/OS system by passing the hostname and port of the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "cm9vdC0xMzM5NzAzMjQ2", "hostname": "your_target_system_host_name", "port": 22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/sourceEnvServices/db2/subsystem?hostName= your_zsystem_host_name &port=2022'
```

After the request is complete, the following JSON response with a list of subsystems is displayed.

```
[{"id":2,"subsystemID":"DBCg","subsystemLocation":"DALLASC","sdsnext":"SAMPLE.SDSNEXIT","sdsnload":"SAMPLE.SDSNLOAD","runLib":null,"db2Admin":false,"db2AdminProgram":"ADB2GEN","db2AdminPlan":"ADB2GEN","db2AdminProgramLib":"SAMPLE.SADBLLIB","creationTime":"2019-07-13 01:26:01","user":{"id":1,"name":"zdtadmin","loginTime":1565621199429,"roles":[{"id":1,"name":"Administrator","creationTime":1562281034407}]},"migrate":0,"creationTime":1562281044010}}
```

3. Obtain the list of schemas by using the subsystem ID with the hostname, and port of the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "cm9vdC0xMzM5NzAzMjQ2", "hostname": "your_target_system_host_name", "port": 22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/db2/v1/schemas?hostName= your_zsystem_host_name&sshPort=2022&subsystemName=your_subsystem_id'
```

After the request is complete, the list of schema names is displayed.

```
["ADB","CLARKG","DSN8110","DSN81110","DSN811SA","DSN8BQRY","DSNRGCOL","IBMUSER","Q","RUNSTATS","SYSIBM","SYSIBMTS","SYSTOOLS"]
```

4. Obtain the list of artifacts by passing the schema name with the hostname, port, and subsystem ID of the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token":"cm9vdC0xMzM5NzAzMjQ2","hostname":" your_target_system_host_name","port":22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/db2/v1/schemas/your_schema_name?hostName= your_zsystem_host_name&sshPort=2022&subsystemName= your_subsystem_id'
```

After the request is complete, the list of Db2 tables is displayed.

```
[{"id":0,"name":"PLAN_TABLE","tableType":"TABLE","schema":{"id":0,"name":"CLARKG"},"database":{"id":0,"name":"DSN00022"},"tableSpaceSize":720,"allocatedTableSpaceSize":0,"unsupportReason":null,"tableSpaceName":"PLANRTAB","creationTime":null,"subSystem":null,"addedByUser":false}]
```

5. Create a Db2 component by using the obtained list of the selected Db2 tables, subsystem ID, the hostname, and port of the z/OS system. If you have a DDL file, you must specify the path to the DDL file in the Db2DDL parameter.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token":"cm9vdC0xMzM5NzAzMjQ2","hostname":" your_target_system_host_name","port":22}' -d '{"name":"test","zSystem":{"hostname":"your_zsystem_host_name","port":2022},"description":"test","ftpSystem":{"hostname":"your_ftp_system_host_name"},"concurrentExtractionNumber":"1","subsystemName":"your_subsystem_id","selectedTables":[{"name":"PLAN_TABLE","tableType":"TABLE","schemaName":"your_db2_schema_name","databaseName":"your_db2_database_name"}],"integrity":true,"db2DDL":""}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/v1/db2'
```

Creating data set components from z/OS for projects defined in AD

Learn about the steps to create application components from IBM Application Discovery by using REST API.

Procedure

1. Call the REST API to authenticate the SSH connection to the z/OS system.
2. Call the REST API to get the list of data sets for a specific project that is defined in AD from a specific z/OS system.
3. Create the component by including the data sets returned from above call. Put the token from the first call into the authorization header.

Creating application images

After the application images are created, you can store the application image data in the specified image storage server.

Creating application images from z/OS volumes or data sets

Learn about the steps to create application images from z/OS volumes or data sets.

Procedure

1. Call the REST API to authenticate the ssh connection to the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"hostname":"YOUR_Z_OS_HOST_NAME","port":22,"username":"ZOSUSER","password":"ZOSPW"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVNFUjM1NTE4ODczMjY3NjU=","port":22}
```

2. Create the application images by including some of the components that you created from above steps. Put the token from the first call into the authorization header.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization:
```

```
{
  "token": "dXNlcjM1NTg5OTg4OTgwNA==",
  "hostname": "YOUR_Z_OS_HOST_NAME",
  "port": 22
} -d
{"name": "testApp1", "zSystem": {
  "hostname": "YOUR_Z_OS_HOST_NAME",
  "port": 22,
  "version": 1,
  "description": "comment",
  "components": [
    {
      "name": "name of component",
      "zSystem": {
        "hostname": "myzos.domain.com",
        "port": 22,
        "version": 1
      }
    }
  ],
  "sysResComponent": {
    "name": "name of the system residence volume component",
    "zSystem": {
      "hostname": "myzos.domain.com",
      "port": 22,
      "version": 1
    }
  }
} 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/v2/zosApplicationImage'
```

If an error code is displayed, for example:

```
{
  "code": 31326,
  "parameters": ["SYS1.A39AAA.LPALIB", "SYS1.AF0MMOD1"]
}
```

The error code '31326' means that "these data sets cannot be extracted because their containing volumes cannot be found: [dataset_names]". For more information about error code, see [“ZD&T response error codes”](#) on page 110.

- Optional: Try again when an error code is displayed at step 2.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "dXNlcjM1NTg5OTg4OTgwNA==", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22} -d {"name": "testApp1-fromLinux", "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22, "ftpSystem": { "hostname": "your_storage_server_host_name", "components": [ { "name": "testComp-fromLinux", "schedules": { "originalScheduleTime": "2018-01-06 13:01:00.0", "originalScheduleTimezoneId": "CTT" }, "concurrentExtractionNumber": 3 } } } } 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/zosApplicationImage'
```

Then, the generated information of application images is returned.

Creating an application image from ADCD package

Learn about the steps to create application images from ADCD package by using REST API.

Procedure

- Copy your ADCD package to the image storage server, create a subfolder under the baseDirectory/adcd, and place the package. This subfolder name, for example, may2017, is treated and used as the source system host name when you create application from this ADCD package and deploy this application image.
- Get the ADCD package information before you create any ADCD application for the first time.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/adcdServices/ADCDS'
```

The following response is displayed.

```
[{"id":1,"name":"z/OS V2.2 May 2017","rsu":"1703"}]
```

- Get the component names included in the ADCD package, and provide the sample subfolder name 'may2017' as the source host name at the end.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/components/may2017'
```

The following response is displayed.

```
[ {
  "id": 8,
  "name": "Customer Information Control System (CICS) - 5.2, 5.3",
  "systemType": "COMP_SYSTEM_TYPE_ADCCD_USER",
  ...
}, {
  "id": 5,
  "name": "Db2 - 11, 12",
  "systemType": "COMP_SYSTEM_TYPE_ADCCD_USER",
  ...
}, {
  "id": 9,
  "name": "z/OS - 2.2",
  "systemType": "COMP_SYSTEM_TYPE_ADCCD_ENV",
  "componentType": "COMP_WITH_VOLUME",
  "zSystem": {
    "id": 101,
    "hostname": "may2017",
    "type": "SYSTEM_TYPE_ADCCD",
    "port": 22,
    "adcdPackage": {
      "id": 1,
      "name": "z/OS V2.2 May 2017",
      "rsu": "1703"
    }
  },
  "volumes": [
    {
      "id": 27,
      "name": "D2SYS1",
    }
  ]
}
...}]
```

4. Create an application image.

- **Creating an application image from ADCCD package**

If you want to create an application image from ADCCD package, select the components as found from the above call (provide the sample subfolder name 'may2017' as the source host name), for example:

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "name": "test-adcd-appl-from-linux",
  "zSystem": {
    "hostname": "may2017",
    "port": 22
  },
  "ftpSystem": {
    "hostname": "your_storage_server_host_name"
  },
  "components": [
    {
      "name": "z/OS - 2.2"
    }
  ]
}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/ADCCDApplicationImage'
```

Then, the generated information of application image is returned.

- **Creating an application image from ADCCD package and data sets from other z Systems**

If you want to create an application image from ADCCD package and data sets from other Z systems, select the components as found from the above call, and specify the data set components that were created from other Z systems, for example:

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "name": "test-adcd-appl-from-linux",
  "zSystem": {
    "hostname": "may2017",
    "port": 22
  },
  "ftpSystem": {
    "hostname": "your_storage_server_host_name"
  },
  "components": [
    {
      "name": "z/OS - 2.2"
    },
    {
      "name": "dataset-component1",
      "zSystem": {
        "hostname": "z-system1",
        "port": 22
      },
      "version": 1
    },
    {
      "name": "dataset-component2",
      "zSystem": {
        "hostname": "z-system2",
        "port": 2022
      },
      "version": 1
    }
  ]
}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/ADCCDApplicationImage'
```

- **Creating an application image from ADCCD package and CICS components**

If you want to create an application image from ADCCD package and CICS components, select the components as found from the above call, and specify the CICS components that were created from other Z systems, for example:

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{
  "name": "test-adcd-appl-from-linux",
  "zSystem": {
    "hostname": "may2017",
    "port": 22
  },
  "ftpSystem": {
    "hostname": "your_storage_server_host_name"
  },
  "components": [
    {
      "name": "z/OS - 2.2"
    },
    {
      "name": "dataset-component1",
      "zSystem": {
        "hostname": "z-system1",
        "port": 22
      },
      "version": 1
    },
    {
      "name": "dataset-component2",
      "zSystem": {
        "hostname": "z-system2",
        "port": 2022
      },
      "version": 1
    }
  ]
}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/ADCCDApplicationImage'
```


2. Run the following command to submit the request.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "dXNlcjM1NTg5OTg4OTgwNA==", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' -d '{"name": "name of the component", "zSystem": {"hostname": "myzos.domain.com", "port": 22}, "version": 1, "newInfo": {"name": "new name", "description": "new comment", "systemType": "COMP_SYSTEM_TYPE_ZSYSTEM_ENV", "concurrentExtractionNumber": 3}, "addItems": [{"volume to be added}], "removeItems": [{"volume to be removed}], "refreshItems": [{"volume to be refreshed}], "duplicate": false, "newVersion": false, "schedules": {"originalScheduleTime": "2018-01-06 13:01:00.0", "originalScheduleTimezoneId": "CTT"}}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/component'
```

Editing application images

After you create application images, you can edit the image, or create a new version or a duplicate of the image.

Procedure

1. Call the REST API to authenticate the ssh connection to the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"hostname": "YOUR_Z_OS_HOST_NAME", "port": 22, "username": "ZOSUSER", "password": "ZOSPW"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

The following response is displayed.

```
{"expiration": 1516202935577, "username": "ZOSUSER", "password": "ZOSPW", "token": "VVNFUjM1NTE4ODczMjY3NjU=", "port": 22}
```

2. Run the following command to submit the request.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "dXNlcjM1NTg5OTg4OTgwNA==", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' -d '{"name": "name of the application", "zSystem": {"hostname": "myzos.domain.com", "port": 22}, "version": 1, "newInfo": {"name": "new name", "description": "new comment"}, "addItems": [{"name": "component to be added", "zSystem": {"hostname": "myzos.domain.com", "port": 22}, "version": 1}], "removeItems": [{"name": "component to be removed", "zSystem": {"hostname": "myzos.domain.com", "port": 22}, "version": 1}], "duplicate": false, "newVersion": false, "sysResComponent": {"name": "name of the system residence volume component", "zSystem": {"hostname": "myzos.domain.com", "port": 22}, "version": 1}}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/application'
```

Deleting application images or components

About this task

To delete the created application images or components, complete the following steps.

Procedure

- To delete an application image, issue the following command.

```
curl -k -u zdtadmin:password -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"name": "testZdtAppl", "zSystem": {"hostname": "your_appl_source_host_name", "port": 22, "version": 1}}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/deleteApplication'
```

- To delete an application component, issue the following command.

```
curl -k -u zdtadmin:password -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"name": "testZosComp", "zSystem": {"hostname": "your_appl_source_host_name", "port": 22, "version": 1}}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/deleteComponent'
```

Deploying the application images to a Linux target environment

You can transfer application image data from the image storage system to a specific Linux target environment by authenticating with a password or an ssh certificate key file.

Procedure

1. Call the REST API to authenticate the SSH connection to the target environments by using a password or an ssh certificate key file.

- Authenticating the connection with a password

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_target_system_host_name", "port": 22, "username": "root", "password": "rootpw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

- Authenticating the connection with an ssh certificate key file

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: multipart/form-data' --header 'Accept: application/json' -F 'sshCredentials={ "hostname": "your_target_system_host_name", "port": 22, "username": "root", "password": "keyPassPhrase"}' -F 'keyFile=@/dir1/dir2/keyFile.txt' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSHkey'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"root","password":"doc1","token":"cm9vdC02NjQ3ODgxMTY=", "port":22}
```

2. Deploy the application images by specifying the application images and target environment you created. Put the token from the first call into the authorization header.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token":"cm9vdC0xMzZM5NzAzMjQ2","hostname":" your_target_system_host_name","port":22}' -d '{ "targetSystem": { "hostname": "your_target_system_host_name", "sshPort": 22, "cp": 3, "ram": 16106127360 , "deploymentDirectory": "/home/ibmsys1" }, "application": { "name": "testZdtAppl", "version": 1, "zSystem": { "hostname": "your_appl_source_host_name", "port": 22 }, "ftpSystem": { "hostname": "your_storage_server_host_name" } }, "doIPL": true, "iplDeviceAddress": "0A80", "iplIODFAddress": "0A81", "iplLoadSuffix": "WS", "targetZosUsername": "TSOUSR1", "targetZosPassword": "TSOPASS1" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/deployLinux'
```

3. Check the status of the deployment.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystem/your_target_system_host_name'
```

If the status is `"status": "DEPLOYED"`, the deployment process is complete.

If the deployment fails, you can obtain the deployment logs by completing the following steps:

- a. Enter the following command.

```
curl -k -u zdtadmin:password -O -X GET --header 'Accept: application/zip' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/zipAllLogs?targetHostname=your_target_system_host_name&sourceHostname=your_appl_source_host_name&sourcePort=22&applicationName=testZdtAppl'
```

Then, you can get a .zip file that contains all related logs.

- b. Extract the file to read the logs.

Deploying the application images to cloud platforms

You can transfer application image data from the image storage system to a specific cloud platform, or use the snapshot of an application image to create an ZD&T instance in the cloud platform.

Deploying to a new instance

Procedure

1. Call the REST API by specifying the application and the cloud platform that you created.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "deploy_label": "a_unique_label", "new_deploy": true, "application": { "name": "testZdtAppl", "version": 1, "zSystem": { "hostname": "your_appl_source_host_name", "port": 22 }, "ftpSystem": { "hostname": "your_storage_server_host_name" } }, "cloud_label": "cloud_platform_name", "create_user": "zdtadmin", "cloudUser": "user_to_login_cloud", "password": "cloud_user_password", "scope": "project", "cp": 3, "ram": 16106127360, "doIPL": true, "deploymentDirectory": "/zdttee", "iplDeviceAddress": "0A80", "iplIODFAddress": "0A81", "iplLoadSuffix": "WS", "targetZosUsername": "TSOUSR1", "targetZosPassword": "TSOPASS1" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/deployCloud'
```

2. Check the status of the deployment.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystems'
```

If the status is `"status": "DEPLOYED"`, the deployment process is complete.

If the deployment fails, you can obtain the deployment logs by completing the following steps:

- a. Enter the following command.

```
curl -k -u zdtadmin:password -O -X GET --header 'Accept: application/zip' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/zipAllLogs?targetHostname=your_target_system_host_name&sourceHostname=your_appl_source_host_name&sourcePort=22&applicationName=testZdtAppl'
```

You will get a zip file that contains all related logs.

- b. Unzip the file to read the logs.

Resuming a failed or interrupted deployment

Procedure

1. Call the REST API by specifying the application and the cloud platform that you created.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "deploy_label": "the_unique_label_specified_during_firsttime_deploy", "new_deploy": false, "application": { "name": "testZdtAppl", "version": 1, "zSystem": { "hostname": "your_appl_source_host_name", "port": 22 }, "ftpSystem": { "hostname": "your_storage_server_host_name" } }, "cloud_label": "cloud_platform_name", "create_user": "zdtadmin", "cloudUser": "user_to_login_cloud", "password": "cloud_user_password", "scope": "project", "cp": 3, "ram": 16106127360, "doIPL": true, "deploymentDirectory": "/zdttee", "iplDeviceAddress": "0A80", "iplIODFAddress": "0A81", "iplLoadSuffix": "WS", "targetZosUsername": "TSOUSR1", "targetZosPassword": "TSOPASS1" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/deployCloud'
```

2. Check the status of the deployment.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystems'
```

If the status is `"status": "DEPLOYED"`, the deployment process is complete.

If the deployment fails, you can obtain the deployment logs by completing the following steps:

- a. Enter the following command.


```
curl -k -u zdtadmin:password -O -X GET --header 'Accept: application/zip' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/zipAllLogs?targetHostname=your_target_system_host_name&sourceHostname=your_appl_source_host_name&sourcePort=22&applicationName=testZdtAppl'
```

You will get a zip file that contains all related logs.

- b. Unzip the file to read the logs.

Deprovisioning a cloud instance

Procedure

- Issue the following command.

```
curl -k -u zdtadmin:password -X DELETE --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "deploy_label": "the_unique_label_specified_during_deploy", "cloudUser": "user_to_login_cloud", "password": "cloud_user_password", "scope": "project" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/deprovisionCloudInstance'
```

Checking the status of a deployment on a target environment

About this task

After you deploy an application image, you can check the status of a deployment on the target environment by issuing the following command:

```
curl -k -u zdtadmin:password -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "targetHostname": "your_target_host_name", "sourceHostname": "application_source_host_name", "sourcePort": 22, "applicationName": "name_of_deployed_application", "applicationVersion": 1}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/Monitoring_services/getDeployStatus'
```

Updating an application image to a Linux target environment

After you deploy an application image successfully, you can update the deployment with a new version of the application image to a specific Linux target environment by authenticating with a password or an ssh certificate key file.

Procedure

1. Call the REST API to authenticate the SSH connection to the Linux target environment by using a password or an ssh certificate key file.

- Authenticating the connection with a password

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_target_system_host_name", "port": 22, "username": "root", "password": "rootpw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSH'
```

- Authenticating the connection with an ssh certificate key file

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: multipart/form-data' --header 'Accept: application/json' -F 'sshCredentials={ "hostname": "your_target_system_host_name", "port": 22, "username": "root", "password": "keyPassPhrase" }' -F 'keyFile=@/dir1/dir2/keyFile.txt' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/authServices/authenticateSSHkey'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"root","password":"doc1","token":"cm9vdC02NjQ3ODgxMTY=", "port":22}
```

2. Update the deployment by specifying the new version of the application image, the deployed application image, and target Linux environment. Put the token from the first call into the authorization header. 3 optional fields were added to specify the IPL parameters. 2 fields were added to specify the

z/OS user credential that is required to deploy an application with data set, you need to use the credential to log in the target z/OS.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "cm9vdC0xMz5NzAzMjQ2", "hostname": "your_target_system_host_name", "port": 22}' -d '{"targetSystem": {"hostname": "your_target_system_host_name", "sshPort": 22, "cp": 3, "ziip": 0, "ram": 5106127360, "deploymentDirectory": "/home/ibmsys1"}, "application": {"name": "your_application_name", "ftpSystem": {"hostname": "your_storage_system_host_name"}, "zSystem": {"hostname": "your_application_source_host_name", "port": 22}, "version": 2}, "doIPL": true, "targetZosUsername": "your_application_source_system_user_name", "targetZosPassword": "your_application_source_system_password", "iplDeviceAddress": "0A80", "iplIIOFAddress": "0A82", "iplLoadSuffix": "00", "deployedApplication": {"name": "your_deployed_application_source_host_name", "zSystem": {"hostname": "your_application_source_host_name", "port": 22}, "version": 1}, "updateEmulator": true, "usePassword": true }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/updateLinux'
```

3. Check the status of the deployment.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystem/your_target_system_host_name'
```

If the status is `"status": "DEPLOYED"`, the deployment process is complete.

If the deployment fails, you can obtain the deployment logs by completing the following steps:

- a. Enter the following command.

```
curl -k -u zdtadmin:password -O -X GET --header 'Accept: application/zip' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/zipAllLogs?targetHostname=your_target_system_host_name&sourceHostname=your_appl_source_host_name&sourcePort=22&applicationName=testZdtAppl'
```

Then, you can get a .zip file that contains all related logs.

- b. Extract the file to read the logs.

ZD&T response error codes

If an error occurs when you use REST APIs, refer to the response error codes for detailed description.

Table 20. Response error codes	
Error ID	Description
10001	The source system already exists.
10002	The source system does not exist.
10003	The target environment already exists.
10004	The target environment does not exist.
10005	The image storage server already exists.
10006	The image storage server does not exist.
10007	The Licensing system \${hostname} already exists.
10008	The Licensing system \${hostname} does not exist.
10009	The Token system already exists.
10010	The Token system does not exist.
10011	The IBM Application Discovery system already exists.
10012	The IBM Application Discovery system does not exist.
10013	The application image already exists.

Table 20. Response error codes (continued)

Error ID	Description
10014	The application image does not exist.
10015	The component already exists.
10016	The component does not exist.
10017	The volume already exists.
10018	The volume does not exist.
10019	The cloud platform already exists.
10020	The cloud platform does not exist.
10027	Target environment hostname <code>{hostname}</code> is not reachable.
10028	Create entity <code>{hostname}</code> failed. Reason: <code>{localizedMessage}</code>
10029	Retrieve entity <code>{hostname}</code> failed. Reason: <code>{errorMessage}</code>
10030	Update entity <code>{hostname}</code> failed. Reason: <code>{errorMessage}</code>
10031	Delete entity <code>{hostname+componentName}</code> failed. Reason: <code>{localizedMessage}</code>
10101	Failed to connect to the remote host <code>{sshHost}</code> on port <code>{sshPort}</code> . Reason: <code>{localizedMessage}</code>
10102	Token expired.
10200	Failed to connect to the server <code>{hostname}</code> on port <code>{port}</code> . Reason: <code>{localizedMessage}</code>
10201	Connection refused.
10202	Invalid user ID or password.
10203	Invalid base directory.
10204	Invalid hostname for server <code>{httpsURL}</code>
10205	Connection closed without indication.
10206	Failed to set the file transfer type.
10207	Failure when storing content on the image storage server.
10208	Failure when retrieving content from the image storage server.
10209	Failed to create directory <code>{pathElement+in+path / path}</code> on the image storage server.
10210	Unable to access the directory <code>{pathElement+in+path / baseDirectory}</code> on the image storage server.
10213	Internal Error.
10214	The old base directory is needed.
10215	This feature is supported only for SFTP storage system.
11001	Installation files download failed.
11003	Deploy request parameter not passed.
11007	Invalid Linux platform of target system. Supported platforms are: Ubuntu 16.04, RedHat 6.x, and RedHat 7.x

Table 20. Response error codes (continued)

Error ID	Description
11011	zPDT installation script error: Parameters invalid.
11012	zPDT installation script error: Primary licensing server is required.
11013	zPDT installation script error: zPDT package name is required.
11014	zPDT installation script error: zPDT installation failed.
11015	zPDT installation script error: Init gen2 failed.
11016	zPDT installation script error: Client Config failed.
11020	Deploy script error: Parameters invalid.
11021	Deploy script error: Devmap required.
11022	Deploy script error: Devmap does not exist.
11023	Deploy script error: Parameter loadparam required.
11024	Deploy script error: Parameter IODF address required.
11025	Deploy script error: Emulator start up message not found.
11026	Deploy script error: Parameter device number required.
11027	Deploy script error: Z is not on.
11028	You cannot deploy to the same system running IBM Z Development and Test Environment Enterprise Edition.
11029	Target server is not available for deploy.
11030	Failed to create directory on the target environment.
11031	Changing the owner of volume/devmap to be ibmsys1 on the target system failed.
11032	Copying script file to /home/ibmsys1 failed.
11033	IBM Z Development and Test environment was not started successfully.
11034	The IPL of z/OS running on IBM Z Development and Test Environment was not successful.
11035	Internal script error when installing ZD&T with missing path.
11036	Patch installation failed when installing zD&T.
11037	The file failed to be transferred to the target environment during deployment.
11100	Some volume image files failed deploying.
11101	Changing the owner of volume image files failed.
11102	Executing command \${comand} on target environment \${hostname} failed.
11103	Generating devmap failed.
11104	Changing owner of devmap failed.
11105	A failure occurred when trying to start the emulator.
11106	A failure occurred when trying to stop the emulator.
11107	A failure occurred when trying to IPL with the given parameters.
11108	A failure occurred when trying to clean the target system with the given parameters.

Table 20. Response error codes (continued)

Error ID	Description
11109	A failure occurred when trying to start an emulator on an undeployed target system.
11110	A failure occurred when trying to remove an image on the target system because the supplied credentials do not match the credentials used to start the deployment.
11111	A failure occurred when trying to obtain licensing information from the target system: \${hostname}.
12000	A Failure occurred reading the devmap file.
12001	A Failure occurred parsing the devmap file.
12002	A Failure occurred writing the devmap file.
12003	Create application failed.
12004	The devmap file does not exist in the specified location.
12005	You do not have the permissions to access the devmap.
12006	You do not have the permissions to access the volume file.
12007	The IPL command for importing a zD&T image was not provided.
20001	Authorization token has expired. Login again.
20002	Authorization token was not supplied.
20003	User already exists.
20004	User does not exist.
21001	Tools server has a network error.
21002	Failed executing ssh command on \${sshHost}.
21003	Failed executing sftp command on \${sshHost}.
21004	Remote command execution at \${sshHost} has been running for too long.
21005	Sending/receiving files to/from \${sshHost} failed when \${doing}
21006	The target environment TCP port routing rules must contain 2022>22.
21007	The target environment port routing rules have syntax error.
21008	The target environment internal IP address has syntax error.
30000	Source system hostname is required.
30001	Source system hostname is invalid.
30002	Application name is required.
30003	Application name is invalid.
30004	Authorization is required.
30005	Authorization is invalid.
30006	Application is required.
30007	Application is invalid.
30008	The application schedule date is invalid.
30009	The application schedule time is invalid.

Table 20. Response error codes (continued)

Error ID	Description
30010	The application schedule date is required.
30011	The application schedule time is required.
30012	The application schedule timezone is invalid.
30013	The application schedule timezone is required.
30014	You cannot choose more than one component that contains system residence volumes.
30020	Component name is required.
30021	Component name is invalid.
30022	Volumes are required.
30023	Volumes are invalid.
30024	A dataset name filter string is required.
30025	The data set name filter string cannot be * or **
30026	The selected data set list is required.
30040	Deploy preference is required.
30041	Deploy preference is invalid.
30042	Image storage server is required.
30043	Component is required.
30044	Licensing information is required.
30045	Target environment is required.
30046	Number of CPs is invalid.
30047	Memory size is invalid.
30048	Deployment label is required.
30049	Not enough CPs to start the emulator.
30050	Not enough memory to start the emulator
30051	Type of storage server is required (for example FTP or SFTP)
30052	The cloud platform label is required.
30053	The user name who created the cloud platform configuration is required.
30054	The user name to login the cloud is required.
30055	The password of the user to login the cloud is required.
30056	The scope of the cloud is required.
30057	The deployment label is already existed.
30058	The deployment label is not found.
30059	The scope of the cloud is invalid.
30060	The request to import an existing ZD&T image failed due to missing required input.

Table 20. Response error codes (continued)

Error ID	Description
30061	The request to validation an existing ZD&T image failed due to missing required input.
30064	The comment is longer than the allotted maximum of 225 characters.
30080	Target environment is required.
30081	Target environment is invalid.
30082	Source system is required.
30083	Source system is invalid.
30084	Image storage server is required.
30085	Image storage server is invalid.
30086	The IBM Application Discovery system is required.
30087	The IBM Application Discovery system information is invalid or empty.
30088	Port is invalid.
30089	The cloud platform is required.
30090	The endpoint URL of the cloud platform is required.
30091	The scope of the cloud platform is required.
30092	The domain name URL of the cloud platform is required.
30093	The project name URL of the cloud platform is required.
30094	The type of the cloud platform is required.
30095	The username of the cloud platform is required.
30096	The password of the cloud platform is required.
30097	No catalog found on the cloud platform.
30098	Authentication of the cloud platform has failed.
30099	Query on the cloud platform has failed.
30100	Credential is required.
30101	Credential is invalid.
30102	Token is required.
30103	Token is invalid.
30104	Target environment hostname is required.
30105	Target environment hostname is invalid.
30106	SSH port is required.
30107	SSH port is invalid.
30108	User ID is required.
30109	Password is required.
30110	Your password has expired.
30111	Target environment label is required.

Table 20. Response error codes (continued)

Error ID	Description
30112	Target environment label is invalid.
30113	A private key file is required.
30114	Saving private key for user \${username} failed!
30115	The file size cannot exceed 50KB.
30116	All IPL parameters needs to be entered, this includes the device address, IODF address, and device suffix.
30120	MVSDSALLOCATE is required.
30121	MVSDSALLOCATE is invalid.
30122	Parameter json string is required.
30123	Parameter json string is invalid.
30124	Image storage server hostname is required.
30125	Image storage server hostname is invalid.
30140	User is required.
30141	User is invalid.
30142	Username is required.
30143	Username is invalid.
30144	ADCD name is required.
30145	ADCD name already exists.
30146	User role is required.
30147	User role is invalid.
30148	User role does not exist.
30149	The version of the application is invalid.
30150	The number of crypto co-processors must be between 1 to 16.
30151	The domain that is defined must be a positive number.
31000	This application image is being processed, please wait for a while.
31001	This application image is being extracted, please confirm to delete it.
31002	This application image is being deployed to \${targetSystem}
31003	This application image is being deprovisioned from \${targetSystem}, please wait for it to finish.
31004	This application image is being deployed to \${targetSystem} but failed, please confirm to delete it.
31005	This selected application component \${componentName} is not in this application.
31006	This selected application component \${componentName} is found in other applications as shown below: \${applicationsList}
31007	This application image is being deleted.
31008	This application image is not available.

Table 20. Response error codes (continued)

Error ID	Description
31009	You must choose at least one component that contains the system residence volume
31010	Only data set application components from another IBM Z mainframe can be added to this application.
31011	The data set application component \${componentName} cannot be added to this application image as it has not been extracted.
31012	This application image cannot be deleted as it deployed to the following target environments: \${targetSystemsList}
31013	Only an ADCD or Z system application needs to be migrated.
31014	This \${applicationName} does not need migration.
31015	This \${applicationName} needs migration before you can deploy it.
31016	The contents of an application image cannot be edited as the image has been deployed to a target environment.
31017	One component cannot be processed in multiple requests.
31018	The name of an application image cannot be changed when creating a new version.
31019	The name for the new application image must be changed when duplicating an application image.
31020	The system residence volume component must be selected when creating an application image.
31021	This application image contains duplicate volume \${volumeName}
31022	This application image contains duplicate dataset \${datasetName}
31023	This ADCD version doesn't support including dataset or CICS component extracted from other z/OS
31024	This ADCD image must include at least one CICS component in order to include CICS components extracted from other z/OS.
31025	This ADCD version doesn't support including Db2 components that are extracted from other z/OS.
31026	This ADCD image cannot include more than one Db2 runtime.
31027	This ADCD image must include at least one Db2 component to include Db2 components that are extracted from other z/OS.
31028	Duplicate table name \${tableName} is found in schema \${schemaName}.
31029	Duplicate database name \${databaseName} is found.
31030	Only one ADCD Db2 component can be selected.
31031	The version of Db2 from component \${componentName} does not match the version of ADCD Db2 component \${componentName}.
31032	The Db2 files failed to be transferred.
31050	Wait for the application component \${applicationComponentName} to finish processing.
31051	The application component \${applicationComponentName} has been deleted.

Table 20. Response error codes (continued)

Error ID	Description
31052	Only application components containing data sets created from an IBM Z mainframe support extraction being stopped or resumed.
31053	The extraction of this application component cannot be resumed as it was previously extracted successfully.
31054	The contents of this application component cannot be edited as the component has been deployed
31055	The name of an application component cannot be changed when creating a new version.
31056	The name for the new application component must be changed when duplicating an application component.
31057	A new version can only be created from the latest version.
31058	This component contains duplicate data set \${datasetName}
31100	Cannot find the related record for given application and target environment.
31101	IPL parameters creation failed.
31102	Read properties file failed.
31103	Input stream contains a malformed Unicode escape sequence.
31104	Internal server error when attempting to read ADCD property file.
31105	No ADCD files.
31106	Wrong arguments passed to the configuring ADCD script.
31107	The specified log \${javaLogsPath} is not found.
31108	The application image size is 0 and could not be deployed to cloud platform.
31109	Deployment to this target system has already been started.
31110	Cannot deploy to cloud using this REST API.
31111	Cannot deprovision a target cloud instance while it is being provisioned.
31112	Cannot deprovision a target cloud instance while it is being deployed.
31113	Deprovision of \${applicationName} from target cloud platform \$ {cloudInstanceName} failed with error return code \${errorCode}
31114	Cannot deprovision a target cloud instance while it is being deprovisioned.
31115	An application image cannot be deployed to a target environment when an image has already been deployed to it.
31116	Specify the credentials to login to z/OS for deploying \${applicationName}
31117	The password for user \${username} to login to z/OS \${systemName} has expired and needs to be changed.
31118	Cannot deploy to a target environment that is being deprovisioned.
31119	This application cannot be deployed because its dataset component \$ {componentName} has not been extracted.
31120	The user ID must not be longer than 8 characters.

Table 20. Response error codes (continued)

Error ID	Description
31121	Authentication failed.
31122	The Linux group zpdtd does not exist.
31123	The Linux user ID ibmsys1 does not exist.
31124	Unsupported platform
31125	{CPUName} is not a supported CPU architecture.
31126	The user performing the deployment does not belong to Linux group zpdtd.
31127	The IBM ZD&T emulator is already running on the target environment.
31128	A software-based license server with an IP address containing 255 in the fourth octet is not supported.
31129	The emulator licensing server hostname {licensingServerName} could not resolve to a valid IP address from the system.
31130	The connection to hostname {remoteHost} on port {portNumber} from the system {sourceSystem} has failed.
31131	The deployment directory {deploymentDirectoryPath} does not exist on the target environment.
31132	Write access is required to the deployment directory {directoryPath}
31133	The deployment directory does not have enough available disk space. {NeededSpace} is required but only has {CurrentAvailableSpace} available.
31143	The required software 'ftp' is not installed on the system {targetSystem}
31144	The required 32-bit library 'stdlib' is not installed on the system {targetSystem}
31145	The required software 'perl' is not installed on the system {targetSystem}
31146	The required software 'zip' is not installed on the system {targetSystem}
31147	The required software 'unzip' is not installed on the system {targetSystem}
31148	The required software 'gzip' is not installed on the system {targetSystem}
31149	The required software 'bc' is not installed on the system {targetSystem}
31150	The required library 'libasound2' is not installed on the system {targetSystem}
31151	You do not have permission to run the command {command} on the system {systemName}
31152	You do not have permission to change the owner of file {fileName} on the system {systemName}
31153	Failed to list all of the privilege management commands.
31154	{DistroName} is not a supported Linux distribution.
31155	The IBM ZD&T emulator needs to be running in order to deploy datasets.
31159	Cannot run command 'nc' to validate the connectivity to hostname {systemName} on port {portNumber} from the system {sourceSystem}
31160	Cannot run command 'getent' to resolve hostname {hostname} from the system {sourceSystem}

Table 20. Response error codes (continued)

Error ID	Description
31162	This application cannot be deployed because it contains component \$ {componentName} that needs to be migrated.
31163	This application cannot be deployed because it needs to allocate at least 1 CP to the image, with a default recommendation of 3.
31164	This application cannot be deployed because it can only allocate up to eight processors to the image.
31165	This application cannot be deployed because it cannot allocate more zIIPs than CPs.
31166	Specify the image that has been deployed to the target environment.
31167	Specify the image that has been deployed to the target environment.
31168	The application \${name} with version \${version} is not found to be deployed on host \${targetSystem}.
31169	Reading devmap failed.
31170	No spare 3390 volume available for restoring data set.
31171	Volume cannot be found at \${location}.
31172	The emulator information cannot be retrieved from the target environment \$ {targetSystem}.
31173	A failure occurred when trying to update the emulator.
31174	IBMUSER cannot be set as the password for ADCD Images.
31175	The RAM that is set for the emulator cannot exceed 18 GB.
31300	Failed to list volumes.
31301	Missing or invalid parameters when executing command \${command}, with output \$ {output}.
31302	An error occurred when allocating a file when executing command \${command}, with output \${output}.
31303	The z/OS SMS settings cannot be acquired.
31304	The z/OS spool volumes cannot be acquired.
31305	The z/OS proclibs cannot be acquired.
31306	The z/OS APF-authorized list cannot be acquired when executing command \$ {command} with output \${output}.
31307	The z/OS TCP/IP settings cannot be acquired when executing command \$ {command}, with output \${output}.
31308	No SYSPRINT/stdout when executing command \${command}, with output \${output}.
31309	The load library for the z/OS extraction tool is not APF authorized. Review section 6.1.7 in the Program Directory for IBM Z Development and Test Environment Enterprise Edition (Program Number 5725-G39, FMID HALMB00) for more information.
31310	READ access is not granted to the SAF profile in the class DASDVOL.
31311	The Volume image file cannot be opened: \${output}.
31312	Compression initialization failed - internal error.

Table 20. Response error codes (continued)

Error ID	Description
31313	Cannot open the volume - internal error.
31314	Command invoked cannot execute.
31315	Command not found.
31316	Invalid argument given to the exit statement.
31317	Fatal error signal with return code \${retCode}.
31318	Dataset allocated failed.
31319	Volume \${volumeFileName} extraction failed.
31320	Failed to list data set.
31321	Failure retrieving IBM Z system information, return code: \${retCode}
31322	Retrieving volume information for data sets failed.
31323	No volume information retrieved for selected data sets.
31324	An ABEND occurred on the IBM Z with the following output. Please send this output along with the System Log to IBM support for further support. \${output}
31325	These data sets cannot be extracted because your user ID has insufficient access to read them: \${dsNames}
31326	These data sets cannot be extracted because their containing volume(s) cannot be found : \${dsNames}
31327	These data sets cannot be extracted because of unknown errors: \${dsNames}
31328	APF authorization failed.
31329	The CICS region information is required.
31330	The CICS region name is required.
31331	The CICS region \${regionName} was already defined.
31332	The CICS region \${regionName} was not found.
31333	The LOADLIB of REXX script in command \${command} was not found. Error output: \${errorOutput}
31334	Parsing CICS CSD got format error. Dump output is: \${output}
31335	You must select some GROUPs defined in the CICS CSD.
31336	Got unexpected exception when trying to retrieve CICS CSD definition for \${CICSRegion}
31337	The input LIST name \${listName} is not a valid CSD LIST name in \${CICSRegion}
31338	The input GROUP name \${listName} is not a valid CSD GROUP name in \${CICSRegion}
31339	Components are created using the CICS region \${CICSRegion}. Need to delete those CICS components first.
31340	Vary Online ZD&T created volumes failed with error code \${errorCode}.
31341	The Db2 subsystem information is required.
31342	The Db2 subsystem ID is required.

Table 20. Response error codes (continued)

Error ID	Description
31343	The Db2 subsystem ID \${ID} was already defined.
31344	The Db2 subsystem ID \${ID} was not found.
31345	An error occurred when allocating required work files for call to DSNTEP2.
31346	The expected DDL file is missing from the tmp directory.
31347	An error occurred when creating target Db2 objects from DSNTEP2.
31348	DDL generation warning.
31349	An error occurred when reading or writing DDL file for update with FEUXPUNA.
31350	An error occurred when executing DSNUTILU or Db2 Load ended with RC > 4.
31351	No tables can be unloaded because input table list is empty or validation resulted in an empty file.
31352	The number of tables exceeds the maximum number of tables that can be unloaded. Check the contents of the table file.
31353	An error occurred when setting up DSNREXX Environment (RXSUBCOM).
31354	An error occurred when connecting to Db2 SSID.
31355	An error occurred when connecting to Db2 Location.
31356	An error occurred when defining Db2 Cursor.
31357	An error occurred when opening Db2 Cursor.
31358	An error occurred during Db2 Prepare.
31359	Invalid parameter passed to FEUXSQLP resulting in error for SQL command setup.
31360	Max Rows returned from SQL Call, max rows is set to \$1.
31361	Insufficient parameters or error reading parameter file.
31362	No tables were found in the table list or an error occurred when reading the table list file.
31363	Db2 unload failed.
31364	An error occurred when updating Db2 Load parameters.
31365	The expected PIPE or file was not found.
31366	An error occurred when allocating the required work files for Db2 Admin Tool DDL generation.
31367	Utility Statement exceeds the maximum length that is allowed.
31368	The schema parameter string is required.
31369	Db2 Admin Tool DDL generation error.
31600	Parameter object required.
31601	Image service is not available on the cloud.
31602	The OpenStack cloud image could not be found.
31603	The label of the cloud platform is required.
31604	The label of the cloud platform already exists.

Table 20. Response error codes (continued)

Error ID	Description
31605	Network is required.
31606	The selection of flavor is required.
31607	The selection of server image is required.
31608	Compute service is not available on the cloud.
31609	The specific flavor cannot be found.
31612	ID is invalid.
31649	The default user ID is required.
31650	Cannot find the uuid of the device.
31652	Availability zone is required.
31653	Floating IP pool is required.
31700	Target system still contains application image, so could not be deleted.
31701	Required parameters are missing for deleting a target environment.
31750	Privilege command option already exists.
31751	The privilege command syntax is required to be defined on the configure target environments page.
31752	ID is invalid.
31753	Privilege command option does not exist.
31754	Cannot find network interfaces from server.
60000	This target environment has not been provisioned.
60001	This target environment cannot be used when it is in use by another operation.
70001	The ZD&T extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS returned an unexpected result.
70992	A Java\u2122 runtime environment is either not installed on this z/OS or is not added to the z/OS UNIX System Services PATH environment variable for this user.
70993	The ZD&T extraction utility SFTP module (/usr/lpp/IBM/zdt/zDTMainframeSFTP.jar) on this z/OS is either not installed or corrupted.
70994	The ZD&T extraction utility SFTP module (/usr/lpp/IBM/zdt/zDTMainframeSFTP.jar) on this z/OS is older than the current version. Please upgrade to the latest version.
70995	The ZD&T extraction utility (/usr/lpp/IBM/zdt/feuc) has not been installed.
70996	The ZD&T extraction utility version file could not be found.
70997	The ZD&T extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS is older than 12.0.1. Please upgrade to the latest version.
70998	The ZD&T extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS is older than the current older. Please upgrade to the latest version.
70999	Internal error acquiring information from z/OS.
79999	An error occurred when connecting to z/OS: \${zosName}.
99998	Unexpected exception.

Table 20. Response error codes (continued)

Error ID	Description
99999	Unknown error.

Updating Enterprise Edition

To update Enterprise Edition, you need to run the installer from the root user ID.

1. Open the directory that contains the installer `ZDT_Install_EE_V12.0.5.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_EE_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_EE_V12.0.5.0.tgz
```

4. Optional: Read the `README.txt` file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

6. Select **1** for ZD&T Enterprise Edition.
7. Select **2** to update.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. After the update completes, run the following command to verify whether the update is successful.

- RHEL operating system

```
rpm -qa | grep zdtapp
```

- UBUNTU operating system

```
dpkg -l | grep zdtapp
```

If the package is installed successfully, the output contains the string `zdtapp 12.0.5.0`.

Alternatively, after you complete steps 1 - 4, you can update ZD&T Enterprise Edition silently by using the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --update --zdtee
```

Uninstalling Enterprise Edition

To uninstall Enterprise Edition, you need to return the license key, and uninstall the License Server first. Then, run the installer from the root user ID.

To uninstall ZD&T Enterprise Edition, complete the following steps:

1. Run the installer from the root user ID.

```
./ZDT_Install_EE_V12.0.5.0.x86_64
```

2. Select **1** for ZD&T Enterprise Edition.
3. Select **3** to uninstall.

Alternatively, after you complete steps 1 - 4, you can uninstall ZD&T Enterprise Edition silently by using the following command.

```
./ZDT_Install_EE_V12.0.5.0.x86_64 --uninstall --zdtee
```

Troubleshooting

Learn how to troubleshoot ZD&T Enterprise Edition 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 ZD&T Enterprise Edition, 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.

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

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 the 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 the IBM Application Discovery server is configured correctly through IBM Application Discovery configuration page.

Table 22. Troubleshooting the application components creation from IBM Application Discovery projects (continued)

Symptom	Problem Determination
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.
No data set 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 data sets 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 data set list is required.	Select the data set list on the right panel of the page.
The port number is not valid.	No character is allowed.

Scenarios

Learn about the scenarios of ZD&T Enterprise Edition for development and test activities.

Provisioning CICS subsystem from a source Z

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 IBM Application Discovery Server” on page 77](#).
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 83](#).
4. Create the application image. For more information, see [“Creating an application image from IBM Z components” on page 84](#).
5. Deploy the application image to the target machine. For more information, see [“Deploying application images” on page 86](#).

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) data sets that are used by CICS. It is defined in OMVS parameter of IEASYSxx, for example:

```
OMVS=(00,01,BP,IZ,RZ), SELECT BPXPRMCS
```

- LPAALSTxx has a list of LPA data sets. It is defined in IEASYSxx as LPA parameter. For example:

```
LPA=00, SELECT LPAALST
```

- 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
```

Parallel Sysplex

Parallel Sysplex is a cluster of IBM mainframes that act together as a single system image with z/OS. Used for disaster recovery, Parallel Sysplex combines data sharing and parallel computing to allow a cluster of up to 32 systems to share a workload for high performance and high availability.

ZD&T Parallel Sysplex can be used to enable a Sysplex environment that is running within z/VM. To run ZD&T Parallel Sysplex, you need to run the installation package on the 8086 Intel-based hardware with RHEL or Ubuntu operating system. This package emulates z/Architecture with virtual I/O and devices.

Before you run ZD&T Parallel Sysplex, you need to ensure that you have the entitlements to run it. To enable ZD&T Parallel Sysplex, prepare a License Server with the license that is applied on it.

Software-based License Server

When you use the software-based licensing, the Software-based License Server provides a server for centralized management of license keys for one or more instances of ZD&T Parallel sysplex. With this method, a USB hardware device is not required.

To prepare the Software-based License Server, install the server and authenticated license key files. The licensee is not authorized to use Parallel Sysplex or any of its components except when the Software-based License Server is activated with a license key file and is accessible by the program, as it provides the proof of license entitlement.

Note: The Software-based License Server and ZD&T Parallel Sysplex cannot be installed on the same machine.

Each license key file is generated with a number of entitled emulated central processors, which is referred as Central Processors (CPs). A single instance of ZD&T 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 more information about the number of CPs that is used with an instance, see section 2.3.4 "Performance" in the [zPDT Guide and Reference](#).

The following topology describes the Software-based License Server and various components of Parallel Sysplex.

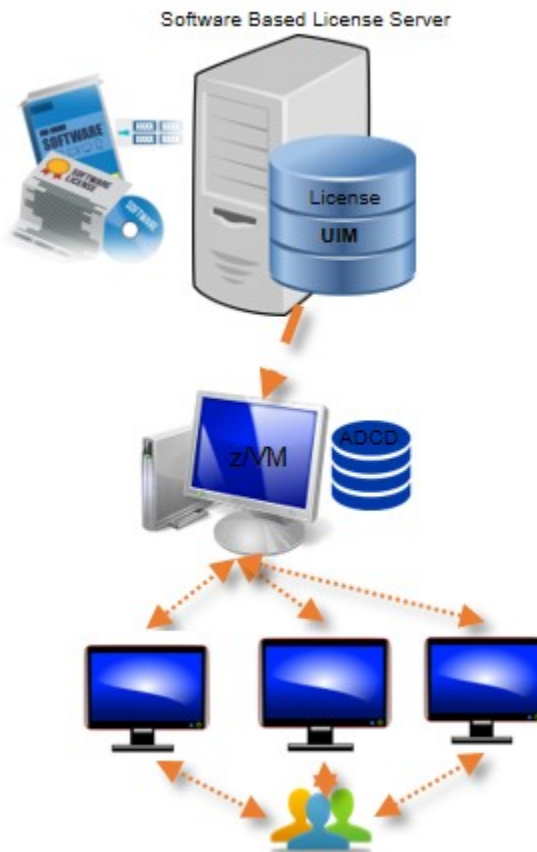


Figure 9. The Software-based License Server and various components of Parallel Sysplex

Hardware-based License Server

The earlier versions of ZD&T Parallel Sysplex require a USB hardware device that contains a license key to control 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. After you order the a license key (called update file) from Rational License Key Center, you need to apply the license key to the USB hardware device. Then, connect the USB hardware device to the machine that is known as Hardware-based License Server.

Note: The Hardware-based License Server and ZD&T Parallel Sysplex cannot be installed on the same machine, and the network connectivity is not needed between the machines.

Each update file is generated with a number of entitled emulated central processors, which is referred as Central Processors (CPs). A single instance of ZD&T requires at least 1 CP, and can run with a maximum of eight CPs. The number of CPs that is needed depends on the number of users and the types and amount of processing required. For more information about the number of CPs that is used with an instance, see section 2.3.4 "Performance" in the zPDT Guide and Reference.

The following topology describes the Hardware-based License Server and various components of Parallel Sysplex.



Figure 10. The Hardware-based License Server and various components of Parallel Sysplex

Rational Tokens

Rational Software License Server (RLKS) uses Rational Tokens to enable IBM supplied products. If you also use other IBM products, and already set up RLKS, you can use Rational Tokens to enable ZD&T Parallel Sysplex.

To enable Parallel Sysplex by Rational Tokens, you need to decide to use Hardware-based License Server or Software-based License Server. In either case, the license key file indicates that Rational Tokens are required.

For ZD&T Parallel Sysplex, each emulated central processor within an instance of a Z virtual machine requires a particular number of Rational Tokens, based on the token value for ZD&T Parallel Sysplex. 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.

The use of Rational Tokens does not replace the requirement for a license key for ZD&T Parallel Sysplex. Either a software-based license key file or a USB hardware device with a valid update file is still required.

The following topology describes the Software-based License Server and Software-based License Server with Rational Tokens and various components of Parallel Sysplex.

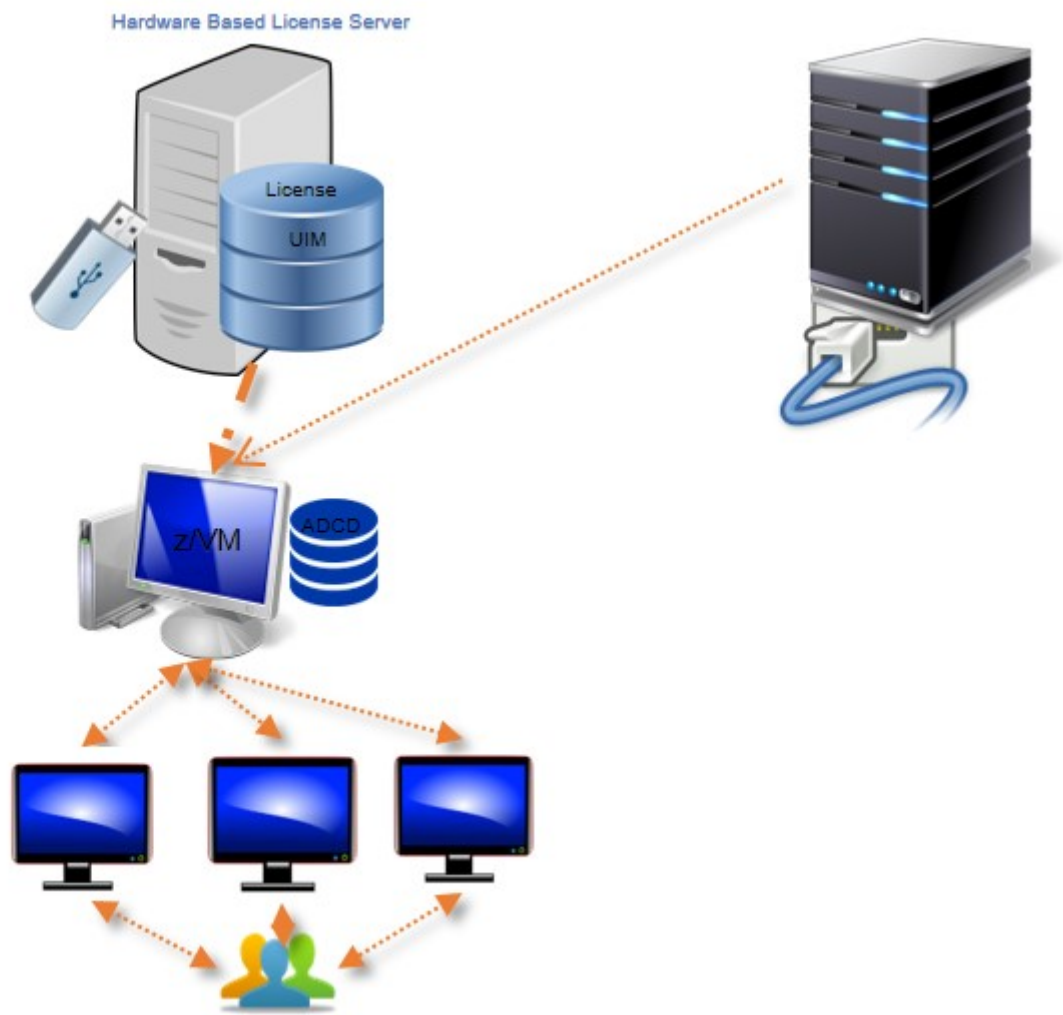


Figure 11. The Hardware-based License Server with Rational Tokens, and various components of Parallel Sysplex

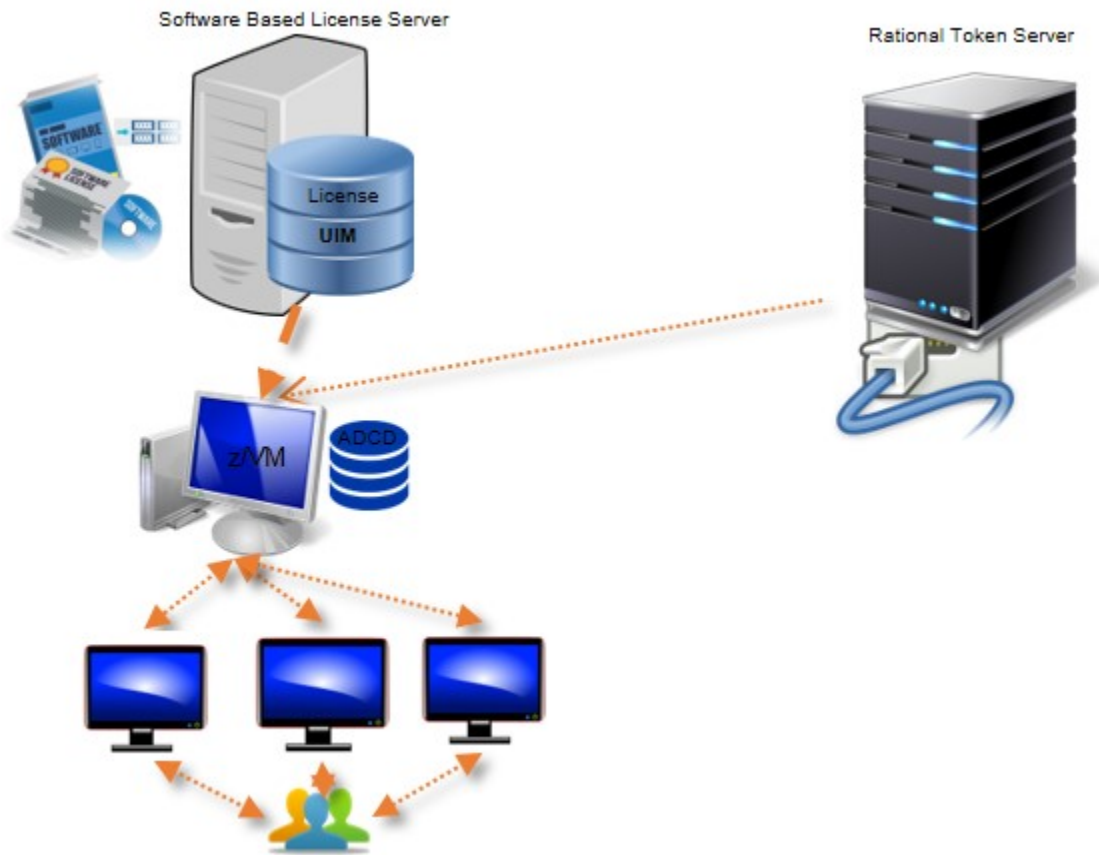


Figure 12. The Software-based License Server with Rational Tokens, and various components of Parallel Sysplex

Application Developers Controlled Distribution (ADCD)

ZD&T Parallel Sysplex comes with the Application Developers Controlled Distribution (ADCD) package, which is known as z/OS ADCD and z/VM distribution to enable ZD&T with Parallel Sysplex. ADCD package enables enterprise users to use and develop their applications with newest set of Mainframe software packages, such as CICS, Db2, IMS, z/OS 2.3, and JES2. z/VM enables the users to create the Parallel Sysplex environment on their commodity 8086 hardware.

For a complete list of z/OS products in ADCD, see [“ADCD z/OS V2R4 November Edition of 2019”](#) on page 178.

Checklist

If you want to install, update, or uninstall Parallel Sysplex, or if you want to update the license server or license, you can follow the steps that are provided in the checklist.

Table 23. New installation checklist for Parallel Sysplex

Installing Parallel Sysplex		Required/ Optional	Complete
1	“Prerequisites” on page 134 Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	<input type="checkbox"/>

Table 23. New installation checklist for Parallel Sysplex (continued)

Installing Parallel Sysplex		Required/ Optional	Complete
2	<u>Obtaining installation packages</u> Describes the steps to download all required packages to install Parallel Sysplex.	Required	<input type="checkbox"/>
3	<u>“Enabling the License Server” on page 139</u> Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
4	<u>“Installing Parallel Sysplex” on page 155</u> Describes the steps to install Parallel Sysplex.	Required	<input type="checkbox"/>
5	<u>“Activating ZD&T Parallel Sysplex instances” on page 156</u> Describes the steps to activate ZD&T instances	Required	<input type="checkbox"/>
6	<u>“Configuring Z instances” on page 159</u> Describes the steps to configure ADCD to start Z instances.	Required	<input type="checkbox"/>
7	<u>“Starting z/VM” on page 160</u> Describes the steps to start z/VM.	Required	<input type="checkbox"/>
8	<u>“Starting Z instances” on page 162</u> Describes the steps to start Z instances.	Required	<input type="checkbox"/>
9	<u>“Stopping Z instances and z/VM” on page 163</u> Describes the steps to stop Z and z/VM.	Optional	<input type="checkbox"/>

Table 24. Update checklist for Parallel Sysplex

Updating Parallel Sysplex		Required/ Optional	Complete
1	<u>“Returning a software-based license key” on page 145 or “Returning a hardware-based license key” on page 153</u> Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	<u>“Prerequisites” on page 134</u> Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	<input type="checkbox"/>
3	<u>Obtaining installation packages</u> Describes the steps to download all required packages to install Parallel Sysplex.	Required	<input type="checkbox"/>

Table 24. Update checklist for Parallel Sysplex (continued)

Updating Parallel Sysplex		Required/ Optional	Complete
4	<ul style="list-style-type: none"> • “Updating the installed Parallel Sysplex” on page 165. Describes the steps to update Parallel Sysplex if you installed ZD&T V11.0 or later versions. • “Uninstalling the earlier versions of ZD&T ” on page 163 and “Installing Parallel Sysplex” on page 155. Describes the steps to uninstall previous versions and install the latest Parallel Sysplex if you installed ZD&T V10.0.1 or earlier versions. 	Required	<input type="checkbox"/>
5	“Enabling the License Server” on page 139 Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
6	“Activating ZD&T Parallel Sysplex instances” on page 156 Describes the steps to activate ZD&T instances	Required	<input type="checkbox"/>
7	“Configuring Z instances” on page 159 Describes the steps to configure ADCD to start Z instances.	Required	<input type="checkbox"/>
8	“Starting z/VM” on page 160 Describes the steps to start z/VM.	Required	<input type="checkbox"/>
9	“Starting Z instances” on page 162 Describes the steps to start Z instances.	Required	<input type="checkbox"/>

Table 25. Update checklist for license server

Updating license server		Required/ Optional	Complete
1	“Returning a software-based license key” on page 145 or “Returning a hardware-based license key” on page 153 Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	“Updating the Software-based License Server” on page 147 or “Updating the Hardware-based License Server” on page 154 Describes the steps to update the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Enabling the License Server” on page 139 Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>

<i>Table 26. Update checklist for license</i>			
Updating license		Required/Optional	Complete
1	“Returning a software-based license key” on page 145 or “Returning a hardware-based license key” on page 153 Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	“Enabling the License Server” on page 139 Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>

<i>Table 27. Uninstallation checklist for Parallel Sysplex</i>			
Uninstalling Parallel Sysplex		Required/Optional	Complete
1	“Returning a software-based license key” on page 145 or “Returning a hardware-based license key” on page 153 Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	“Uninstalling the Software-based License Server” on page 148 or “Uninstalling the Hardware-based License Server” on page 155 Describes the steps to uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Uninstalling Parallel Sysplex” on page 166 Describes the steps to uninstall Parallel Sysplex.	Required	<input type="checkbox"/>

Planning

Before you start with ZD&T Parallel Sysplex, learn about the information about hardware and software requirements, installation packages, and the steps to set up basic Linux system.

Prerequisites

Before you install or update ZD&T Parallel Sysplex, you need to check the hardware and software requirements for Parallel Sysplex, and set up the base Linux system.

Hardware and software requirements

For a complete list of ZD&T hardware and software requirements, you can generate the report from [Software Product Compatibility Reports](#).

Setting up the base Linux system

Before you access the Linux Software repository to download the required software, you need to ensure the following prerequisites.

- You have access to internet on your machine.
- You have root or sudo access to the Linux operating system.

- Your Linux system is configured to download the required Linux packages. You can use `yum install` or `apt-get` commands to install the package that might be required during the installation procedure.
- Acquiring Linux knowledge

To set up the base Linux system, you need to have some Linux knowledge to ensure that you can go through some basic concepts and commands from internet. The following commands might be required to use when you install the package.

- Obtaining access to software repository to run YUM or `apt-get` commands

Make sure that you have access to software repository to run YUM commands on RHEL machine, or your Ubuntu machine have access to run `apt-get` commands. ZD&T Installer will install all required packages. However, if you don't want ZD&T installer to install the required packages that are listed below, you need to install the packages before you start ZD&T Installer.

- YUM commands on RHEL operating system

```
yum -y install iptables
yum -y install ftp
yum -y install libstdc++.i686
yum -y install perl
yum -y install zip
yum -y install unzip
yum -y install gzip
yum -y install bc
```

- `apt-get` commands on UBUNTU operating system

```
apt-get -y install iptables
dpkg --add-architecture i386
apt-get -y update
apt-get -y install libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 lib32stdc++6
apt-get -y install ftp
apt-get -y install perl
apt-get -y install zip
apt-get -y install unzip
apt-get -y install gzip
apt-get -y install bc
apt-get -y install libasound2
apt-get -f install
```

- Setting up the system time

Set your hardware clock to Coordinated Universal Time (UTC) to avoid problems when Daylight Saving Time starts and stops. The License Server 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 the standard time.

- Installing a TN3270e client for the MVS console

The following TN3270e clients are examples of clients that can be used with the recent ZD&T offerings:

- x3270 (recent versions)
- Recent Personal Communications releases that run on the Windows system

The most commonly used TN3270e client is x3270 running on the native Linux host. This client is used in the sample start script for ZD&T. 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
```

- Managing the firewall and other security functions

You must manage the firewall and other security functions that you install with your Linux. To simplify the configuration and operation, disable any firewall when you first work with zPDT. 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)
```

Note:

- ZD&T Parallel Sysplex installer opens the ports 1947, 3270, 9451, 9450 to communicate with the License Server and terminal emulator. Any restriction that might prevent the installer from opening the ports should not be on your Linux machine. You need to deactivate any firewall that is active, or ask your network administrator to deactivate the firewall.

Installation packages

Download the required installation packages from Passport Advantage before you install or update ZD&T Parallel Sysplex.

To download the package, complete the following steps:

1. Log on to [Passport Advantage](#).
2. Select **Software Downloads** and **Media Access**.
3. Select **Program offering and agreement number**, and click **Continue**.
4. Enter the part description or part number, and click **Finder**.
5. Optionally, you can click the alphabetical order list to display and view the product by name.
6. Select **All Operating Systems** in the **Operating system** field, and **All Languages** in the **Languages** field. Then, click **Go**.
7. If you want to download the individual media from the list, click **Select individual files** to expand the list.
8. Verify the e-assemblies that you want to download with the list that is mentioned below in the table.

Note: Verify the integrity of downloaded ADCD packages by using the MD5SUM that is in the adcd.md5, ps.md5 and z/VM files.

	Name	Part No.	Package Type	File Name
1	IBM Z Development and Test Environment Parallel Sysplex V12.0.5 Installation Multilingual	CC5FXML	ZD&T Parallel Sysplex Installer	ZDT_Install_PS_V12.0.5.0.tgz
2	IBM Z Development and Test Environment Parallel Sysplex V12.0.5 Installation checksum	CC5FYML	checksum (md5 checksum)	ps.md5
3	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	CC59MML	ADCD November 2019 Edition RSU level 1909	A4RES1.ZPD
4	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	CC59NML	ADCD November 2019 Edition RSU level 1909	A4RES2.gz
5	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 3 of 19 - System volume Multilingual	CC59PML	ADCD November 2019 Edition RSU level 1909	A4SYS1.gz

	Name	Part No.	Package Type	File Name
6	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	CC59QML	ADCD November 2019 Edition RSU level 1909	A4CFG1.gz
7	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	CC59RML	ADCD November 2019 Edition RSU level 1909	A4USS1.gz
8	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	CC59SML	ADCD November 2019 Edition RSU level 1909	A4USS2.gz
9	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	CC59TML	ADCD November 2019 Edition RSU level 1909	A4PRD1.gz
10	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	CC59UML	ADCD November 2019 Edition RSU level 1909	A4PRD2.gz
11	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	CC59VML	ADCD November 2019 Edition RSU level 1909	A4PRD3.gz
12	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	CC59WML	ADCD November 2019 Edition RSU level 1909	A4PRD4.gz
13	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	CC59XML	ADCD November 2019 Edition RSU level 1909	A4DIS1.gz
14	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	CC59YML	ADCD November 2019 Edition RSU level 1909	A4DIS2.gz
15	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	CC59ZML	ADCD November 2019 Edition RSU level 1909	A4DIS3.gz
16	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	CC5A0ML	ADCD November 2019 Edition RSU level 1909	A4PAGA.gz
17	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	CC5A1ML	ADCD November 2019 Edition RSU level 1909	A4PAGB.gz

	Name	Part No.	Package Type	File Name
18	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	CC5A2ML	ADCD November 2019 Edition RSU level 1909	A4PAGC.gz
19	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	CC5A3ML	ADCD November 2019 Edition RSU level 1909	A4USR1.gz
20	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	CC5A4ML	ADCD November 2019 Edition RSU level 1909	SARES1.ZPD
21	IBM Z Development and Test Environment V12.0.5 with z/OS 2.4 Part 19 of 19 – zCX Container volume	CC5A5ML	ADCD November 2019 Edition RSU level 1909	A4ZCX1.gz
22	IBM Z Development and Test Environment V12.0.5 with Rational Team Concert volume 6.0.6 Multilingual	CC5A6ML	ADCD November 2019 Edition RSU level 1909	A4BLZ1.gz
23	IBM Z Development and Test Environment V12.0.5 with DB2 V12 Part 1 of 2 Multilingual	CC5ABML	ADCD November 2019 Edition RSU level 1909	A4DBC1.gz
24	IBM Z Development and Test Environment V12.0.5 with DB2 V12 Part 2 of 2 Multilingual	CC5ACML	ADCD November 2019 Edition RSU level 1909	A4DBC2.gz
25	IBM Z Development and Test Environment V12.0.5 with DB2 V11 Part 1 of 2 Multilingual	CC5ADML	ADCD November 2019 Edition RSU level 1909	A4DBB1.gz
26	IBM Z Development and Test Environment V12.0.5 with DB2 V11 Part 2 of 2 Multilingual	CC5AEML	ADCD November 2019 Edition RSU level 1909	A4DBB2.gz
27	IBM Z Development and Test Environment V12.0.5 with CICS 5.4 Multilingual	CC5AFML	ADCD November 2019 Edition RSU level 1909	A4C541.gz
28	IBM Z Development and Test Environment V12.0.5 with CICS 5.5 Multilingual	CC5AGML	ADCD November 2019 Edition RSU level 1909	A4C551.gz
29	IBM Z Development and Test Environment V12.0.5 with IMS 14.1 Multilingual	CC5AHML	ADCD November 2019 Edition RSU level 1909	A4IME1.gz
30	IBM Z Development and Test Environment V12.0.5 with IMS 15.1 Multilingual	CC5AIML	ADCD November 2019 Edition RSU level 1909	A4IMF1.gz
31	IBM Z Development and Test Environment V12.0.5 with IBM Installation Manager 1.4.3 Multilingual	CC5AJML	ADCD November 2019 Edition RSU level 1909	A4INM1.gz

	Name	Part No.	Package Type	File Name
32	IBM Z Development and Test Environment V12.0.5 with DB2 Archive Logs Multilingual	CC5AKML	ADCD November 2019 Edition RSU level 1909	A4DBAR.gz
33	IBM Z Development and Test Environment V12.0.5 with Tivoli Omegamon XE products 5.5.1 Multilingual	CC5ALML	ADCD November 2019 Edition RSU level 1909	A4KAN1.gz
34	IBM Z Development and Test Environment V12.0.5 with WAS V9.0 Part 1 of 2 Multilingual	CC5AMML	ADCD November 2019 Edition RSU level 1909	A4W901.gz
35	IBM Z Development and Test Environment V12.0.5 with WAS V9.0 Part 2 of 2 Multilingual	CC5ANML	ADCD November 2019 Edition RSU level 1909	A4W902.gz
36	z/VM 6.4 #1 of 6 RES volume / M01RES Multilingual	CNRL2ML	z/VM 6.4	M01RES.gz
37	z/VM 6.4 #2 of 6 Page volume / M01P01 Multilingual	CNRL3ML	z/VM 6.4	M01P01.gz
38	z/VM 6.4 #3 of 6 System volume / M01S01 Multilingual	CNRL4ML	z/VM 6.4	M01S01.gz
39	z/VM 6.4 #4 of 6 Work volume 1 / M01W01 Multilingual	CNRL5ML	z/VM 6.4	M01W01.gz
40	z/VM 6.4 #5 of 6 Common volume 1 / VMCOM1 Multilingual	CNRL6ML	z/VM 6.4	VMCOM1.gz
41	z/VM 6.4 #6 of 6 Release volume 1 / 640RL1 Multilingual	CNRL7ML	z/VM 6.4	640RL1.gz
42	z/VM 6.4 - Checksum	CNVA5ML	z/VM 6.4	zvm.md5
43	IBM Z Development and Test Environment V12.0.5 Rocket Open Source Languages and Tools for z/OS	CC5APML	ADCD November 2019 Edition RSU level 1909	ZDTRKT.gz
44	IBM Z Development and Test Environment V12.0.5 ADCD Checksum Multilingual	CC5ARML	ADCD package MD5 checksum	nov2019_adcd_md5.txt

Enabling the License Server

To operate ZD&T Parallel Sysplex and authorize the licensee to use the product or any of its components, you need to enable the License Server. You can follow the steps that are provided in the checklist.

<i>Table 28. Enablement checklist for software-based licensing</i>			
Software-based license		Required/ Optional	Complete
1	“Installing the Software-based License Server” on page 141 Describes the steps to install the Software-based License Server	Required	<input type="checkbox"/>

Table 28. Enablement checklist for software-based licensing (continued)

Software-based license		Required/ Optional	Complete
2	Obtaining an update file Describes the steps to obtain an update file for software license server.	Required	<input type="checkbox"/>
3	Applying the update file to the Software-based License Server Describes the steps to apply the update file to Software-based License Server.	Required	<input type="checkbox"/>

Table 29. Enablement checklist for hardware-based licensing

Hardware-based license		Required/ Optional	Complete
1	“Installing the Hardware-based License Server” on page 150 Describes the steps to install the Hardware-based License Server.	Required	<input type="checkbox"/>
2	Obtaining an update file for Hardware-based License Server Describes the steps to obtain an update file for Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Applying the update file to the Hardware-based License Server” on page 152 Describes the steps to apply the update file to ZD&T Parallel Sysplex.	Required	<input type="checkbox"/>

Table 30. Switching from Hardware-based License Server to Software-based License Server

Hardware-based license to Software-based license		Required/ Optional	Complete
1	“Returning a hardware-based license key” on page 153 Describes the steps to return a hardware-based license key.	Required	<input type="checkbox"/>
2	“Uninstalling the Hardware-based License Server” on page 155 Describes the steps to uninstall Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Installing the Software-based License Server” on page 141 Describes the steps to install Software-based License Server.	Required	<input type="checkbox"/>

<i>Table 30. Switching from Hardware-based License Server to Software-based License Server (continued)</i>			
Hardware-based license to Software-based license		Required/Optional	Complete
4	<u>Obtaining an update file for Software-based License Server</u> Describes the steps to obtain an update file for Software-based License Server.	Required	<input type="checkbox"/>
5	<u>Applying the update file to the Software-based License Server</u> Describes the steps to apply the update file to Software-based License Server.	Required	<input type="checkbox"/>

Software-based licensing

When you use the software-based licensing, the Software-based License Server provides a server for centralized management of license keys for one or more instances of ZD&T.

Before you enable the software-based licensing, you need to know the terms that are used in the enablement process.

Host ID and host name

Each license key file is uniquely identified in the Rational License Key Center with the host name and host ID of the Software-based License Server 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 Software-based License Server license key file. 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 Software-based License Server for which it was generated.

Capability configuration and capacity

Specific combinations of Parallel Sysplex support and use of Rational Tokens define the capability configuration of the license key and of the Software-based License Server. 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 Software-based License Server. That license key file allows operating within only one of the following 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.

Installing the Software-based License Server

To install the Software-based License Server, run the installer from the root user ID.

1. Open the directory that contains the installer ZDT_Install_PS_V12.0.5.0.tgz.

2. Change the authority.

```
chmod 755 ZDT_Install_PS_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PS_V12.0.5.0.tgz
```

4. Optional: Read the README .txt file for the complete installation steps.

5. Run the installer.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

6. Select **1** for the Software-based License Server.

7. Select **1** to install.

8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.

9. Read the warning message, and accept or decline by entering y or n.

10. Optional: Enter y to install all needed dependencies or enter n to decline.

By entering y, all required dependencies will be installed. The list of dependencies are mentioned in the “Prerequisites” on page 134. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

11. After the installation completes, run the following commands to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep aksusbd  
rpm -qa | grep zpdtuim
```

- UBUNTU operating system

```
dpkg -l | grep aksusbd  
dpkg -l | grep zpdtuim
```

Important: The installer creates the ibmsys1 as the non-root user ID to make ZD&T work, you need to switch to ibmsys1 and change the password. If you like to create the non-root user ID, use the following commands for reference. Make sure that the non-root user ID is a part of group zpdt.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpdt -m ibmsys2
```

To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

Alternatively, after you complete steps 1 - 4, you can install the Software-based License Server silently.

- To install the Software-based License Server on the virtual machine, run the following command:

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --install --zdtswlicense --ackVM
```

- To install the dependencies, run the following command:

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --install --zdtswlicense --ackVM --dep
```

Obtaining an update file

To enable the Software-based License Server, you must first obtain a unique update file, and then apply the update file to the Software-based License Server.

For Software-based License Server

Learn about the steps to obtain an update file for Software-based License Server.

1. Log in to the machine where you installed the Software-based License Server.
2. Go to `/opt/IBM/LDK`, and run the following command from the root user ID. This step is to create a file that is known as the request file in the root home directory.

```
./request_license
```

3. Log in to Rational License Key Center (RLKC), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
4. On the left side of the screen, select **Get Keys**.
5. Select the product line for **IBM Z Development and Test Environment**.
6. Select the license type that you want to apply to the key. If you purchased different license types, select **IBM Z DEVELOPMENT AND TEST ENVIRONMENT WITH PARALLEL SYSPLEX PER RVU**. Then, one or more boxes are displayed to show your orders.
7. Check the boxes next to the orders from which you would like to use licenses, and click **Next**.
8. A screen is displayed to show a table that you must complete. Provide the request file that you generated at step 2, and enter a number in the **Number of Emulated CPs** field. For more help, click the column headings in the table.
9. Click **Generate** at the bottom of the page. Then, a screen is displayed to download the update files.
10. Click **Download** for each update file that is generated to save the update file. Keep this file in the Linux machine where you have installed the Software-based License Server. 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: Use the binary transfer mode, but not a text mode such as ASCII. Also, spaces are not allowed in the path on the Linux machine.

For Software-based License Server with Rational Tokens

Learn about the steps to obtain an update file for Software-based License Server with Rational Tokens.

1. From the root user ID on the Software-based License Server, run the following 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 Software-based License Server. You can use the file 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 Rational License Key Center, and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
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 contains IBM Z Development and Test Environment. Make a note of the part number for the ZD&T 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.

8. Click **Generate** to generate the Rational Token license file. You are presented with a screen that shows 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 Server** to generate and download the update file for your Software-based License Server.
11. A **Required Information** page opens and displays a table that you must complete. Provide the request file for the Software-based License Server for which the license key file is 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 ZD&T and ZD&T 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 software-based license server update file. For more help, click the column headings in the table.
12. Click **Generate** at the bottom of the page. A window opens for you to download the update file.
13. To save the file, click **Download** for each file that is 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.

Applying the update file to the Software-based License Server

After you obtain the update file, you need to apply it to the Software-based License Server to enable the license server.

1. Log in to the computer. If you are not logging in as the root user ID, enter `su` that is followed by the root password.
2. Change to the `/opt/IBM/LDK` directory:

```
cd /opt/IBM/LDK
```

3. Run the following command:

```
./update_license <path of the updated file>
```

This file must have downloaded in your machine at the steps of [“For Software-based License Server” on page 143](#). This command produces several messages that indicate that the update is successful.

4. Use the root user ID to run the following command:

```
/opt/IBM/LDK# ./query_license
```

Then, you can see the number of available CPs. For example, the output is shown as below.

The following key is available:

```
HASP-SL key_id=432975633343422885 feature(s):
FID Feature Name      Expiration      Logins MaxLogins
333- CPU License     Tue Oct 30,2018 19:59:59    39  100
334- ADCD License     Tue Oct 30,2018 19:59:59     0   1
```

Host Information: zdt-lic-mgr localhost

These are the currently active sessions:

KeyID	FID	FeatureName	Address	User	Machine
432975633343422885	333	CPU License	9.26.158.161	ibmsys1	zdt-dev4 Mon
3					

- Optional: go to `/usr/z1090/bin`, and run the following command from the non-root user ID (ibmsys1).

```
./uimserverstart
```

Returning a software-based license key

All license key files expire after one year from the date they were generated, or after less than a year if the 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. Also, if you need to move, update or uninstall the Software-based License Server, you must first return the activated license key.

Returning the license key less than 31 days before expiration

To generate a new license key file for a Software-based License Server, 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 Software-based License Server. You do not need to return the physical software-based license server 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.

To return a software-based license less than 31 days, complete the following steps:

- From the root user ID on the Software-based License Server, run the following 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 Software-based License Server, and is used to generate the software-based license key file.

- If needed, transmit the `Hostname_XXXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.
- Log in to the [Rational License Key Center](#), and select your account.
- On the left side of the screen, select **View Keys by Host**.
- Select the **Hostname** corresponding to the license key you want to return.
- A table is displayed with data for the **Hostname** selected. At the far right of the table, click the **Change** link.
- The interface displays a list of the license keys for this license entitlement. Locate the license key of the Software-based License Server you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXXX.zip` file that you just created.
- Click **Initiate Return**.
- A message is displayed to confirm that the license entitlements were successfully returned.

Returning the license key more than 30 days before expiration

To generate a new license key file for Software-based License Server, 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 Software-based License Server. You do not need to return the physical software-based license server 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 Software-based License Server as part of the return process. This process consists of these general steps:

- Create a software-based license server request file from the Software-based License Server 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 Software-based License Server.
- 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. You can also use the **Return Keys** link.

To return a software-based license less than 30 days, complete the following steps:

1. Create a software-based license server request file from the Software-based License Server where the license key file is installed:
 - a. From the root user ID on the Software-based License Server, 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 Software-based License Server, and is used to generate the software-based license key file.
 - b. If needed, transmit the `Hostname_XXXXXXXXXX.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](#), 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 Software-based License Server you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXXX.zip` file that you just created.
 - f. Click **Initiate Return**.
 - g. A window opens, prompting you to download the software-based license server update file with the file to invalidate the license on the Software-based License Server.
 - 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 software-based license server update file that invalidates the license key to the Software-based License Server.
3. Install the license key file that invalidates the current license key on the Software-based License Server.
 - a. From the root user ID on the Software-based License Server, unzip the `Hostname_XXXXXXXXXX_update.zip` file, run the following command:

```
/opt/IBM/LDK/update_license Hostname_XXXXXXXXXX_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 the following command from the root user ID.

```
systemctl restart aksusbd.service
```

The Software-based License Server 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 Software-based License Server, run the following 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 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_XXXXXXXXXX.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](#), 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 Software-based License Server 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 Software-based License Server you are returning. Click **Browse**, and browse to and select the `Hostname_XXXXXXXXXX.zip` file that you just created.
 - f. Click **Complete Return**.
 - g. A message is displayed to confirm that the license entitlements were successfully returned.

Returning Rational Tokens for Software-based License Server

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 Software-based License Server. You do **not** need to return the physical update file to the Rational License Key Center.

1. Log in to [Rational License Key Center](#) and select your account.
2. On the left side of the screen, select **Return Keys**.
3. Select **IBM Rational Tokens**.
4. A list with license entitlements that are assigned to them from the same Order Line is displayed. Locate the host description you are working with, and click **Return**. A message is displayed to confirm that the license entitlements were successfully returned.

Updating the Software-based License Server

To update the Software-based License Server, you need to return the existing license first. Then, run the installer from the root user ID, and enable the license server again.

Before you update the Software-based License Server, you need to return the existing license key. For the steps to return a software-based license key, see [“Returning a software-based license key” on page 145](#).

To update the Software-based License Server, complete the following steps:

1. Open the directory that contains the installer `ZDT_Install_PS_V12.0.5.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V12.0.5.0.tgz
```

4. Optional: Read the `README.txt` file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

6. Select **1** for Software-based License Server.

7. Select **2** to update.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. After the update completes, run the following command to verify whether the update is successful.

- RHEL operating system

```
rpm -qa | grep aksusbd  
rpm -qa | grep zpdtuim
```

- UBUNTU operating system

```
dpkg -l | grep aksusbd  
dpkg -l | grep zpdtuim
```

Alternatively, after you complete steps 1 - 4, you can update the Software-based License Server silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --update --zdtswlicense
```

After the Software-based License Server is updated, you need to enable the license server by running the steps that are described in [“Enabling the License Server”](#) on page 139.

Uninstalling the Software-based License Server

To uninstall the Software-based License Server, you need to return the license key, stop the Software-based License Server and UIM server first. Then, run the installer from the root user ID.

Before you uninstall the Software-based License Server, you need to complete the following steps first.

1. Return the existing license key by running the steps that are described in [“Returning a software-based license key”](#) on page 145.
2. Stop the Software-based License Server by using the following command from the root user ID.

```
service aksusbd stop
```

3. Stop the UIM server by using the command from a non-root user ID.

```
cd /usr/z1090/bin  
./uimserverstop
```

To uninstall the Software-based License Server, complete the following steps:

1. Run the installer from the root-user ID.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

2. Select **1** for Software-based License Server.
3. Select **3** to uninstall.
4. After the uninstallation completes, run the following command to verify whether the uninstallation is successful.

For the RHEL operating system, run the command.

```
rpm -qa | grep z1091
```

For the UBUNTU operating system, run the command.

```
dpkg -l | grep z1091
```


Alternatively, you can uninstall the Software-based License Server silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --uninstall --zdtswlicense
```

Hardware-based licensing

If you purchased hardware-based license or hardware-based license with Rational Tokens, you need to enable a hardware license server to operate ZD&T Parallel Sysplex and authorize the licensee to use the product or any of its components. To enable the hardware license server, you can follow the steps that are provided in the checklist.

Before you enable the hardware-based licensing, you need to know the terms that are used in the enablement process.

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.



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 ZD&T 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.

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 ZD&T capable of operating within only one of the following parameters:

- 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.

Installing the Hardware-based License Server

To install the Hardware-based License Server, run the installer from the root user ID.

1. Open the directory that contains the installer ZDT_Install_PS_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

6. Select **3** for the Hardware-based License Server.
7. Select **1** to install.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter Yes to accept or No to decline the terms.
9. Optional: Select **Y** to install all needed dependencies or select **N** to decline.

By selecting **Y**, all required dependencies will be installed. The list of dependencies are mentioned in the “Prerequisites” on page 134. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

10. After the installation completes, run the following command to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep z1091
```

- UBUNTU operating system

```
dpkg -l | grep z1091
```

If the package is installed successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Important: The installer creates the `ibmsys1` as the non-root user ID to make ZD&T work, you need to switch to `ibmsys1` and change the password. If you like to create the non-root user ID, use the following commands for reference. Make sure that the non-root user ID is a part of group `zpdt`.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpdt -m ibmsys2
```

To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

Alternatively, after you complete the steps 1 - 4, you can install the Hardware-based License Server silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --install --zdthwlicense
```

To install the dependencies silently, run the following command:

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --install --zdthwlicense --dep
```

Obtaining an update file

To enable a Hardware-based License Server, you must first obtain a unique update file for the Hardware-based License Server.

For Hardware-based License Server

Learn about the steps to obtain an initial update file for the Hardware-based License Server.

1. Log in to the [Rational License Key Center \(RLKC\)](#), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
2. On the left side of the screen, Click **Get Keys**.
3. Select the product line for **IBM Rational Developer for System z Unit test**.
4. Select the license type that you want to apply to the key. If you purchased different license types, select **IBM RATIONAL DEVELOPMENT AND TEST ENVIRONMENT FOR Z SYSTEMS WITH PARALLEL SYSPLEX PER RVU FOR Z SYSTEMS LIC** . Then, one or more boxes are displayed to show your 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 ZD&T 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.

5. Check the boxes next to the orders from which you would like to use licenses, and click **Next**.
6. Complete the table that is displayed. You must enter the USB hardware device serial numbers that you want to activate, enter a number in the **Emulated Central Processor** field. For more help, click the column headings in the table.
7. Click **Generate** at the bottom of the page. Then, a screen is displayed to download the update files.
8. Click **Download** for each update file that is generated to save the update file. Keep this file in the Linux machine where you have installed the ZD&T Hardware License Server. 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: Use the binary transfer mode, but not a text mode such as ASCII. Also, spaces are not allowed in the path on the Linux machine.

For Hardware-based License Server with Rational Tokens

Learn about the steps to obtain an update file for Hardware-based License Server with Rational Tokens.

1. Log in to [Rational License Key Center \(RLKC\)](#), and select your account. If you do not have access to RLKC, click **Don't have a password**, and fill the form with the information in your purchase order.
2. On the left side of the screen, click **Get Keys**.
3. Select the product line for **IBM Rational Tokens**.
4. On the **Select License Keys** screen, one or more boxes for orders are displayed. Find the order that contains IBM Z Development and Test Environment. Make a note of the part number for the ZD&T 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.
6. Click **Generate** to generate the Rational Token license file. A screen with the generated license information is displayed.
7. Click **Download License Keys** to download the Rational Token license file to your computer. This token license file 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 to show a table that you must complete with the USB hardware device serial numbers and associated ZD&T 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 ZD&T and ZD&T 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. A screen to download the update files is displayed.
11. Click **Download** for each update file to save the update file. 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, 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.

Applying the update file to the Hardware-based License Server

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.

Note:

1. Never set the system clock to a future date or time when the USB hardware device is plugged in. Otherwise, the hardware device is damaged.
2. If the USB hardware device is attached to a computer with a clock that is set to a future date, the device remembers the future date. When the time is set back to the current time, the USB device does not recognize this change, and will not work until the clock reaches the previously set future time. If the clock was set to a future time or date, which causes that ZD&T is unable to authenticate with the USB hardware device, 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).

To apply the update file to the ZD&T Parallel Sysplex machine, complete the following steps:

1. Connect the USB hardware device to the computer that is running the Hardware-based License Server.

Note: Connect only the USB hardware device that is being activated to your computer during this process. Otherwise, unpredictable results might occur.

2. Log in to the computer. If you are not running as the root user ID, enter `su` that is followed by the root password.
3. Change to the `/usr/z1090/bin` directory:

```
cd /usr/z1090/bin
```

4. Run the following command where `update-file.zip` is the full path and file name of the update file.

```
./Z1091_token_update -u update-file.zip
```

This command produces several messages that indicate that the update is successful.

5. After the update file is successfully applied, unplug the USB hardware key, and wait for at least 10 seconds. Then, reconnect the hardware key.
6. Restart the local SHK license server from the root user ID by using the following command:

```
/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart
```

This step can ensure that the system discovers the newly applied licenses.

7. Optionally, verify that the licenses are now available on your USB hardware device by using the command:

```
./Z1091_token_update -status
```

Tip: After ZD&T is started, you can verify the effective Z serial number, and the expiration date of the license.

8. Go to `/usr/z1090/bin`, and run the following command from the non-root user ID (ibmsys1).

```
./uimserverstart
```

Returning a hardware-based license key

USB hardware device activations expire after one year from the date they were generated, or after less than a year if the 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 return a hardware-based license key or Rational Tokens for Hardware-based License Server, and renew the USB hardware device activation each year.

Returning an existing hardware-based license key

For perpetual license entitlements, the hardware-based license are set to expire one year from the date that 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 that 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 the term end. After the 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. 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 way is to use the **View Keys by Host** link. You can also use the **Return Keys** link.

For ZD&T, 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 \(RLKC\)](#), 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 ID** column.
4. A table is displayed with data for the selected USB hardware device. 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 that you are working with, and click **Return**. A message is displayed to confirm that the license entitlements were successfully returned.

Returning Rational Tokens for Hardware-based License Server

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.

1. Log in to [Rational License Key Center \(RLKC\)](#), and select your account.
2. On the left side of the screen, select **Return Keys**.
3. Select **IBM Rational Tokens**.
4. A list of devices with license entitlements that are assigned to them from the same Order Line is displayed. Locate the 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.

Updating the Hardware-based License Server

To update the Hardware-based License Server, you need to return the exiting license first. Then, run the installer from the root user ID to update the Hardware-based License Server, and enable the license server again.

Before you update the Hardware-based License Server, you need to return the exiting license key. For the steps to update the Hardware-based License Server, see [“Returning a hardware-based license key” on page 153](#).

To update the Hardware-based License Server, complete the following steps:

1. Open the directory that contains the installer `ZDT_Install_PS_V12.0.5.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PS_V12.0.5.0.tgz
```

4. Optional: Read the `README.txt` file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

6. Select **3** for Hardware-based License Server.
7. Select **2** to update.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.

If the package is updated successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Alternatively, after you complete steps 1 - 4, you can update the Hardware-based License Server silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --update --zdthwlicense
```

After the Hardware-based License Server is updated, you need to enable the license server by running the steps that are described in [“Enabling the License Server” on page 139](#).

Uninstalling the Hardware-based License Server

To uninstall the Hardware-based License Server, you need to return the license key, stop the UIM server and Hardware-based License Server first. Then, run the installer from the root user ID.

Before you uninstall the Hardware-based License Server, you need to complete the following steps first.

1. Return the exiting license key by running the steps that are described in [“Returning a hardware-based license key” on page 153](#).
2. Stop the UIM server by using the command from a non-root user ID.

```
cd /usr/z1090/bin
./uimserverstop
```

3. Stop the Hardware-based License Server from the root user ID by using the following command.

```
cd /opt/safenet_sentinel/common_files/sentinel_keys_server
./loadserv stop
```

To uninstall the Hardware-based License Server, complete the following steps:

1. Run the installer from the root user ID.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

2. Select **3** for Hardware-based License Server.
3. Select **3** to uninstall.

Alternatively, you can uninstall the Hardware-based License Server silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --uninstall --zdthwlicense
```

Installing Parallel Sysplex

To install ZD&T Parallel Sysplex, run the installer from the root user ID.

1. Open the directory that contains the installer ZDT_Install_PS_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PS_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

6. Select **2** for ZD&T Parallel Sysplex.
7. Select **1** to install.

8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. Optional: Enter y to install all needed dependencies or select n to decline.

By entering y, all required dependencies will be installed. The list of dependencies are mentioned in the “Prerequisites” on page 134. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

10. After the installation completes, run the following command to verify whether the installation is successful.

- RHEL operating system

```
rpm -qa | grep z1091
```

- UBUNTU operating system

```
dpkg -l | grep z1091
```

If the package is installed successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Important: The installer creates the `ibmsys1` as the non-root user ID to make ZD&T work, you need to switch to `ibmsys1` and change the password. If you like to create the non-root user ID, use the following commands for reference. Make sure that the non-root user ID is a part of group `zpdtd`.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpdtd -m ibmsys2
```

To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

Alternatively, after you complete steps 1 - 4, you can install ZD&T Parallel Sysplex silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --install --zdtps
```

To install the dependencies, run the following command:

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --install --zdtps --dep
```

Activating ZD&T Parallel Sysplex instances

This topic describes the steps to activate ZD&T Parallel Sysplex client to obtain the license from Software-based License Server or Hardware-based License Server.

Software-based license

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

Procedure

To activate ZD&T instances, complete the following steps:

1. Login as the root user ID, and enter the following command:


```
# /usr/z1090/bin/gen2_init
```

This setup command needs to be done only once.

It performs several actions:

- a. The Software-based License Server 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 Software-based License Server client.
 - b. The Software-based License Server client is started. The Software-based License Server 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 hardware-based license or a different Software-based License Server, 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 Software-based License Server.
 - b. Set **Gen2 BackupServer** field if you want to set up Software-based License Server 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. Enter the following command from /usr/z1090/bin.

```
./query_license
```

If the information of Software-based License Server is not displayed, complete the following steps:

- a. Go to /etc/hasplm/hasplm.ini and edit the file, and enter 1 to the value of aggressive mode and 0 to broadcast mode.
 - b. Enter `service aksusbd restart` command.
 - c. Wait for 5 minutes, and enter `./query_license` command to verify the information of the license server.
5. 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>
```

`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.

Hardware-based license

The computers that authenticate from the Hardware-based 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 the machine where you have installed Parallel Sysplex.
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 Server.
 - b. Leave **Gen2 BackupServer** empty. This field is used for Software-based License Server.
 - c. Set **Gen1 Contact Server** field to point to IP address or hostname of Hardware-based License Server.
 - d. Set **Gen1 BackupServer** if you want to set up the failover Hardware-based License Server. Set the field to point to IP address or hostname of failover Hardware-based License Server.
 - e. Set **UIM ContactServer** if you want to use different UIM server than Hardware-based 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>
```

`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.

Configuring Z instances

You need to set up the ADCD package and create the device map to configure Z instances for development and test purposes.

Setting up the ADCD and z/VM package

Before you configure Z instances, set up the ADCD and z/VM package.

When you run the installer, a non-root user ID (ibmsys1) is created. The user ID is required to enable the z/OS images. You can also create other non-root user IDs.

To set up the ADCD package, complete the following steps:

1. Switch to the non-root user ID (ibmsys1).
2. Create a folder that is named `volumes` under `/home/ibmsys1`, and keep all ADCD and z/VM volumes in the `/home/ibmsys1/volumes` directory that you downloaded at step 2 of “Checklist” on page 131.
3. Make sure that all volumes are owned by the non-root user ID. You can use the following command to change the owner and group.

```
chown ibmsys1 *
chgrp zpdt *
```

Note: Make sure that the folder that contains *.gz and .ZPD volumes are also owned by the non-root user ID.

4. Decompress all .gz volumes by using the Linux command.

```
gunzip *.gz
```

5. Decrypt .ZPD volumes by using the command from `/usr/z1090/bin`.

```
./Z1091_ADCD_install <path of volume>/A4RES1.ZPD <output path of volume>/A4RES1
```

For example, `./Z1091_ADCD_install /home/ibmsys1/volumes/A4RES1.ZPD /home/ibmsys1/volumes/A4RES1`.

6. Make sure that all decompressed and decrypted files are executable. Otherwise, enter the following command:

```
chmod 755 *
```

For a complete list of z/OS products in ADCD, see “ADCD z/OS V2R4 November Edition of 2019” on page 178.

Creating the device map

A sample program `create_devmap.pl` is available in the `/opt/ConfigGuideSample` directory where you have installed ZD&T.

To create the device map from a non-root user ID, run the following command:

```
perl <path to command>/create_devmap.pl <path to disks> > <generated devmap>
```

For example, `perl create_devmap.pl /home/ibmsys1/volumes > generateddevmap`

If you create the devmap manually, or edit the devmap that is created by the `create_devmap.pl` utility, you can run the following command from `/usr/z1090/bin` to verify it after you create the devmap manually.

```
awsckmap /home/ibmsys1/volumes/generateddevmap
```

Then, open the devmap and add `cpuopt zvm_couplingfacility` in the `[system]` stanza.

If you are using Rational token with hardware-based or software-based license, add the following information to the `[system]` stanza.

```
rdtserver 27000@rational token (RLKS) hostname/IP Address
```

For more information on the sections and syntax of device maps, see the "1090 Control Files" section of the [zPDT Guide and Reference](#).

Starting z/VM

To start z/VM, issue `awsstart` under `ibmsys1` user ID or equivalent home by using directory structure `/usr/ibmsys1`.

Restriction: You cannot use the `runzpd` script when you perform a sysplex activation.

To start z/VM, complete the following steps:

1. Switch to a non-root user ID.
2. Switch to the path where you defined your devmap. For example, if your devmap is located in `/usr/ibmsys1/volumes`, enter the following command:

```
cd /usr/ibmsys1/volumes
```

3. Enter the following command.

```
awsstart generateddevmap
```

`generated devmap` is the name of your devmap.

4. Open the terminal emulator, and enter the IP and port (3270) of the machine where you have installed Parallel Sysplex to configure the emulator. If you use Linux 3270, use the following command to start.

```
x3270 -port 3270 <session_name_in_devmap>@localhost &
```

5. Open the devmap, and find the address of system residence volume for z/VM (M01RES).
6. IPL the z/VM system by entering the following command, where address 200 is the device address in the devmap of the system residence volume for z/VM (M01RES).

```
Ipl 200 parm 0700
```

7. When the 3270 session is switched to operator console as shown below, press F10 and escape key to next page. Then, enter the address of VMCOM1 from DEVMAP at the `pdvol=` in the master console screen

```
STAND ALONE PROGRAM LOADER: z/VM VERSION 6 RELEASE 4.0
-----
DEVICE NUMBER:  0200      MINIDISK OFFSET:  39      EXTENT:  1
-----
MODULE NAME:    CPLoad    LOAD ORIGIN:    1000
-----
-----IPL PARAMETERS-----
pdvol=201 pdnum=1 paging63
-----
-----COMMENTS-----
-----
-----
9= FILELIST  10= LOAD  11= TOGGLE EXTENT/OFFSET
```

8. Enter force if it is required in the IPL process.

```
12:42:46 HCPASK9205A To change to a FORCE start, enter FORCE.
12:42:46 HCPASK9205A To stop processing, enter STOP.

force

CP READ  ZVM640
```

9. Open another terminal emulator sessions (#701), and enter Enable All in the master console of z/VM with the device address 0700.



In case the connection to master console is lost, or the master console is not displayed after you issue **awsstart** command, issue the following commands in the Linux machine where you issued the **awsstart** command by using the non-root user ID.

```
oprmsg 'VARY CN(*),ACTIVATE'
oprmsg 'vary 0700,console'
```

Starting Z instances

You need to start Z instances from an active 3270 session that is connected to z/VM.

To start Z instances, complete the following steps:

1. When the z/VM screen is displayed, enter a valid z/VM user ID and password. For example, use BASEAD as the user ID and ZVM640 as the password.
2. Enter the following command.

```
TERM CONMODE 3270
```

3. Enter IPL command.

```
ipl 0a80 loadparm 0a82xx
```

xx is the loadparm. For example, `ipl 0a80 loadparm 0a82ws`.

4. Open another 3270 terminal emulator session. Then, enter the following command.

```
dial <userid>
```

<userid> is the z/VM userid used to IPL Z instances.

5. When the z/OS logon screen is displayed, enter the ADCD user ID to log on.

Stopping Z instances and z/VM

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 ZD&T Parallel Sysplex contains sample startup and shutdown scripts for the systems that are available and active in that distribution.

You need to remember the IPL loadparm that is used to start ZD&T Parallel Sysplex. To shut down ZD&T Parallel Sysplex, use the respective shutdown command. Find all shutdown scripts are in the ADCD.Z24A.PARMLIB(SHUT**).

To shut down Z instances, complete the following steps:

1. Enter S SHUTXX in the master console.

XX is the last 2 characters of shutdown script that is generally equivalent to loadparms. To reply to any pending message, enter this command from the master console. If you do not have a master console, enter /S SHUTXX in SDSF, or enter `oprmsg "S SHUTXX"` in the Linux machine.

2. During the shutdown process, you might need to respond to z/OS console messages, for example, when IMS, TSO, or z/OS UNIX are stopped. To check which programs are still running, enter D J, L. Ensure that VTAM and all subsystems are ended.
3. Enter \$P JES2 in the master console to close JES2.
4. Enter QUIESCE in the master console.
5. Enter SHUTDOWN in the z/VM operator console.
6. Enter the command `awsstop` in the Linux machine where you run IPL command for z/VM by using the non-root user ID (ibmsys1) to completely shut down ZD&T Parallel Sysplex.

Note: Do not make any change in the ADCD.Z24A.* library. If you want to edit any parm or proc, ADCD has USER.Z24A.* as a high-level concatenated library. You can create your own proc or parm in the library.

Updating Parallel Sysplex

If you installed ZD&T V11.0 or later versions, you can update Parallel Sysplex directly to a new version. If you installed ZD&T V10.0.1 or earlier versions, you need to uninstall the earlier version before you install the current Parallel Sysplex.

Before you update Parallel Sysplex, you need to return the exiting licenses.

- For the steps to return the software-based license, see [“Returning a software-based license key”](#) on page 145.
- For the steps to return the hardware-based license, see [“Returning a hardware-based license key”](#) on page 153.

Uninstalling the earlier versions of ZD&T

If you installed zD&T V10.0.1 or earlier versions, use the steps below to uninstall the previous versions..

To uninstall any supported version, complete the following steps.

Version 9.0 or above	<ul style="list-style-type: none"> • Stop any currently active instance of RD&T. <ul style="list-style-type: none"> – To stop a base instance of RD&T, see “Stopping Z instances ” on page 22. – If you are running a product license server: <ol style="list-style-type: none"> 1. Stop the UIM server by using the command from a non-root user ID. <pre data-bbox="951 401 1185 453">cd /usr/z1090/bin ./uimserverstop</pre> 2. Enter these two commands to stop the product license server: <pre data-bbox="963 590 1390 663"># cd /opt/safenet_sentinel/ common_files/sentinel_keys_server # ./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 x 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 ./'Uninstall IBM Rational Development and Test Environment for z Systems V8.5.x'. 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 x is the version that is installed. • Go to the subdirectory entitled Uninstall_Rational Developer for z Unit Test V8.0.x. • Run the command ./"Uninstall_Rational Developer for z Unit Test V8.0.x". 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".
-------------	--

After the uninstallation completes, you can install the new version of ZD&T by completing the steps that are described in [“Installing Parallel Sysplex” on page 155](#).

Note: ZD&T Parallel Sysplex doesn't support older version of zD&T or RD&T. So, to use ZD&T Parallel Sysplex, you need to uninstall older version of ZD&T and install ZD&T V12.0.

Uninstalling a version of zD&T or RD&T 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 ZD&T 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 information about uninstalling the Software-based License Server, see [“Uninstalling the Software-based License Server” on page 148](#).

Updating the installed Parallel Sysplex

If you installed ZD&T V11.0 or above version, run the installer from the root user ID to update ZD&T Parallel Sysplex directly.

1. Open the directory that contains the installer ZDT_Install_PS_V12.0.5.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V12.0.5.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V12.0.5.0.tgz
```

4. Optional: Read the README.txt file for the complete installation steps.
5. Run the installer.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

6. Select **2** for ZD&T Parallel Sysplex.
7. Select **2** to update.

8. Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
9. After the update completes, run the following command to verify whether the update is successful.
 - RHEL operating system

```
rpm -qa | grep z1091
```

- UBUNTU operating system

```
dpkg -l | grep z1091
```

If the package is updated successfully, the following output is displayed.

```
z1091-1-10.55.04.x86_64
```

Alternatively, after you complete steps 1 - 4, you can update ZD&T Parallel Sysplex silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --update --zdtps
```

Uninstalling Parallel Sysplex

To uninstall Parallel Sysplex, you need to return the license key, and uninstall the License Server first. Then, run the installer from the root user ID.

To uninstall ZD&T Parallel Sysplex, complete the following steps:

1. Stop the z/VM and Z instances. For more information, see [“Stopping Z instances and z/VM” on page 163](#)
2. Run the installer from the root user ID.

```
./ZDT_Install_PS_V12.0.5.0.x86_64
```

3. Select **2** for ZD&T Parallel Sysplex.
4. Select **3** to uninstall.

Alternatively, you can run uninstall ZD&T Parallel Sysplex silently by using the following command.

```
./ZDT_Install_PS_V12.0.5.0.x86_64 --uninstall --zdtps
```

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 ZD&T](#).

For a complete listing of the currently available technotes, see [ZD&T 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 ZD&T](#).

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 [ZD&T Forum](#).

Obtaining the latest editions of Redbooks

IBM Redbooks® provide information about major components of ZD&T.

ZD&T

To obtain the latest edition of the *zPDT Guide and Reference (SG24-8205)* for use with ZD&T, go to the [IBM Redbooks website](#) and search for *zPDT Guide and Reference (SG24-8205)*.

Troubleshooting tips

Symptom	Problem Determination
For any problem after installation of ZD&T (creating a devmap, installing z/OS, starting or stopping ZD&T, 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 ZD&T.	Any message beginning with AWS is a zPDT message. Enter the following zPDT command <pre>msgInfo message-number</pre> 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 .

Installing ZD&T

Table 32. Troubleshooting the Installation of the ZD&T	
Symptom	Problem Determination
Message CRIMA1076E in the install log when you are uninstalling a version of ZD&T	For the step-by-step process to perform if an uninstall fails., see “Uninstalling the earlier versions of ZD&T” on page 27.
<p>When you are attempting to run launchpad.sh or launchpad-console.sh, you get the following error:</p> <pre>./InstallerImage_linux.gtk.x86_64/ tools/imcl: Permission denied</pre>	<p>This issue can happen for several reasons:</p> <ul style="list-style-type: none"> You are attempting the launchpad command without running as root. Ensure that you are executing this command as root 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.
<p>When you are attempting to perform an Installation Manager update for ZD&T, or to run launchpad.sh or launchpad-console.sh, 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"> If these messages are from an attempt to migrate to a new version of ZD&T 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 the earlier versions of ZD&T” on page 27. If these messages are from running launchpad.sh or launchpad-console.sh, 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 the earlier versions of ZD&T” on page 27.

Table 32. Troubleshooting the Installation of the ZD&T (continued)

<p>You receive any error message when you are uninstalling a previous version of ZD&T using the steps outlined in “Uninstalling the earlier versions of ZD&T” on page 27.</p>	<ol style="list-style-type: none"> 1. If you are uninstalling Version 9.0 or 9.1, run the <code>z1091_removeall</code> command <ul style="list-style-type: none"> • Start a Linux console as user root. • Change Directory (<code>cd</code>) to <code>/usr/z1090/bin</code>, the folder that contains the <code>z1091_removeall</code> command. • Run the command as follows: <pre>./z1091_removeall</pre> • Reply Yes to any prompts that ask if you are sure that you want to uninstall 2. If you still get some form of error messages, such as failure to uninstall the Sentinel Keys Server (<code>zpd-t-shk-server</code>) or Sentinel System Driver (<code>sntl-sud</code>), attempt these steps in the following order: <ul style="list-style-type: none"> • Enter <code># rpm -e --allmatches z1090</code> • Enter <code># rpm -e --allmatches z1091</code> • Enter <code># rpm -e --allmatches zpd-t-shk-server</code> • Enter <code># rpm -e --allmatches sntl-sud</code>
<p>When an installation fails for the ZD&T Software-based License Server, and the log displays the following entries:</p> <pre>error: Failed dependencies: libc.so.6 is needed by aksusbd-7.40-1.i386 libc.so.6(GLIBC_2.0) is needed by aksusbd-7.40-1.i386 libc.so.6(GLIBC_2.1) is needed by aksusbd-7.40-1.i386 libc.so.6(GLIBC_2.2) is needed by aksusbd-7.40-1.i386 libc.so.6(GLIBC_2.3.4) is needed by aksusbd-7.40-1.i386 libc.so.6(GLIBC_2.4) is needed by aksusbd-7.40-1.i386 libpthread.so.0 is needed by aksusbd-7.40-1.i386 libpthread.so.0(GLIBC_2.0) is needed by aksusbd-7.40-1.i386 libpthread.so.0(GLIBC_2.1) is needed by aksusbd-7.40-1.i386 libpthread.so.0(GLIBC_2.3.2) is needed by aksusbd-7.40-1.i386</pre>	<p>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.</p>

USB Hardware device activation and licensing

Troubleshooting the USB Hardware device activation and licensing

Table 33. USB Hardware device activation and licensing. Troubleshooting USB Hardware device activation and licensing

<p>Symptom</p>	<p>SecureUpdateUtility or Z1091_token_update does not apply the update file to the USB hardware device.</p>
-----------------------	---

Table 33. USB Hardware device activation and licensing. Troubleshooting USB Hardware device activation and licensing (continued)

Problem Determination	<ul style="list-style-type: none"> • 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. • Ensure that the hardware device is recognized by your Linux system by plugging in the device and issuing the <code>lsusb</code> command. If the device is available, you see an entry that shows "Rainbow Technologies Inc".
Symptom	Unit not found message that is issued during the <code>SecureUpdateUtility -u</code> or the <code>z1091_token_update</code> command
Problem Determination	<p>This issue can happen for several reasons:</p> <ul style="list-style-type: none"> • You try to apply an update file with one serial number to a USB hardware device with a different serial number. • 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 <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>
Symptom	<p>When you are starting ZD&T, (<code>awsstart</code>) 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"> • <pre>Error: Failed to get license. Return code: 312</pre> • <pre>Error: Unable to get expected license: xxxx</pre> <p>or</p> <pre>Error: Unable to locate expected license: xxxx</pre> <ul style="list-style-type: none"> • <pre>AWSEMI315E zPDTA License Unavailable for CPU 0 AWSEMI005I Waiting for 1090 license</pre> • <pre>CPU Address Out Of Range</pre> • <pre>SFNTGetLicense failed. Return code: 312</pre>

Table 33. USB Hardware device activation and licensing. Troubleshooting USB Hardware device activation and licensing (continued)

<p>Problem Determination</p>	<p>This issue can happen for several reasons:</p> <ul style="list-style-type: none"> • 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. • 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"> – 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. – Ensure the <code>SecureUpdateUtility -u</code> or <code>Z1091_token_update -u</code> command responded with a "Success" message. • The requested license type is not correct. <ul style="list-style-type: none"> – Validate that if your device map includes the <code>cpuopt zvm_couplingfacility</code> directive, your USB hardware key includes <code>sysplex</code> enabling licenses. If no <code>sysplex</code> license is available, your device map cannot enable <code>sysplex</code> operation. – 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 ZD&T instance cannot participate in a Rational Token licensing system. • 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. • 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"> – 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 <code>License ContactServer</code> is <code>localhost</code> if you are attempting to use a local USB. No other fields need be completed. • 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. • You do not have the correct type of USB Hardware Device. <ul style="list-style-type: none"> – Perform the following command from a Linux terminal that is running as root: <pre style="background-color: #f0f0f0; padding: 5px;">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</p> <pre style="background-color: #f0f0f0; padding: 5px;">z1091-xxx and not z1090-xxx</pre> – 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.
-------------------------------------	--

<p>Problem Determination (continued)</p>	<ul style="list-style-type: none"> • You do not have connectivity from the instance that is performing the <code>awsstart</code> to the product license server. <ul style="list-style-type: none"> – Enter a <code>UIMCHECK</code> on the client instance to verify whether there is proper connectivity. You see the message <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <pre>The remote QUERY command executed successfully</pre> </div> – If this procedure does not complete successfully, try the following procedure. <ul style="list-style-type: none"> - 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 <code>License ContactServer</code> is the IP address of the product license server, that the <code>License PortNumber</code> is 9450, and that <code>License ipv6</code> is yes or no as is appropriate for your installation. - From a Linux terminal on the instance that is running ZD&T, 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"> • Ensure that the server was started once by a non-root ID with the <code>uimserverstart</code> command. - 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 <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <pre>iptables -L -n</pre> </div> <p>Ensure ports 9450 and 9451 are allowed.</p> - 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 <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <pre>/opt/safenet_sentinel/common_files/sentinel_keys_server/sntlconfigsrv.xml</pre> </div> <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> <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <pre>/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart</pre> </div>
<p>Symptom</p>	<p>When you are starting Rational Development and Test Environment for z Systems, (<code>awsstart</code>) you get the following messages at the Linux terminal that indicate an expired license</p> <div style="background-color: #f0f0f0; padding: 5px; margin: 5px 0;"> <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> </div>

Problem Determination	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>ZD&T Activation Guide (SC27-6630)</i> .
Symptom	Time cheat errors
Problem Determination	<ul style="list-style-type: none"> 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.

Troubleshooting installing, operating, and uninstalling the Software-based License Server

Learn how to troubleshoot problems installing, operating, and uninstalling the Software-based License Server.

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

<i>Table 34. Trouble shooting the Installation of z/OS volume images</i>	
Symptom	Problem Determination
<p>You get the following errors while performing a <code>z1091_ADCD_install</code> 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 169.</p> <p>Also this can occur if the USB Hardware device, either local or in a product license server, has been updated with a <code>.upw</code> update file instead of a <code>.zip</code> update file. If you are migrating a ZD&T instance from version 9 to version 9.1, you must request a new update file from the Rational License Key Center to install the ADCD S1RES1 and SARES1 volumes.</p>

Defining the ZD&T machine characteristics (DEVMAP)

<i>Table 35. Defining the ZD&T machine characteristics (DEVMAP). Troubleshooting defining the ZD&T machine characteristics (DEVMAP)</i>	
Symptom	Problem Determination
Unable to run <code>find_io</code> command at <code>create_devmap.pl</code>	The path to the <code>find_io</code> command is not available. Ensure that you run <code>aws_bashrc</code> from the user ID that runs <code>create_devmap.pl</code> and ZD&T. This updates the path to the needed files.

Starting and stopping ZD&T

<i>Table 36. Starting and stopping ZD&T. Troubleshooting the starting and stopping of ZD&T</i>	
Symptom	Problem Determination
<pre> AWSSTA140E - Initialization failure after license obtained </pre>	<p>Ensure that you are starting ZD&T from the home directory of the user who is created for running ZD&T.</p> <p>Ensure that you have run the <code>aws_bashrc</code> command while logged on as that user, and from that user's home directory.</p>
<p>The following messages occur during startup:</p> <pre> AWSECH004S Unable to define RAS/FEDC memory, RC=-3. AWSECH004S Unable to define RAS/FEDC memory, RC=-3. </pre>	<p>This message is seen with many <code>awsckd</code> definitions.</p> <p>The Linux kernel setting <code>SHMMNI</code>, 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 <code>kernel.shmmni = 8192</code> to <code>/etc/sysctl.conf</code> and activate it using the command <code>sysctl -p</code>.</p>
<p>The following message occurs when <code>ipling z/VM 6.2</code></p> <p>“CP requires hardware features not available on this processor”</p>	<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.</p>

z/OS networking issues

<i>Table 37. z/OS networking issues . Troubleshooting z/OS networking issues</i>	
Symptom	General z/OS networking configuration errors

Table 37. z/OS networking issues . Troubleshooting z/OS networking issues (continued)

Problem Determination	<ul style="list-style-type: none"> • Check syslog (z/OS console messages) for errors during the start of TCP/IP or VTAM. • 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.
Symptom	Unable to establish a network tunnel on a 10 . x . x . x network.
Problem Determination	<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</p> <pre>192.168.1.2 netmask 255.255.255.0</pre>
Symptom	Cannot connect to ZD&T z/OS using a remote 3270 terminal emulator
Problem Determination	<ul style="list-style-type: none"> • 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 • 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 • 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) • 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.
Symptom	Cannot ping to a z/OS defined IP address on the ZD&T instance

Table 37. z/OS networking issues . Troubleshooting z/OS networking issues (continued)

<p>Problem Determination</p>	<ul style="list-style-type: none"> • 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 • 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. <ul style="list-style-type: none"> – Defined a 2nd OSA, with an IP address in the same subnet as the IP address of the Linux ethernet adapter – Defined a default route through that OSA – Updated your TCPIP.DATA files to point to your domain name server and the host name for your z/OS – Updated your procedures to point to the new TCPIP.DATA files – Updated your IPNODES definitions to include the z/OS host name and IP address – Updated your resolver parms to point to the new IPNODES definitions – Updated your VTAM definitions to add the tunnel OSA and any other OSAs defined • 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"> – Check your firewall configuration with the command <pre style="background-color: #f0f0f0; padding: 5px;">iptables -L -n</pre> – 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 – Check your routing definitions in BEGINROUTES or OMPROUTE. Ensure the <code>firsthop</code> address is the Linux tunnel address if all traffic is being routed through the Linux. – For more information, see Chapter 7 “LANs” in the zPDT Guide and Reference.
<p>Symptom</p>	<p>Cannot FTP or Telnet to a z/OS defined IP address on the ZD&T instance</p>
<p>Problem Determination</p>	<ul style="list-style-type: none"> • Verify that you can ping to the address from the remote location. If you cannot, follow the steps defined here. • If you can ping, check your firewall configuration with the command <pre style="background-color: #f0f0f0; padding: 5px;">iptables -L -n</pre> <ul style="list-style-type: none"> – Ensure that the appropriate ports are allowed; typically ports 21 for FTP and 23 for Telnet

Performance

Table 38. Performance. Troubleshooting performance issues

<p>Symptom</p>	<p>High volume traffic through an emulated OSA-Express ethernet adapter on ZD&T suffers poor throughput.</p>
-----------------------	--

Table 38. Performance. Troubleshooting performance issues (continued)

<p>Problem Determination</p>	<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 data-bbox="399 369 1466 646"> # 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 ZD&T, such as the runzpdT script shipped with the product, that script can be modified to enter one or more of these commands each time ZD&T 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 data-bbox="399 1045 1466 1100"> 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>
<p>Symptom</p>	<p>ZD&T performs slowly. CPUs reach 100% utilization.</p>
<p>Problem Determination</p>	<p>Verify that you are meeting the minimum requirements for both the hardware and the software. See "ZD&T prerequisites" and paragraph 2.3.2 "zPDT instances" in the zPDT Guide and Reference.</p>

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 feutlicm log can be viewed with the less command while ZD&T 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 RDTLOG=TTY, if set before you start ZD&T, routes all Rational Token-related logging to the Linux terminal in addition to the log.

Variable RDTDEBUG=DEBUG, if set before you start ZD&T, adds more information to the logs. If RDTLOG=TTY is also set, these additional messages are also written to the Linux terminal.

Do not set RDTLOG=TTY in everyday use, because it sends frequent unsolicited messages to your Linux terminal. Setting RDTDEBUG=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 ZD&T renews them before that time. ZD&T renews tokens approximately every half hour.

When ZD&T 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 Software-based License Server and USB hardware device activation

Configuration and usage for ZD&T is described in [“Configuring Z instances” on page 20](#) and the [zPDT Guide and Reference](#).

Reference

The reference material in this section of the IBM Knowledge Center provides supporting information for the instructions for using ZD&T.

z/OS 2.4 ADCD reference

Learn about general release information of ADCD z/OS V2R4 November Edition of 2019, hints and tips, and the customization of new ADCD release for ZD&T.

ADCD z/OS V2R4 November Edition of 2019

Learn about the new release ADCD z/OS V2R4 November Edition of 2019.

To learn what's new in ADCD z/OS V2R4 November Edition of 2019 and the products that are contained in this release, see [ADCD z/OS V2R4 November Edition of 2019](#).

Note: The SMS section in the general release information is replaced by the SMS that is provided by ZD&T. For more information, see [“Customizing ADCD for ZD&T” on page 179](#).

Hints and tips to manage the z/OS 2.4 ADCD

Learn about the information that is regarding the contents of the ADCD z/OS V2R4 November Edition for 2019. Some additional notes and tips are provided to effectively manage the ADCD system. Also, the instructions for starting WebSphere Application Server, z/OSMF, Health Checker are included in this document.

For more information about the hints and tips, see [Application Developers Controlled Distribution \(ADCD\) Release Guide z/OS V2R4 November Edition of 2019](#).

Note: The following topics from [Application Developers Controlled Distribution \(ADCD\) Release Guide z/OS V2R4 November Edition of 2019](#) are changed.

- **ACS routines provided on ADCD**

The SMS configuration that is documented in [“Customizing ADCD for ZD&T”](#) on page 179 replaces the ACS routines that are provided by ADCD.

- **Allocating Db2 archive logs in to an esoteric device**

Esoteric devices are not required with the new SMS configuration. The ADCD volume D3DBAR was added to the SGARCH Storage Group. You do not need to define extra volumes to the Esoteric via IODF and HCD. If extra space is required for Db2 archive logs, you can easily add extra SMS volumes to the SGARCH or SGBASE Storage Groups. For more information, see [“Adding a volume to an SMS Storage Group”](#) on page 182.

- **Using the esoteric devices**

Esoteric devices are made obsolete by SMS because all the default allocations of data will be directed to SMS-managed volumes, and the esoteric specification in JCL might be overridden. With no need to remember and specify appropriate esoteric names for JCL DD statements, the requirement of coding JCL can be simplified. For more information, see [“Customizing ADCD for ZD&T”](#) on page 179.

Customizing ADCD for ZD&T

This section documents that the default SMS configuration that is provided with ADCD is replaced. The new SMS configuration that is added by ZD&T alters some of the procedures that are documented in the General Release information and ADCD Release Guide z/OS V2R4 November Edition of 2019.

SMS (System-Managed Storage) introduces a level of control on how data is allocated on the z/OS system. To handle the data set allocation, several SMS constructs are used. The new SMS configuration that is provided is configured for ADCD systems, and some SMS configuration is mandatory on all z/OS systems. However, the default setup of the ADCD system has a limited SMS configuration. Therefore, the following new SMS configuration that replaces the default ADCD setup is provided to be activated as the default for ZD&T.

This new SMS configuration can provide these additional enhancements:

- Prevent general JCL errors where the ADCD system failed to define Esoteric Device Table entries that might exist on your source systems.
- Eliminate the request to modify customer JCL to fit into ADCD where new data set allocations are required.
- Ensure that all customer data is allocated on non-ADCD system volumes, although the new SMS configuration is flexible enough that the data can be manually allocated if needed.
- Allow a separation of data based on the middleware or application (CICS, MQS, IMS, Db2, and other applications) that the data might belong to.
- Allow the easier portability of data because the data can be consolidated on the specific volumes.
- Allow the easier use of SMS features, such as EAV, VIO, and EXT attributes.
- Provide a more flexible method for experienced users to modify the configuration for their specific needs.

The new SMS Configuration Dataset (SCDS) is named SYS1.S0W1.DFSMS.SCDS, and the existing default ADCD configuration data set SYS1.S0W1.SCDS is retained to ensure that you can revert to the default configuration. The SMS configuration is encapsulated in both the SCDS and a PDS where ACS routines are sourced from.

Naming conventions of SMS constructs:

DC*	Data Classes
SC*	Storage Classes
SG*	Storage Groups

SMS constructs also include Management Class. However, these constructs are not used in the emulated environments.

SMS configuration ACS routines data set (SYS1.S0W1.DFSMS.CNTL)

The partitioned data set of this new SMS configuration contains the following members:

DATACLAS	ACS Routine to assign Data Class
STORCLAS	ACS Routine to assign Storage Class
STORGRP	ACS Routine to assign Storage Group
TEST****	ACS Testing members to verify that ACS routines assign correct constructs for the allocation of data and that data is directed to the expected Storage Group and disk volumes.

• Data Classes

Data Classes include the following members:

- DCDB2
- DCDB2EXT
- DCEATTR
- DCEXT
- DCEXTEAV
- DCLARGE
- DCLIB
- DCPDS
- DCPDSE
- DCSEQ
- DCZFS

Note:

- Data Classes describe allocation parameters for data set types and allow allocations to assume the defaults to fulfill an allocation. Then, you do not have to know which parameters are required.
- Data Classes can be used for non-SMS data sets. You can use Data Classes to allocate new data sets even if the data sets are ultimately on the non-SMS volumes.
- Parameters that are set in a Data Class can be used or overridden by user specifications.
- Some Data Classes might require that the data sets be placed on SMS-managed volumes.

• Storage Classes

Storage Classes include the following members:

- SCAPPL
- SCARCH
- SCBASE
- SCCICS
- SCCIMS
- SCDB2
- SCEXTEAV
- SCMQS
- SCNOSMS

- SCNOVIO

Note:

- Storage Classes are the key to SMS-managed data. A data set without a storage class is non-SMS managed.
- ACS routines are the rules that ultimately control the data placement. ACS routines can be set up to define whether the user specifications are accepted or not. For example, ACS routines can define whether you can use SCCICS for temporary data sets.
- In this configuration, Storage Classes dictate which disk volumes data is allocated, although the environmental conditions can alter the disk volumes data.
- In this configuration, the Storage Group names (SCxxx <> SGxxx) are easily discernible except SCNOSMS.
- SCNOSMS is a special Storage Class that can be specified by any user when the user wants to control the data set placement. To use SCNOSMS, you must specify the disk volume location for your data. Also, when you use SCNOSMS, this special class will signify SMS ACS routines not to assign a storage class, and the storage class becomes null. Therefore, no Storage Group exists.
- SCNOVIO is another special Storage Class that is set up to prevent VIO allocation for the data that does not perform well. What is VIO (Visual input/output)? VIO is to allocate data in the system memory instead of disk for any transient data. VIO can boost a good performance, particularly in the emulated environments where disk I/O is a larger bottleneck on the real hardware. As no Storage Class can explicitly force a VIO allocation, the ACS routines are set up. Then, all the temporary data sets that are less than 2, 000, 000 kilobytes are directed to VIO.
- Except SCAPPL, SCNOVIO, and SCNOSMS, the ACS routines can override a user specification. The ACS routines can be easily changed, and the user specification can be accepted by SMS.
- SCAPPL is intended to be used as the only class that users can access, SMS will not override the class that is specified by users. With SCAPPL, you can control the allocation of the application data, which can provide a way to facilitate portability.

• **Storage Groups**

Storage Groups include the following members:

- SGAPPL
- SGARCH
- SGBASE
- SGCICS
- SGDB2
- SGEXTEAV
- SGIMS
- SGMQS
- SGVIO

Note:

- Storage Groups dictate which disk volumes data is placed.
- This SMS configuration mainly uses the Storage Class as the determining factor to assign Storage Group. However, some exceptions exist.
- SGBASE is special in this configuration. This group acts as the secondary group for all other Storage Groups to avoid allocation failures when other Storage Group volumes are not present or not online.

The Storage Group Volume assignments are as follows.

SGCICS	CICS00 to CICS20 that includes 33 volumes (Hex)
SGMQS	MQS000 to MQS020 that includes 33 volumes (Hex)

SGAPPL	APPL00 to APPL20 that includes 33 volumes (Hex)
SGARCH	ARCH00 to ARCH20 that includes 33 volumes (Hex)
SGBASE	USER00 to USER4F that includes 81 volumes (Hex)
SGDB2	DB2000 to DB2040 that includes 65 volumes (Hex)
SGEXTEAV	EAV000 to EAV020 that includes 33 volumes (Hex)
SGIMS	IMS000 to IMS020 that includes 33 volumes (Hex)

Note: These volumes are not assigned by ADCD, and they need to be created by users.

Except for these volumes, some basic ADCD volumes are also assigned to Storage Groups because few SMS-managed volumes exist in the default ADCD configuration. These additional ADCD volumes are necessary to the Storage Groups. However, the ADCD volumes are defined to Storage Groups in a Quiesced State. The volumes that are in the Quiesced state will take on only new allocations when no other online volumes are in the Storage Group. The ADCD SMS volumes are shown as follows.

- In the SGBASE Storage Group:
 - D3USR1
 - A4USR1
- In the SGDB2 Storage Group:
 - B3DBB2 (ADCD May 2018 Edition)
 - B3DBC2 (ADCD May 2018 Edition)
 - C3DBB2 (ADCD November 2018 Edition)
 - C3DBC2 (ADCD November 2018 Edition)
 - D3DBB2 (ADCD May 2019 Edition)
 - D3DBC2 (ADCD May 2019 Edition)
 - A4DBB2 (ADCD November 2019 Edition)
 - A4DBC2 (ADCD November 2019 Edition)

Future Editions of ADCD volumes will be added to the Storage Group constructs.

Adding a volume to an SMS Storage Group

If a Storage Group is running low on the available free space, you can add a volume to the Storage Group. SGBASE Storage Group is defined as a destination for all other Storage Groups. You can always add volumes to this pool if you are unsure where to allocate extra space or if you do not have a preference to allocate the data.

To add a volume, complete the following steps:

1. Allocate the backing file in the hosting Linux system by using zPDT command ALCKKD.
2. Add the new file to the awsckd Manager Stanza within the zPDT devmap file, typically the devmap file at /home/ibmsys1/zdt/volumes/aprof1. To determine the location of the active devmap, use the awsstat command.

```
ibmsys1@adcdtst1 ~]$ awsstat
config file: /home/ibmsys1/zdt/volumes/aprof1, 3270port: 3270, Instance: ibmsys1
DvNbr S/Ch --Mgr-- IO Count --PID-- -----Device Information-----
0300 9 AWSCKD 11 1909 /home/ibmsys1/zdt/volumes/ZDTPY
0301 10 AWSCKD 1 1910
0302 11 AWSCKD 1 1911
0303 12 AWSCKD 1 1912
0304 13 AWSCKD 1 1915
0305 14 AWSCKD 1 1916
```

3. If extra space is required for Db2 archive logs, add extra SMS volumes to the SGARCH or SGBASE Storage Groups.

4. dynamically add the new disk to the running system by using the command `awsmount` with the `-m` option when you have spare or unassigned disk statements that are coded for the disk manager stanza in your devmap file.
5. If the volume cannot be added dynamically, shutdown z/OS, and stop the emulator by using the command `awsstop`. Then, enter `awsstart` to restart the emulator with the updated devmap file.
6. Use `ICKDSF` to initialize the new volume. Make sure that the `STGR` parameter is used on the initialization to ensure that the volume is marked as SMS-managed. Make sure that the used volume name is already listed under the Storage Group Volume assignments, then you do not need to update the SMS configuration.
7. Bring the volume online to z/OS by using the command `Vary xxxx,online`.
8. Enter `D SMS,VOL(XXXX)` to verify that the volume is online to an SMS Storage Group.

Portability of Application Data

When you deploy a ZD&T system with ADCD and decide to port a specific application to the ZD&T system, you must create and allocate `APPLxx` disk volumes to use the `SCAPPL` and `SGAPPL` SMS constructs. These volumes can contain all the application-related data in a specific known location. Then, you can copy the entire volumes to somewhere else for further testing or other needs. Although some other Storage Group constructs can potentially be used to achieve the same purpose, such as `SGCICS`, `SGIMS`, or `SGDB2`, the existing ACS routines will not allow the data to be forced into these groups.

Overflow strategy

The ACS routines assign `SGBASE` as the secondary group for all other Storage Groups. Any disk volume that belongs to the `SGBASE` Storage Group might potentially be the target for the new allocations of SMS-managed data, even if a different Storage Group that contains a different volume was the primary target. For example, if the primary Storage Group is full or offline to the system, SMS will route the allocation of the Storage Group and all associate disk volumes in the Storage Group to the secondary storage. Then, the allocation failure of this primary Storage Group will be avoided.

Simplicity

The existing ACS routines are simplistic, and the new SMS configuration intends to provide the following advantages.

- Provide an environment that helps prevent allocation failures.
- Ensure that the user data is not placed inappropriately, and new users can easily use SMS features.
- Provide a foundation that is easy to modify for more experienced users.
- Provide many constructs for future use. Or, provide these constructs as a base or example that you can learn or build upon your own needs.

NetView automation for ZD&T

This section documents some system automations that are available to start or stop the system and certain z/OS subsystems.

z/OS system startup and shutdown are handled by NetView®. There are two tasks (`CNMPSSI` and `CNMPROC`) that are started automatically at the beginning of the IPL process. When NetView has initialized, it will start the remaining z/OS subsystem tasks.

NetView Commands

To communicate with NetView, you can use one of the following methods.

- Modify z/OS command. For example, `F CNMPROC,xxxxxxx`.
- Use the subsystem command character at the z/OS console. For example, `%NETV xxxxxx`.

Valid NetView Commands are displayed as follows.

Table 39. Valid NetView Commands

Commands	Description
SHUTSYS	Shut down all z/OS system tasks.
GETSTAT	Show the status of all z/OS system tasks that are managed by NetView.
STRTTASK <i>task name</i>	Start a managed task.
STOPTASK <i>task name</i>	Stop a managed task.
SETSTAT <i>task name</i>	Set a specific status for a managed task.
DSPSTAT	Display and update the status for all managed tasks.
GETGLBVA <i>task name</i>	Display the task global variables, for example, start or stop command.
GETHELP	Display the list of valid commands.

NetView managed task status settings

The output of the GETSTAT command might show any of the following status settings for a managed task.

- **NOSTART**

The status of NOSTART indicates that the task is not automatically started after an IPL. Typically, this status occurs because the software was not selected when the z/OS image is built from ZD&T Enterprise Edition. In some cases, when software is not selected during image build, the disk volumes are not present to support starting these tasks. If there are no disk volume dependencies for the task that has a status of NOSTART, you can start the task by using the STRTTASK command that is described in the preceding table for NetView Commands.

- **COMPLETE**

The status of COMPLETE indicates that the task is transient, or multiple tasks have the same start or stop task name, or the task does not remain active in the z/OS system after the task is started. The tasks that have a status of COMPLETE are considered as UP after the tasks are started, and the continuous status of the tasks will not be tracked. Therefore, the tasks that have a status of COMPLETE might be down, and manual verification might be required if some errors occur with these tasks.

- **UP**

Task is up.

- **DOWN**

Task is down.

- **IPL**

All z/OS tasks are set with an initial status of IPL when the system is restarted. If a task still has the status of IPL after an IPL is complete, some errors might occur with the automation and manual actions might be required to start the task.

To list all tasks that are managed by NetView run the GETSTAT command on the z/OS system, the startup messages are displayed as follows.

During the system IPL process, the following examples of messages might be displayed.

```
INFO: 011620 17:44:10:      : OPRMSG:   BPXF002I FILE SYSTEM DFH550.JVMPROFS.ZFS WAS
INFO: 011620 17:44:10:      : OPRMSG:   NOT MOUNTED. RETURN CODE = 00000080, REASON CODE =
EF096055
```

or

```
INFO: 011520 17:05:33:      : OPRMSG:  *   IGGN505A SPECIFY UNIT FOR DFH550.CICS.SDFHLINK ON
A4C551 OR CANCEL
```

The first example of message indicates that the volume where the ZFS file is allocated was not included in the ZD&T z/OS image. The system will attempt to mount all ADCD ZFS files regardless of what is contained in the ZD&T image. This process can provide more flexibility by allowing ZFS mounts to succeed after dynamically adding volumes to the running system.

The second example relates to CICS Version 5.5. Since CICS V5.5 was not included in the ZD&T z/OS image, some disk volume files are intentionally excluded. Therefore, the data sets on the volume cannot be added to the system LINKLIST. The IGGN505A message will be automatically responded by ZD&T Enterprise Edition to bypass this condition and the message can be safely ignored.

Manual IPL Information

If you want to IPL the z/OS system manually by using the zPDT IPL command, you need to use the following IPL string.

```
ipl 0a80 parm 0a82nv
```

This IPL string allows z/OS system parmlib automation to handle some early IPL responses that might be required. These responses can normally be made by ZD&T Enterprise Edition. However, when the manual IPL process of z/OS is done, ZD&T Enterprise Edition will not be aware of any outstanding responses that might be needed.

Setting up Rational Tokens

Rational Token licensing is an entitlement that you can purchase and use to run ZD&T. With Rational Tokens, ZD&T 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 ZD&T. 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 ZD&T. 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 RDTCONF to point to a directory that contains the Rational configuration files. For example:

```
export RDTCONF=/etc/yourconffiles/
```

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 ZD&T offering.

When you are using Rational Tokens, an instance of ZD&T 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 ZD&T 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 ZD&T 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 ZD&T 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 ZD&T 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 `RDTLOG=TTY`, if set before you start ZD&T, routes all Rational Token-related logging to the Linux console in addition to the log.

Variable `RDTDEBUG=DEBUG`, if set before you start ZD&T, adds more information to the logs. If `RDTLOG=TTY` is also set, these additional messages are also written to the Linux console.

Do not set `RDTLOG=TTY` in everyday use because it sends frequent unsolicited messages to your Linux console. Setting `RDTDEBUG=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 ZD&T renews them before that time. ZD&T renews tokens approximately every half hour.

When ZD&T 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 ZD&T systems.

Note: For definitions of some of the terms used in these topics, see the [“Glossary”](#) on page 202.

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 ZD&T 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 ZD&T systems. [Figure 13 on page 187](#) 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 *Software-based License Server* and *license server* are used interchangeably. zPDT refers to both the ISV zPDT product (1090) and the ZD&T (1091) product except where distinctions are noted.

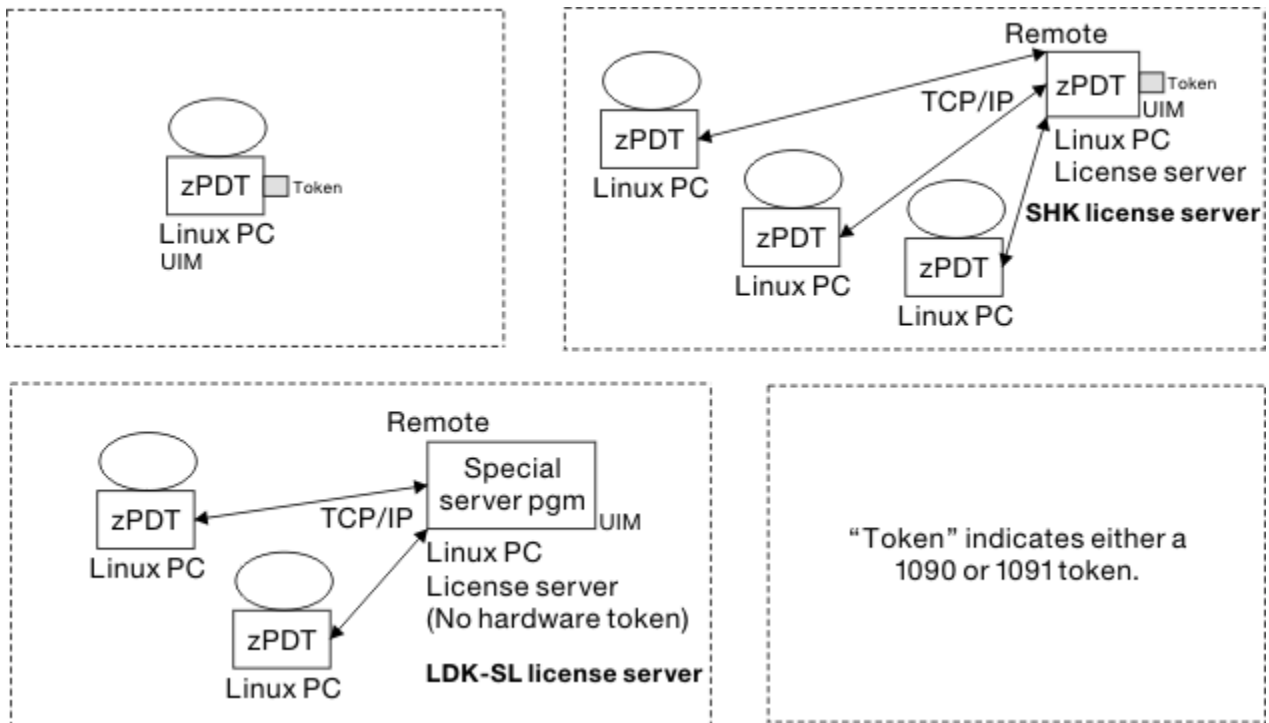


Figure 13. Options for obtaining zPDT licenses

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 Software-based License Server 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 13 on page 187 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 14 on page 189 illustrates the conceptual operation.

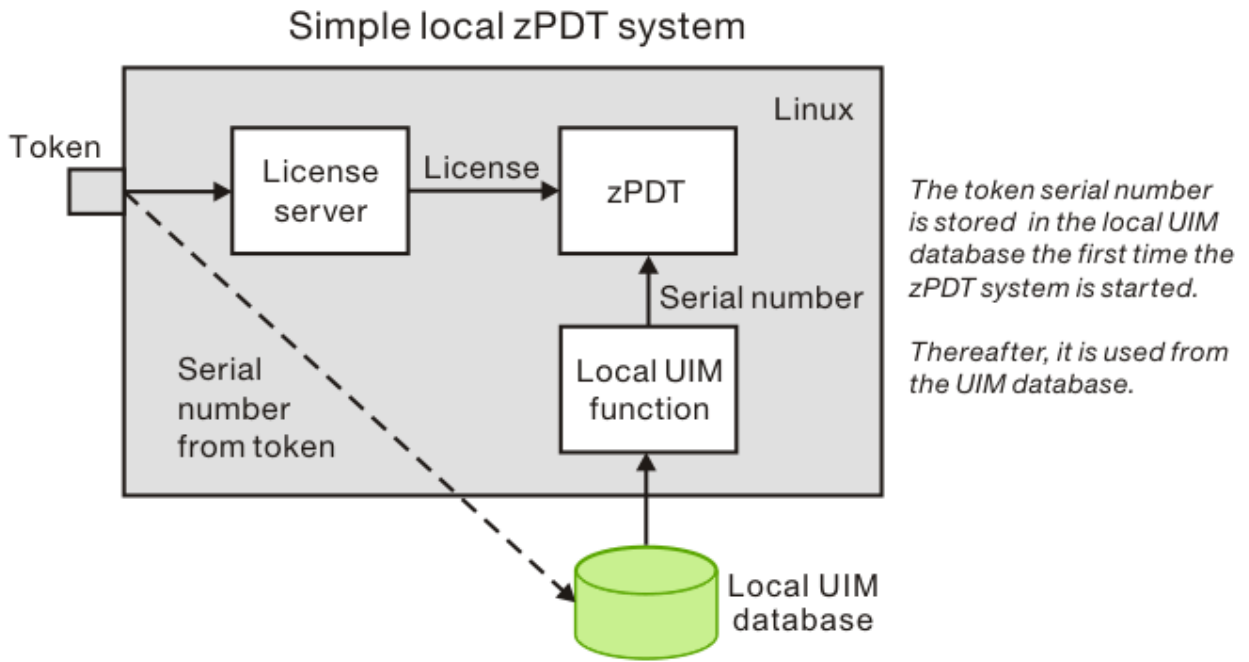


Figure 14. Simple local operation

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 15 on page 189](#) illustrates the general concept.

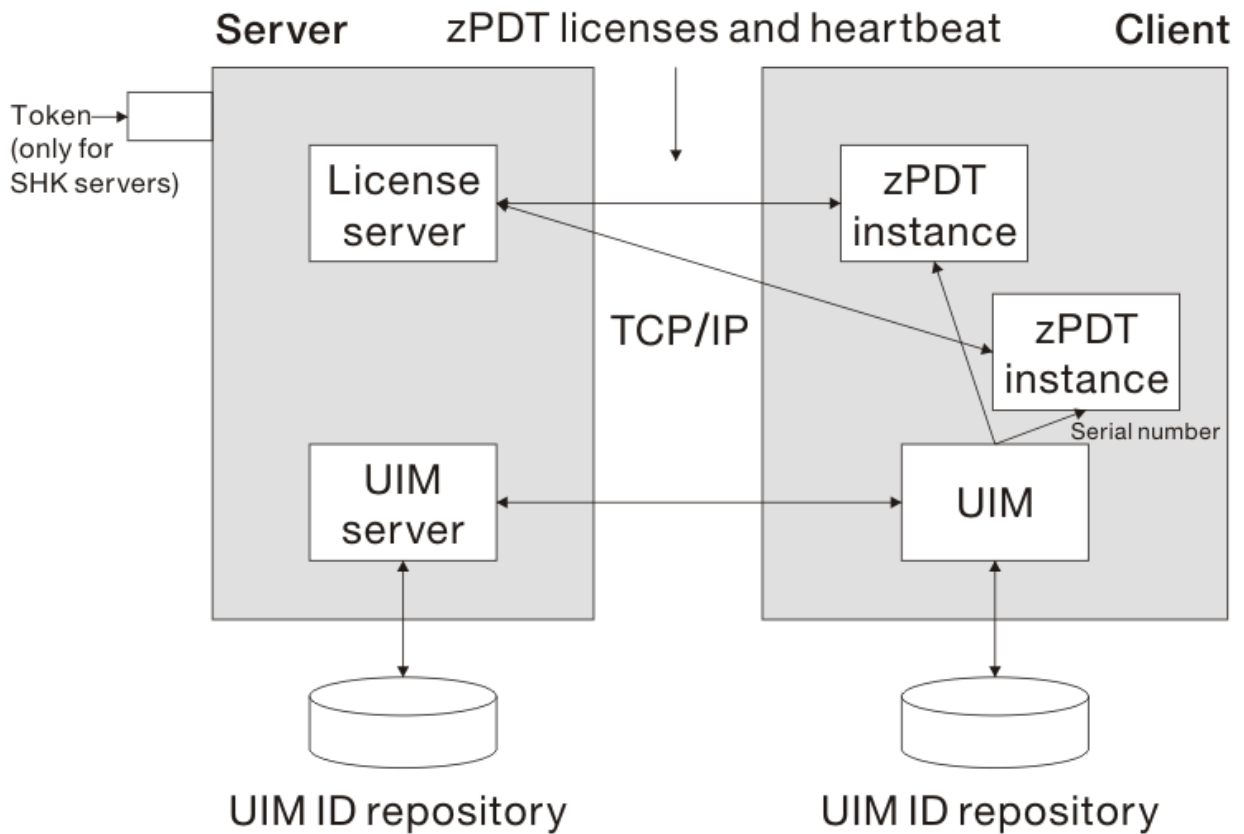


Figure 15. License and UIM servers

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 14 on page 189](#), 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 15 on page 189](#):

- 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 199](#).

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, overtype 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 191](#) and [“LDK client configuration” on page 192](#)

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 ZD&T 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 `sntlconfigsrvr.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 192. 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 Software-based License Server 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 udp --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 191 or “[LDK client configuration](#)” on page 192.
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 Software-based License Server 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	E6D96D01-493E-11CB-AD29-B8F42F7F8461	2016	009

Security

If the Software-based License Servers 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 udp --dport 1947 -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 ZD&T 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 200](#).

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.

Glossary

Find terms that are used in the ZD&T IBM Knowledge Center.

activation

See [“USB hardware device activation” on page 206](#).

Application component

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

Application image

Collection of application components that originate from the same IBM Z, ZD&T system, or ADCD package. The application components can be packaged together as a single deliverable for a target system.

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 ZD&T 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 ZD&T 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.

Software-based License Server

The remote ZD&T that distributes authentication by using software-based license keys.

See also [“license server” on page 204](#).

Software-based License Server update file

The specific license key file that is generated in the Rational License Key Center and installed on a Software-based License Server 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 Software-based License Server.

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 204.

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 Software-based License Server, and a web interface to the Software-

based License Server. 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 204.

Target environment

One (or more) x86 Intel (or compatible) systems capable of running ZD&T that users can deploy an application image to.

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” on page 205.

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 ZD&T 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 ZD&T 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

A

access, product license server [158](#)
activate a USB hardware device for v9.1 [19](#), [58](#), [152](#)
activation [56](#), [151](#)
activation, license [56](#), [151](#)
activation, Rational Token [48](#), [143](#)
activation, replacement update file [20](#), [59](#), [153](#)
activation, troubleshooting USB hardware device [178](#)
activation, USB hardware device [56](#), [151](#)
applying the update file [49](#), [144](#)

C

connections, troubleshooting Rational License Key Servers [177](#), [186](#)

D

device activation [19](#), [58](#), [152](#)
device activation, troubleshooting [178](#)
device map, defining [20](#), [159](#)

E

enabling product operation [44](#), [139](#)

F

file, update [57](#), [151](#)

H

hardware device activation [19](#), [58](#), [152](#)
hardware device activation, troubleshooting [178](#)

I

initial Rational Token license [57](#), [151](#)
initial Rational Token license, Steps to get [57](#), [151](#)
initial update file, Steps to get [57](#), [151](#)
installing Software License Server [46](#)
installing Software-based License Server [141](#)

K

key [18](#), [54](#)

L

license file replacement [20](#), [59](#), [153](#)
license key [18](#), [54](#)

O

obtain an update file [18](#), [47](#), [48](#), [56](#), [143](#), [151](#)
Obtain an update file [56](#), [151](#)
Obtaining update files for Rational Tokens [57](#), [151](#)

P

pointing to a Rational License Key Server [185](#)
prerequisites [11](#), [37](#), [134](#)
product license server access, set up [158](#)
product operation
 enabling [44](#), [139](#)

R

Rational License Key Center [18](#), [47](#), [48](#), [56](#), [143](#), [151](#)
Rational License Key Server, pointing to [185](#)
Rational License Key Servers, troubleshooting connections [177](#), [186](#)
Rational Token activation [48](#), [143](#)
Rational Token license, initial [57](#), [151](#)
Rational Tokens [185](#)
Rational Tokens, update file [57](#), [151](#)
replacement license file [20](#), [59](#), [153](#)
replacement update file [20](#), [59](#), [153](#)

S

set up product license server access [158](#)
software license server [49](#)
Software License Server
 installing [46](#)
software-based license server [144](#)
Software-based License Server
 installing [141](#)
Steps to get the initial update file [57](#), [151](#)
Steps to get the Rational Token license [57](#), [151](#)

T

Tokens [185](#)
troubleshooting and support [166](#)
troubleshooting connections, Rational License Key Servers [177](#), [186](#)
troubleshooting USB hardware device activation [178](#)

U

update file [18](#), [47](#), [48](#), [56](#), [143](#), [151](#)
update file, initial [57](#), [151](#)
update file, obtain [56](#), [151](#)
update file, replacement [20](#), [59](#), [153](#)
update file. Rational Tokens [57](#), [151](#)
USB hardware device [18](#), [54](#), [56](#), [151](#)
USB hardware device activation [19](#), [58](#), [152](#)

USB hardware device activation, troubleshooting [178](#)
USB hardware device and license activation [56](#), [151](#)
USB hardware device licensing [18](#), [54](#)

V

V9.1 USB hardware device activation [19](#), [58](#), [152](#)

