

*IBM Z Development and Test
Environment v13.3.0*



Contents

IBM Z Development and Test Environment.....	1
Overview.....	1
What's new in ZD&T 13.3.0.....	3
Learning resources.....	4
Notices.....	5
Trademarks.....	6
Terms and conditions for product documentation.....	6
Learners Edition.....	7
Checklist.....	9
Prerequisites.....	11
Installation packages.....	12
Validating the downloaded installer file.....	14
Installing Learners Edition.....	15
Enabling a license key.....	17
Applying the update file to the ZD&T for Learners Edition machine.....	17
Configuring Z instances.....	18
Setting up the ADCD package	18
Creating the device map.....	19
Starting Z instances.....	19
Stopping Z instances	21
Configuring the network.....	21
Updating Learners Edition.....	25
Uninstalling the earlier versions of ZD&T	25
Updating Learners Edition to the latest version.....	27
Uninstalling Learners Edition.....	28
Personal Edition.....	28
Checklist.....	30
Prerequisites.....	32
Installation packages.....	33
Validating the downloaded installer file.....	36
Installing Personal Edition.....	36
Enabling a license key.....	38
Obtaining an update file from Rational License Key Center.....	39
Applying the update file to the ZD&T Personal Edition machine.....	39
Returning an existing license key.....	40
Configuring Z instances.....	41
Setting up the ADCD package	41
Creating the device map.....	42
Starting Z instances.....	42
Stopping Z instances	43
Configuring the network.....	44
Updating Personal Edition.....	47
Uninstalling the earlier versions of ZD&T	47
Updating Personal Edition to the latest version.....	49
Uninstalling Personal Edition.....	50
Enterprise Edition.....	51

Checklist.....	55
Prerequisites.....	58
Installing Sandbox in a cluster with internet access.....	66
Installing Sandbox in a cluster with a restricted network.....	67
Installing the Sandbox Operator.....	69
Verifying the installation of the Sandbox Operator.....	70
Installing and using the IBM License Service.....	71
Configuring the cluster.....	71
Configuration reference.....	75
Installation packages.....	87
Validating the downloaded installer file.....	90
Enabling the Software-based License Server.....	90
Installing Software-based License Server.....	91
Obtaining an update file.....	92
Applying the update file to the Software-based License Server machine.....	94
Combining all the available licenses on a single license server.....	95
Returning a software-based license key.....	96
Updating the Software-based License Server.....	98
Uninstalling the Software-based License Server.....	99
Installing Enterprise Edition.....	100
Installing the Enterprise Edition web server.....	100
Installing a signed certificate on the Enterprise Edition web server.....	101
Optional: Starting the Enterprise Edition web server.....	102
Configuring the Enterprise Edition web server.....	103
Installing the ZD&T z/OS Extraction Utilities.....	104
Optional(Legacy): Installing ZD&T emulator.....	105
Updating the Enterprise Edition web server.....	106
Uninstalling the Enterprise Edition web server.....	106
User's Guide.....	107
Overview of Enterprise Edition.....	107
Firewall ports list.....	110
Limitations.....	112
Web UI User's Guide.....	114
REST APIs User's Guide.....	149
Troubleshooting.....	188
Setting up a Windows machine as an FTP server.....	188
Troubleshooting the configuration of the Application Discovery server.....	189
Troubleshooting the components creation from IBM Application Discovery projects.....	189
Scenarios.....	190
Provisioning CICS subsystem from a source Z.....	190

Parallel Sysplex..... 190

Checklist.....	194
Prerequisites.....	197
Installation packages.....	199
Validating the downloaded installer file.....	202
Enabling the license servers.....	202
Software-based licensing.....	204
Hardware-based licensing.....	211
Installing Parallel Sysplex.....	218
Activating ZD&T Parallel Sysplex instances.....	219
Software-based license.....	219
Hardware-based license.....	221
Configuring Z instances.....	221
Setting up the ADCD and z/VM package.....	221
Creating the device map.....	222
Starting z/VM.....	223

Starting Z instances.....	224
Stopping Z instances and z/VM.....	225
Updating Parallel Sysplex.....	225
Uninstalling the earlier versions of ZD&T	226
Updating the installed Parallel Sysplex.....	227
Uninstalling Parallel Sysplex.....	228
z/OS 2.4 ADCD reference.....	228
IBM Extended z/OS ADCD for ZD&T.....	228
Modified ADCD z/OS V2R4 May Edition of 2021.....	230
Hints and tips to manage the z/OS 2.4 ADCD.....	230
Customizing ADCD for ZD&T.....	231
Automating NetView for ZD&T.....	236
Adding a task to NetView Automation.....	238
Configuring TCP/IP network	239
Improving Java application performance.....	240
Troubleshooting and support.....	242
Finding known problems.....	242
IBM Software Support.....	242
Obtaining the latest editions of Redbooks.....	242
Troubleshooting tips.....	242
Installing ZD&T.....	243
USB Hardware device activation and licensing.....	244
Installing, operating, and uninstalling the Software-based License Server.....	248
Installing z/OS volume images.....	248
Defining the ZD&T machine characteristics (DEVMAP).....	249
Starting and stopping ZD&T.....	249
z/OS networking issues.....	249
Performance	251
Connecting with Rational License Key Servers.....	252
Troubleshooting Software-based License Server and USB hardware device activation	253
Reference.....	253
Setting up Rational Tokens.....	253
Installing, configuring, and administering a Rational License Key Server.....	253
Updating the device map to point to a Rational License Key Server.....	254
Troubleshooting connections with Rational License Key Servers.....	254
zPDT license servers.....	255
Concepts.....	256
Client installation and configuration	259
Server installation and configuration.....	261
Notes.....	263
Scenarios.....	265
Renewing licenses.....	269
Searching servers.....	270
Allocating multiple licenses.....	270
Glossary.....	271
Index.....	276

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](#)

[Personal Edition](#)

[Enterprise Edition](#)

[Parallel Sysplex](#)

Learn more

[ZD&T product page](#)

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

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

Connect with experts

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

[Forum for ZD&T on IBM Community](#)

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

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](#)

[zPDT Sysplex Extensions - 2020](#)

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 four different editions:

ZD&T for Learners Edition

ZD&T for Learners Edition is for personal, individual learning use only. It is not for the test and development of commercial products. It enables a qualified learner to run an IBM Z distribution on a personal computer. For more information about ZD&T for Learners Edition, see [Learners Edition](#).

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](#).

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 create, and manage images from an existing Z or ADCD packages. Then, you can provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to your target environment. Also, you are entitled to use Wazi Sandbox for cloud native development by provisioning and maintaining z/OS instances on Red Hat® OpenShift®. For more information about ZD&T Enterprise Edition, see [Enterprise Edition](#).

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](#).

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

Feature	Learners Edition	Personal Edition	Enterprise Edition	Parallel Sysplex
License type	USB	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	Yes
Multiple users	No	No	Yes	Yes
Scalable ²	No	No	Yes	Yes
Cloud/VM support ³	No	No	Yes	Yes
REST API	No	No	Yes	No
Web UI	No	No	Yes	No
Native Linux interface	Yes	Yes	Yes for advanced Linux users	Yes
Dataset transfer support	No	No	Yes	No
Volume transfer support	No	No	Yes	No
Integration with IBM AD	No	No	Yes	No
OpenStack Cloud support	No	No	Yes	No
Docker container support	No	No	Yes	Yes
Red Hat OpenShift support	No	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 13.3.0

The following new features and changes are provided in ZD&T 13.3.0.

ZD&T Personal Edition

A new signed installer file `ZDT_Install_PE_V13.3.0.tgz` is available for Personal Edition. You can validate the completeness of the downloaded package by running a command with the signing certificate that is packaged in the installer file.

For more information, see [“Validating the downloaded installer file” on page 36](#) and [“Installing Personal Edition” on page 36](#).

ZD&T Enterprise Edition

• A new installer

A new signed installer file `zdt-install.tgz` is available for Enterprise Edition. You can validate the completeness of the downloaded package by running a command with the signing certificates that are packaged in the installer file. For more information, see [“Validating the downloaded installer file” on page 90](#).

The new installer file contains the following multiple installers:

- `zdt-install` that installs ZD&T Enterprise Edition web server. For more information, see [Installing the Enterprise Edition web server](#).
- `zdt-license-server` that installs the Software-based License Server. For more information, see [“Installing Software-based License Server” on page 91](#).
- `zdt-emulator` that installs ZD&T emulator. For more information, see [“Optional\(Legacy\): Installing ZD&T emulator” on page 105](#).

• New REST APIs for Linux target environments

For new-boarding customers, you can create a Linux target environment by using a new REST API. In this case, if succeeded, the response contains a UUID that represents the Linux target environment. With this UUID, you can manage the Linux target environment or provision instances to this target environment by using other new REST APIs. However, limitations exist. Migration is not supported yet. Also, comparing with the REST APIs that are provided in earlier versions, some features are not supported, for example, updating a provisioning.

For more information, see [“Creating and managing Linux target environments” on page 150](#) and [“Provisioning instances to a Linux target environment” on page 165](#).

• A new method of editing image properties in the web server

You can not only edit image properties when modifying selected components for a created image, but also choose to edit image properties directly on the **Images** page.

For more information, see section "Editing image properties" in topic [Monitoring and managing all created images](#).

- **Sandbox Operator**

- You do not need high privileges to run the Sandbox container because it no longer runs as root.
- You can better control how to expose z/OS ports in OpenShift by using Routes, ClusterIP Services, and NodePort Services.

ZD&T Parallel Sysplex

A new signed installer file `ZDT_Install_PS_V13.3.0.tgz` is available for Parallel Sysplex. You can validate the completeness of the downloaded package by running a command with the signing certificate that is packaged in the installer file.

For more information, see [“Validating the downloaded installer file”](#) on page 202 and [“Installing Parallel Sysplex”](#) on page 218.

IBM Extended z/OS ADCD for ZD&T

The following components are updated in IBM Extended z/OS ADCD for ZD&T 13.3.

- Debug for z/OS 14.2.2, 15.0.3
- Dependency Based Build 1.1.2
- Explorer Extensions 15.0.3
- Explorer for z/OS 3.1.1.26, 3.2.0.17
- RSE API 1.0.7
- Z Open Automation Utilities 1.1.1

For the complete list, see [“IBM Extended z/OS ADCD for ZD&T”](#) on page 228.

zPDT

The zPDT[®] driver that is used in all ZD&T 13.3.0 editions is updated to zPDT 1.10.55.12. The new emulator includes the following fixes.

- Enhancement to allow defining z memory in the devmap with Gb.
- Fixed an unexpected Specification Exception condition for Compression Call instruction (CMPSC).
- Fixed z1090instcheck on Ubuntu 20.04.
- Enhanced Software License Server configuration process to support more rules and enhanced error message.

Related information

[What's new in ZD&T V13.2.0](#)

[What's new in ZD&T V13.1.0](#)

[What's new in ZD&T V13.0.0](#)

[What's new in ZD&T V12.0.5](#)

[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 Trial program is powered by ZD&T](#)
- [developerWorks®: IBM Z Development and Testing Hub](#)
- [developerWorks forum for ZD&T](#)
- [zPDT Guide and Reference](#)
- [zPDT Sysplex Extensions - 2020](#)
- [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.

Learners Edition

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

ZD&T for Learners Edition can create an environment to learn about mainframe technologies 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 for Learners Edition, you need to run the Learners 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 Learners Edition.



Figure 1. The topology and various components of Learners Edition

ZD&T for Learners Edition provides the following features:

- Current® levels of IBM z/OS software that can offer additional learning experiences with new middleware and runtime capabilities
- An approachable and portable environment for education on Z for a single user

License Key (USB)

ZD&T for Learners Edition comes with a USB hardware device that contains the fingerprint that is used to identify the entitlement of user. After you install Learners 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 17.

Application Developers Controlled Distribution (ADCD)

ZD&T for Learners Edition comes with the Application Developers Controlled Distribution (ADCD) package, which is known as z/OS ADCD. A z/OS ADCD for ZD&T for Learners Edition contains many of the products and subsystems that you might need to try z/OS coding 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 [Products contained in this release](#).

3270 terminal emulator

After Learners 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 for Learners 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 for Learners Edition, see [“Checklist”](#) on page 9.

Checklist

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

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

Installing Learners 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 Learners Edition.	Required	<input type="checkbox"/>
3	“Installing Learners Edition” on page 15 Describes the steps to run the installation package.	Required	<input type="checkbox"/>
4	“Enabling a license key” on page 17 Describes the steps to obtain a license key, and apply the key to the machine.	Required	<input type="checkbox"/>
5	Configuring Z instances 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 Describes the steps to start Learners Edition and Z instances.	Required	<input type="checkbox"/>
7	Stopping Z instances Describes the steps to stop Learners Edition and z/OS images.	Optional	<input type="checkbox"/>
8	Configuring the network Describes various ways to configure the network.	Optional	<input type="checkbox"/>

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

Updating Learners Edition		Required/ Optional	Complete
1	“Prerequisites” on page 11 Describes the software, hardware requirements, and the steps to set up the basic Linux system for newer version of ZD&T.	Required	<input type="checkbox"/>

Table 3. Update checklist (continued)

Updating Learners Edition		Required/ Optional	Complete
2	<p><u>Obtaining installation packages</u></p> <p>Describes the steps to download all required packages to install Learners Edition.</p>	Required	<input type="checkbox"/>
3	<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"> – <u>“Updating Learners Edition to the latest version” on page 27</u> • If you installed zD&T V10.0.1 or earlier versions, complete the following steps: <ul style="list-style-type: none"> – <u>“Uninstalling the earlier versions of ZD&T ” on page 25</u> – <u>“Installing Learners Edition” on page 15</u> <p>Describes the steps to update Learners Edition.</p>	Required	<input type="checkbox"/>
4	<p><u>“Enabling a license key” on page 17</u></p> <p>Describes the steps to obtain a license key, and apply the key to the machine.</p>	Required	<input type="checkbox"/>
5	<p><u>Configuring Z instances</u></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><u>“Enabling a license key” on page 17</u></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 for Learners Edition, complete the following steps:

Table 5. Uninstallation checklist

Uninstalling Learners Edition		Required/ Optional	Complete
1	<p><u>“Uninstalling Learners Edition” on page 28</u></p> <p>Describes the steps to uninstall Personal Edition.</p>	Required	<input type="checkbox"/>

Prerequisites

Before you install or update ZD&T for Learners 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](#).

To download and extract files, the following hardware is also required.

- 4.1 GB is required to download files.
- There are 33 extracted files. 15 GB is needed for one of the files, and 8 GB is needed for each of others.
- 300 GB is needed to hold the ACDC files.

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 Learners Edition”](#) on page 15 and [Configuring the network](#).

Installation packages

An email with a location to download the required installation packages is sent to you.

After you download the required installation packages, you can verify the integrity of downloaded ADCD packages by using the MD5SUM that is in the adcd.md5 and pe.md5 files. To do a checksum on the downloads, use command `md5sum -c dec2020_adcd_md5.txt`.

	Name	Part No.	Package Type	File Name
1	IBM Z Development and Test Environment for Learners Edition 13.2.0 Installation Multilingual	G01W5ML	ZD&T for Learners Edition Installer	ZDT_Install_LE_V13.2.0.tgz
2	IBM Z Development and Test Environment for Learners Edition 13.2.0 Installer checksum Multilingual	G01W6ML	checksum (md5 checksum)	pe.md5
3	ADCD z/OS V2.4 for IBM Z Development and Test Environment 13.2 RSU 2009 Multilingual eAssembly	G012SML	N/A	N/A
4	z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	G01V0ML	ADCD December 2020 Edition RSU level 2009	C4RES1.ZPD

	Name	Part No.	Package Type	File Name
5	z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	G01V1ML	ADCD December 2020 Edition RSU level 2009	C4RES2.gz
6	z/OS 2.4 Part 3 of 19 - System volume Multilingual	G01V2ML	ADCD December 2020 Edition RSU level 2009	C4SYS1.gz
7	z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	G01V3ML	ADCD December 2020 Edition RSU level 2009	C4CFG1.gz
8	z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	G01V4ML	ADCD December 2020 Edition RSU level 2009	C4USS1.gz
9	z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	G01V5ML	ADCD December 2020 Edition RSU level 2009	C4USS2.gz
10	z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	G01V6ML	ADCD December 2020 Edition RSU level 2009	C4PRD1.gz
11	z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	G01V7ML	ADCD December 2020 Edition RSU level 2009	C4PRD2.gz
12	z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	G01V8ML	ADCD December 2020 Edition RSU level 2009	C4PRD3.gz
13	z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	G01V9ML	ADCD December 2020 Edition RSU level 2009	C4PRC4.gz
14	z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	G01VBML	ADCD December 2020 Edition RSU level 2009	C4DIS1.gz
15	z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	G01VCML	ADCD December 2020 Edition RSU level 2009	C4DIS2.gz
16	z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	G01VDML	ADCD December 2020 Edition RSU level 2009	C4DIS3.gz
17	z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	G01VFML	ADCD December 2020 Edition RSU level 2009	C4PAGA.gz
18	z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	G01VGML	ADCD December 2020 Edition RSU level 2009	C4PAGB.gz
19	z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	G01VHML	ADCD December 2020 Edition RSU level 2009	C4PAGC.gz
20	z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	G01VJML	ADCD December 2020 Edition RSU level 2009	C4USR1.gz
21	z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	G01VKML	ADCD December 2020 Edition RSU level 2009	SARES1.ZPD
22	z/OS 2.4 Part 19 of 19 – zCX Container volume	G01VLML	ADCD December 2020 Edition RSU level 2009	C4ZCX1.gz

	Name	Part No.	Package Type	File Name
23	Engineering Workflow Management 7.0.1 Multilingual	G01VMML	ADCD December 2020 Edition RSU level 2009	C4BLZ1.gz
24	DB2® V12 Part 1 of 2 Multilingual	G01VNML	ADCD December 2020 Edition RSU level 2009	C4DBC1.gz
25	DB2 V12 Part 2 of 2 Multilingual	G01VPML	ADCD December 2020 Edition RSU level 2009	C4DBC2.gz
26	DB2 V11 Part 1 of 2 Multilingual	G01VQML	ADCD December 2020 Edition RSU level 2009	C4DBB1.gz
27	DB2 V11 Part 2 of 2 Multilingual	G01VRML	ADCD December 2020 Edition RSU level 2009	C4DBB2.gz
28	CICS 5.6 Multilingual	G01VSML	ADCD December 2020 Edition RSU level 2009	C4C560.gz
29	CICS 5.5 Multilingual	G01VTML	ADCD December 2020 Edition RSU level 2009	C4C551.gz
30	IMS 14.1 Multilingual	G01VVML	ADCD December 2020 Edition RSU level 2009	C4IME1.gz
31	IMS 15.2 Multilingual	G01VWML	ADCD December 2020 Edition RSU level 2009	C4IMF1.gz
32	IBM Installation Manager 1.9 Multilingual	G01VXML	ADCD December 2020 Edition RSU level 2009	C4INM1.gz
33	DB2 Archive Logs Multilingual	G01VYML	ADCD December 2020 Edition RSU level 2009	C4DBAR.gz
34	Tivoli® Omegamon XE products 5.5.1 Multilingual	G01VZML	ADCD December 2020 Edition RSU level 2009	C4KAN1.gz
35	WAS V9.0 Part 1 of 2 Multilingual	G01W0ML	ADCD December 2020 Edition RSU level 2009	C4W901.gz
36	WAS V9.0 Part 2 of 2 Multilingual	G01W1ML	ADCD December 2020 Edition RSU level 2009	C4W902.gz
37	Zowe™ Open Source Project Multilingual	G01W2ML	ADCD December 2020 Edition RSU level 2009	C4ZWE1.gz
38	Extended ADCD Checksum Multilingual	G01W4ML	ADCD package MD5 checksum	dec2020_adcd_md5.txt

Validating the downloaded installer file

ZD&T for Learners Edition installer now supports the signing of the installer. With the signed installer, you can validate the completeness of the downloaded package by running a command with the signing certificate that is packaged in the installer.

ZD&T for Learners Edition installer file ZDT_Install_LE_V13.2.0.tgz that you download from IBM is a compressed file. The installer file contains the installer ZDT_Install_LE_V13.2.0.x86_64 and the corresponding signature file ZDT_Install_LE_V13.2.0.x86_64.sig.

To validate the signature file for the installer, run the following command.

```
openssl dgst -sha256 -verify <PUBLICKEYNAME> -signature <signature_file> <FILETOSIGN>
```

For example, run the following command from the directory that contains the extracted installer files.

```
openssl dgst -sha256 -verify signature/public.pem -signature ZDT_Install_LE_V13.2.0.x86_64.sig  
ZDT_Install_LE_V13.2.0.x86_64
```

If the installation file is valid, you will receive the message of `Verified OK`.

Installing Learners Edition

Learn about the steps to install ZD&T for Learners Edition.

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

1. Open the directory that contains the installer `ZDT_Install_LE_V13.2.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_LE_V13.2.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_LE_V13.2.0.tgz
```

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

5. Run the installer.

```
./ZDT_Install_LE_V13.2.0.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](#).

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.

```
z1091-1-10.55.08.02.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
```

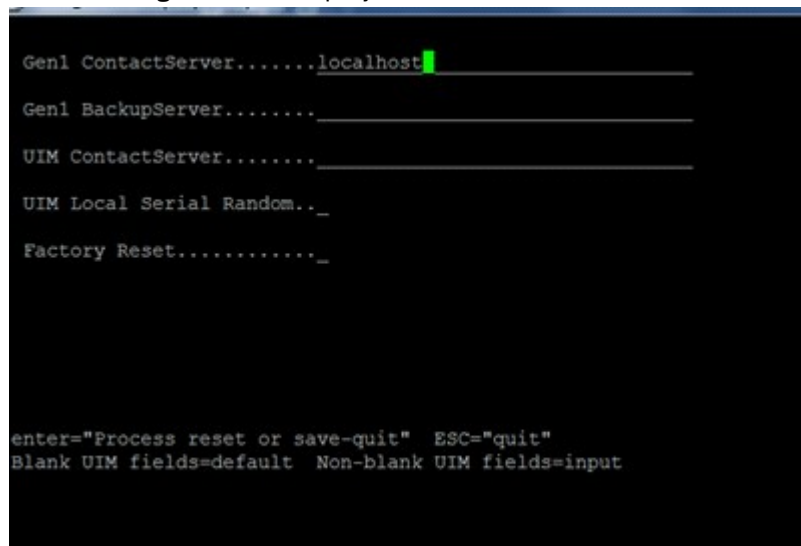
To delete the user ID, use the following command:

```
userdel -r ibmsys2
```

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

```
./clientconfig
```

The following screen is displayed.



```
Gen1 ContactServer.....localhost
Gen1 BackupServer.....
UIM ContactServer.....
UIM Local Serial Random.._
Factory Reset....._

enter="Process reset or save-quit"  ESC="quit"
Blank UIM fields=default  Non-blank UIM fields=input
```

Note: The installer changes the iptables and Firewall. If you choose to install Learners 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 Learners Edition with network configuration, see [Configuring the network](#) 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 Learners Edition without automated network configuration, run the following command:

```
./ZDT_Install_LE_V13.2.0.x86_64 --install
```

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

```
./ZDT_Install_LE_V13.2.0.x86_64 --install --net
```

- To install the dependencies, run the following command.

```
./ZDT_Install_LE_V13.2.0.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 Learners Edition interactively.

Enabling a license key

ZD&T for Learners 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 for Learners 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



Applying the update file to the ZD&T for Learners 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 for Learners Edition machine, complete the following steps:

1. Connect the USB hardware device to the computer that is running ZD&T for Learners 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
```

Configuring Z instances

You need to set up the ADCD package and create the device map to configure Z instances for educational 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](#).
3. Decompress all `.gz` volumes by using the Linux command.

```
gunzip *.gz
```

4. Set the environment variables by using the following command.

```
export PATH=/usr/z1090/bin:$PATH  
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH  
export MANPATH=/usr/z1090/man:$MANPATH  
ulimit -c unlimited  
ulimit -d unlimited
```

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

```
./Z1091_ADCD_install <path of volume>/C4RES1.ZPD <output path of volume>/C4RES1
```

For example:

```
./Z1091_ADCD_install  
/home/ibmsys1/volumes/C4RES1.ZPD  
/home/ibmsys1/volumes/C4RES1
```

6. 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.

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

```
chmod 755 *
```

For a complete list of z/OS products in ADCD, see [Products contained in this release](#).

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, ensure that you set the environment variables by using the following command.

```
export PATH=/usr/z1090/bin:$PATH  
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH  
export MANPATH=/usr/z1090/man:$MANPATH  
ulimit -c unlimited  
ulimit -d unlimited
```

Then, 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 for Learners 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`). Then, ensure the environment variables shown in the example below are defined before you run the script.

For example, you can copy the `./runzpdtd` script to the `/home/ibmsys1/volumes` directory, and run the script as follows.

```
export PATH=/usr/z1090/bin:$PATH
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH
export MANPATH=/usr/z1090/man:$MANPATH
ulimit -c unlimited
ulimit -d unlimited
./runzpdtd -d mygenerateddevmap -l cs
```

The script takes the following options:

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

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. Set the environment variables by using the following command.

```
export PATH=/usr/z1090/bin:$PATH
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH
export MANPATH=/usr/z1090/man:$MANPATH
ulimit -c unlimited
ulimit -d unlimited
```

4. Enter the following command to start ZD&T for Learners Edition.

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

5. 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.

6. Enter the IPL command:

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

For example, `ipl 0a80 parm 0a82CS`

7. 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 December Edition of 2020](#).

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.

```
opirmsg 'VARY CN(*),ACTIVATE'
opirmsg '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 for Learners 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 for Learners Edition. To shut down ZD&T for Learners Edition, use the respective shutdown command. All shutdown scripts are in the ADCD.Z24C.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 for Learners Edition.

Note: Do not make any change in the ADCD.Z24C.* library. If you want to edit any parm or proc, ADCD has USER.Z24C.* 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 for Learners Edition installer. For more information, see [Installing Learners Edition](#).

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 udp -- 0.0.0.0/0 0.0.0.0/0 udp 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 -- 0.0.0.0/0 172.26.1.2 tcp dpts:0:21
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:23:1946
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:1948:2021
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpt:2022
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:2023:3269
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:3271:5900
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:5903:9449
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 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 for Learners 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.
2. 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.

3. 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 --tunnel_ip=172.26.1.1 # 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
```

4. IPL the ZD&T z/OS with the CS loadparm option.
5. Copy GBLRESOL, TCPDATA, and PROF2 from ADCD.Z24C.TCPPARMS to USER.Z24C.TCPPARMS.
6. Copy FTPD, NFSC, NFSS, PORTMAP, RESOLVER, and TCPIP from ADCD.Z24C.PROCLIB to USER.Z24C.PROCLIB.
7. 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

```

8. Create the member IPNODES in USER.Z24C.TCPPARMS.

```

000001 X.XX.XXX.XXX ZOS-DEV4 ZOS-DEV4.ibm.com
000002 172.26.1.1      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.Z24C.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.Z24C.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

```



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

13. Verify whether the static IP works.

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

```
v net, id=osatr12, act
```

Updating Learners Edition

If you installed ZD&T V11.0 or later versions, you can update Learners 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 Learners Edition.

For the detailed steps to update Learners 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 43.– 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 1262 1187 1318">cd /usr/z1090/bin ./uimserverstop</pre>2. Enter these two commands to stop the product license server:<pre data-bbox="951 1457 1451 1528">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 Learners Edition"](#) on page 15.

Note: ZD&T for Learners Edition doesn't support older version of ZD&T. So, to use ZD&T for Learners 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 Learners Edition to the latest version

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

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

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

1. Open the directory that contains the installer `ZDT_Install_LE_V13.2.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_LE_V13.2.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_LE_V13.2.0.tgz
```

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

```
./ZDT_Install_LE_V13.2.0.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](#).

Important: Remember to download the volumes of ADCD May 2020 edition from passport advantage, and decompress the ***.gz** volume on the same machine where you install ZD&T V13.0.0.

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.

```
z1091-1-10.55.08.02.x86_64
```

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

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

```
./ZDT_Install_LE_V13.2.0.x86_64 --update
```

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

```
./ZDT_Install_LE_V13.2.0.x86_64 --update --net
```

After the update completes, you can follow the steps that are described in [“Enabling a license key”](#) on page 17 and [“Configuring Z instances”](#) on page 41.

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 Learners Edition

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

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

1. Stop any running Z instance. For detailed steps, see [Stopping Z instances](#).
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_LE_V13.2.0.x86_64
```

4. Type **3** to uninstall.

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

```
ZDT_Install_LE_V13.2.0.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 25.

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 32.

The following topology describes the various components of Personal Edition.



Figure 3. 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 38.

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 [Products contained in this release](#).

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 30](#).

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:

<i>Table 6. New installation checklist</i>			
Installing Personal Edition		Required/ Optional	Complete
1	“Prerequisites” on page 32 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 36 Describes the steps to run the installation package.	Required	<input type="checkbox"/>
4	“Enabling a license key” on page 38 Describes the steps to obtain a license key, and apply the key to the machine.	Required	<input type="checkbox"/>
5	Configuring Z instances 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 Describes the steps to start Personal Edition and Z instances.	Required	<input type="checkbox"/>
7	Stopping Z instances Describes the steps to stop Personal Edition and z/OS images.	Optional	<input type="checkbox"/>
8	Configuring the network Describes various ways to configure the network.	Optional	<input type="checkbox"/>

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

Table 7. Update checklist

Updating Personal Edition		Required/ Optional	Complete
1	<p>“Returning an existing license key” on page 40 Describes the steps to return an existing hardware-based license key.</p>	Required	<input type="checkbox"/>
2	<p>“Prerequisites” on page 32 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 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 49 • 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 47 – “Installing Personal Edition” on page 36 <p>Describes the steps to update Personal Edition.</p>	Required	<input type="checkbox"/>
5	<p>“Enabling a license key” on page 38 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 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 8. License update checklist

Updating the license		Required/ Optional	Complete
1	<p>“Returning an existing license key” on page 40 Describes the steps to return an existing hardware-based license key.</p>	Required	<input type="checkbox"/>
2	<p>“Enabling a license key” on page 38 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 9. Uninstallation checklist

Uninstalling Personal Edition		Required/ Optional	Complete
1	“Returning an existing license key” on page 40 Describes the steps to return an existing hardware-based license key.	Required	<input type="checkbox"/>
2	“Uninstalling Personal Edition” on page 50 Describes the steps to uninstall Personal Edition.	Required	<input type="checkbox"/>

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](#).

To download and extract files, the following hardware is also required.

- 4.1 GB is required to download files.
- There are 33 extracted files. 15 GB is needed for one of the files, and 8 GB is needed for each of others.
- 300 GB is needed to hold the ACDC files.

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 36](#) and [Configuring the network](#).

Installation packages

Check the version you are using and 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. To do a checksum on the downloads, use the command `md5sum -c may2021_adcd_md5.txt`.

	Name	Part No.	Image Classification	File Name
1	IBM Z Development and Test Environment Personal Edition 13.3.0 Installation Multilingual	M03WRM L	Required	ZDT_Install_P E_V13.3.0.tgz
2	IBM Z Development and Test Environment Personal Edition 13.3.0 Installer checksum Multilingual	M03WSM L	Required	pe.md5
3	ADCD z/OS V2.4 for IBM Z Development and Test Environment 13.3 RSU 2103 Multilingual eAssembly	G050QML	Optional, ADCD May 2021 Edition RSU level 2103	N/A
4	z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	M03WTM L	Optional, ADCD May 2021 Edition RSU level 2103	D4RES1.ZPD
5	z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	M03WVM L	Optional, ADCD May 2021 Edition RSU level 2103	D4RES2.gz
6	z/OS 2.4 Part 3 of 19 - System volume Multilingual	M03WWM L	Optional, ADCD May 2021 Edition RSU level 2103	D4SYS1.gz
7	z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	M03WXM L	Optional, ADCD May 2021 Edition RSU level 2103	D4CFG1.gz
8	z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	M03WYM L	Optional, ADCD May 2021 Edition RSU level 2103	D4USS1.gz
9	z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	M03WZM L	Optional, ADCD May 2021 Edition RSU level 2103	D4USS2.gz
10	z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	M03X0ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD1.gz
11	z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	M03X1ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD2.gz
12	z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	M03X2ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD3.gz
13	z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	M03X3ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRC4.gz
14	z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	M03X4ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS1.gz

	Name	Part No.	Image Classification	File Name
15	z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	M03X5ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS2.gz
16	z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	M03X6ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS3.gz
17	z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M03X7ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGA.gz
18	z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	M03X8ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGB.gz
19	z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	M03X9ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGC.gz
20	z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	M03XBML	Optional, ADCD May 2021 Edition RSU level 2103	D4USR1.gz
21	z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	M03XCML	Optional, ADCD May 2021 Edition RSU level 2103	SARES1.ZPD
22	z/OS 2.4 Part 19 of 19 – zCX Container volume	M03XDML	Optional, ADCD May 2021 Edition RSU level 2103	D4ZCX1.gz
23	Engineering Workflow Management 7.0.1 Multilingual	M03XFML	Optional, ADCD May 2021 Edition RSU level 2103	D4BLZ1.gz
24	DB2 V12 Part 1 of 2 Multilingual	M03XGML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBC1.gz
25	DB2 V12 Part 2 of 2 Multilingual	M03XHML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBC2.gz
26	DB2 V11 Part 1 of 2 Multilingual	M03XJML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBB1.gz
27	DB2 V11 Part 2 of 2 Multilingual	M03XKML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBB2.gz
28	CICS 5.6 Multilingual	M03XLML	Optional, ADCD May 2021 Edition RSU level 2103	D4C560.gz
29	CICS 5.5 Multilingual	M03XMM L	Optional, ADCD May 2021 Edition RSU level 2103	D4C551.gz
30	IMS 14.1 Multilingual	M03XNML	Optional, ADCD May 2021 Edition RSU level 2103	D4IME1.gz
31	IMS 15.2 Multilingual	M03XPML	Optional, ADCD May 2021 Edition RSU level 2103	D4IMF1.gz
32	IBM Installation Manager 1.9 Multilingual	M03XQML	Optional, ADCD May 2021 Edition RSU level 2103	D4INM1.gz
33	DB2 Archive Logs Multilingual	M03XRML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBAR.gz

	Name	Part No.	Image Classification	File Name
34	Tivoli Omegamon XE products 5.5.1 Multilingual	M03XSML	Optional, ADCD May 2021 Edition RSU level 2103	D4KAN1.gz
35	WAS V9.0 Part 1 of 2 Multilingual	M03XTML	Optional, ADCD May 2021 Edition RSU level 2103	D4W901.gz
36	WAS V9.0 Part 2 of 2 Multilingual	M03XVML	Optional, ADCD May 2021 Edition RSU level 2103	D4W902.gz
37	Zowe Open Source Project Multilingual	M03XWML	Optional, ADCD May 2021 Edition RSU level 2103	D4ZWE1.gz
38	ADCD Metadata for Tools Multilingual	M03XXML	Optional, ADCD May 2021 Edition RSU level 2103	ADCDTOOLS.XML
39	Extended ADCD Checksum Multilingual	M03XYML	Optional, ADCD package MD5 checksum	may2021_adcd_md5.txt

Validating the downloaded installer file

ZD&T Personal Edition installer now supports the signing of the installer. With the signed installer, you can validate the completeness of the downloaded package by running a command with the signing certificate that is packaged in the installer.

ZD&T Personal Edition installer file `ZDT_Install_PE_V13.3.0.tgz` that you download from IBM is a compressed file. The installer file contains the installer `ZDT_Install_PE_V13.3.0.x86_64` and the corresponding signature file `ZDT_Install_PE_V13.3.0.x86_64.sig`.

To validate the signature file for the installer, run the following command.

```
openssl dgst -sha256 -verify <PUBLICKEYNAME> -signature <signature_file> <FILETOSIGN>
```

For example, run the following command from the directory that contains the extracted installer files.

```
openssl dgst -sha256 -verify signature/public.pem -signature ZDT_Install_PE_V13.3.0.x86_64.sig ZDT_Install_PE_V13.3.0.x86_64
```

If the installation file is valid, you will receive the message of `Verified OK`.

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_V13.3.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PE_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PE_V13.3.0.tgz
```

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

```
./ZDT_Install_PE_V13.3.0.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](#).

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 32](#). 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.

```
z1091-1-10.55.12.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
```

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

```
./clientconfig
```

The following screen is displayed.

```
Gen1 ContactServer.....localhost
Gen1 BackupServer.....
UIM ContactServer.....
UIM Local Serial Random.._
Factory Reset....._

enter="Process reset or save-quit" ESC="quit"
Blank UIM fields=default Non-blank UIM fields=input
```

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](#) 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_V13.3.0.x86_64 --install
```

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

```
./ZDT_Install_PE_V13.3.0.x86_64 --install --net
```

- To install the dependencies, run the following command.

```
./ZDT_Install_PE_V13.3.0.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 32.
- 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 4. 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](#).
3. Decompress all `.gz` volumes by using the Linux command.

```
gunzip *.gz
```

4. Set the environment variables by using the following command.

```
export PATH=/usr/z1090/bin:$PATH
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH
export MANPATH=/usr/z1090/man:$MANPATH
ulimit -c unlimited
ulimit -d unlimited
```

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

```
./Z1091_ADCD_install <path of volume>/C4RES1.ZPD <output path of volume>/C4RES1
```

For example:

```
./Z1091_ADCD_install
/home/ibmsys1/volumes/C4RES1.ZPD
/home/ibmsys1/volumes/C4RES1
```

6. 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.

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

```
chmod 755 *
```

For a complete list of z/OS products in ADCD, see [Products contained in this release](#).

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, ensure that you set the environment variables by using the following command.

```
export PATH=/usr/z1090/bin:$PATH
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH
export MANPATH=/usr/z1090/man:$MANPATH
ulimit -c unlimited
ulimit -d unlimited
```

Then, 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`). Then, ensure the environment variables shown in the example below are defined before you run the script.

For example, you can copy the `./runzpd` script to the `/home/ibmsys1/volumes` directory, and run the script as follows.

```
export PATH=/usr/z1090/bin:$PATH
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH
export MANPATH=/usr/z1090/man:$MANPATH
ulimit -c unlimited
ulimit -d unlimited
./runzpd -d mygenerateddevmap -l cs
```

The script takes the following options:

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

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 `./runzpd` 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. Set the environment variables by using the following command.

```
export PATH=/usr/z1090/bin:$PATH
export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH
export MANPATH=/usr/z1090/man:$MANPATH
ulimit -c unlimited
ulimit -d unlimited
```

4. Enter the following command to start ZD&T Personal Edition.

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

5. 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.

6. Enter the IPL command:

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

For example, `ipl 0a80 parm 0a82CS`

7. 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 December Edition of 2020](#).

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.

```
oprmmsg 'VARY CN(*),ACTIVATE'
oprmmsg '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.Z24D.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 `oprmmsg "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.Z24D.* library. If you want to edit any parm or proc, ADCD has USER.Z24D.* 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](#).

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 udp -- 0.0.0.0/0 0.0.0.0/0 udp 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 -- 0.0.0.0/0 172.26.1.2 tcp dpts:0:21
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:23:1946
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:1948:2021
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpt:2022
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:2023:3269
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:3271:5900
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 tcp dpts:5903:9449
ACCEPT tcp -- 0.0.0.0/0 172.26.1.2 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.
2. 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.`

3. 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 --tunnel_ip=172.26.1.1 # 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
```

4. IPL the ZD&T z/OS with the CS loadparm option.
5. Copy GBLRESOL, TCPDATA, and PROF2 from ADCD.Z24D.TCPPARMS to USER.Z24D.TCPPARMS.
6. Copy FTPD, NFSC, NFSS, PORTMAP, RESOLVER, and TCPIP from ADCD.Z24D.PROCLIB to USER.Z24D.PROCLIB.
7. 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
```

8. Create the member IPNODES in USER.Z24D.TCPPARMS.

```
000001 X.XX.XXX.XXX ZOS-DEV4 ZOS-DEV4.ibm.com
000002 172.26.1.1 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.Z24D.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.Z24D.TCPPARMS(TCPDATA).

```

000072 ; No prefix is required if the TCPIP.DATA file is not being shared.
000073 ;
000074 SOW1:  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.Z24D.TCPPARMS to PROFILE, and edit it.

```

000070 ;This device defines the tunnel
000071 DEVICE PORTA MPCIPA
000072 LINK TAP0 IPAQENET PORTA
000073 HOME 172.26.1.2 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 172.26.1.1 255.255.255.0 = 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	IPv6
Path Address	Name	State	Address	Address
-----	-----	-----	-----	-----
F0	eth0	UP, RUNNING	fa:16:3e:2f:c9:9c	
9.28.160.193				

```

*          virbr0          UP, NOT-RUNNING  fe80::f816:3eff:fe2f:c99c%eth0
192.168.122.1          52:54:00:77:c7:21

*          virbr0-nic      DOWN                52:54:00:77:c7:21
*

A0        tap0            UP, RUNNING      12:3e:b7:80:ad:d1
172.26.1.1

A1        tap1            DOWN             02:a1:a1:a1:a1:a1
*          fe80::103e:b7ff:fe80:add1%tap0

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.

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

```
v net, id=osatr12, 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 30](#).

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 43. – 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="964 590 1451 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 Personal Edition"](#) on page 36.

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_V13.3.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PE_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PE_V13.3.0.tgz
```

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

```
./ZDT_Install_PE_V13.3.0.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](#).

Important: Remember to download the volumes of ADCD May 2020 edition from passport advantage, and decompress the *.gz volume on the same machine where you install ZD&T V13.0.0.

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.

```
z1091-1-10.55.12.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_V13.3.0.x86_64 --update
```

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

```
./ZDT_Install_PE_V13.3.0.x86_64 --update --net
```

After the update completes, you can follow the steps that are described in [“Enabling a license key”](#) on page 38 and [“Configuring Z instances”](#) on page 41.

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 40.

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](#).
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_V13.3.0.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_V13.3.0.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 47.

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 create, and manage images from an existing Z or ADCD packages. Then, you can provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to your target environment.

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, and manage the images, and provision instances from an existing Z or ADCD packages.
- Creating and managing images from various sources.
- Provisioning 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 artifacts such as volumes, data sets, CICS, Db2 and so on, and transfer the artifacts to the storage server. Then, you can provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to your target environment.

The Enterprise Edition web server provides the following features:

- Integrating with IBM Application Discovery (AD) to automate the identification of data sets for extraction.
- Integrating with company LDAP account to enable administrators to set up accounts with minimum efforts.
- Scheduling the extraction or provisioning 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 provisioning process can install ZD&T emulator, and make the system ready. You do not need to remember the ZD&T native commands.
- 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 provision the necessary volumes or data sets that are required to the target environment.

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 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 storage server to hold the required artifacts.

Note: ZD&T Enterprise Edition web server, 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 activate the emulated Central Processors (CPs) that are used by a ZD&T instance except when the Software-based License Server is activated with a license key file and is accessible by the program, because it provides the proof of license entitlement. For other functions that are provided by ZD&T Enterprise Edition, license activation is not required.

Each license key file is generated and is entitled to a number of emulated central processors. Central processors are also known as 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.

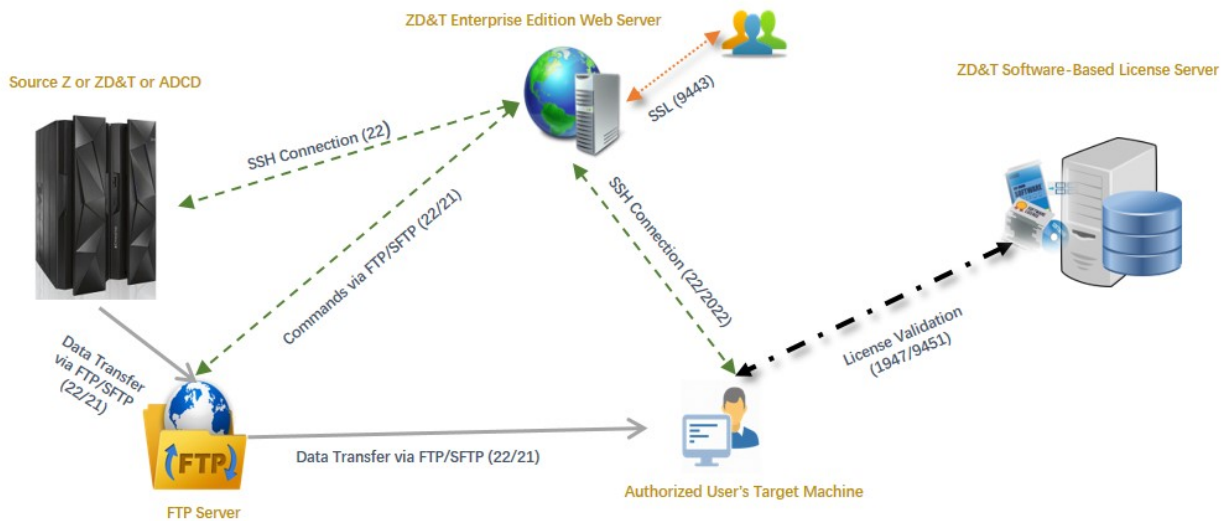


Figure 5. The AUSS and various components of Enterprise Edition

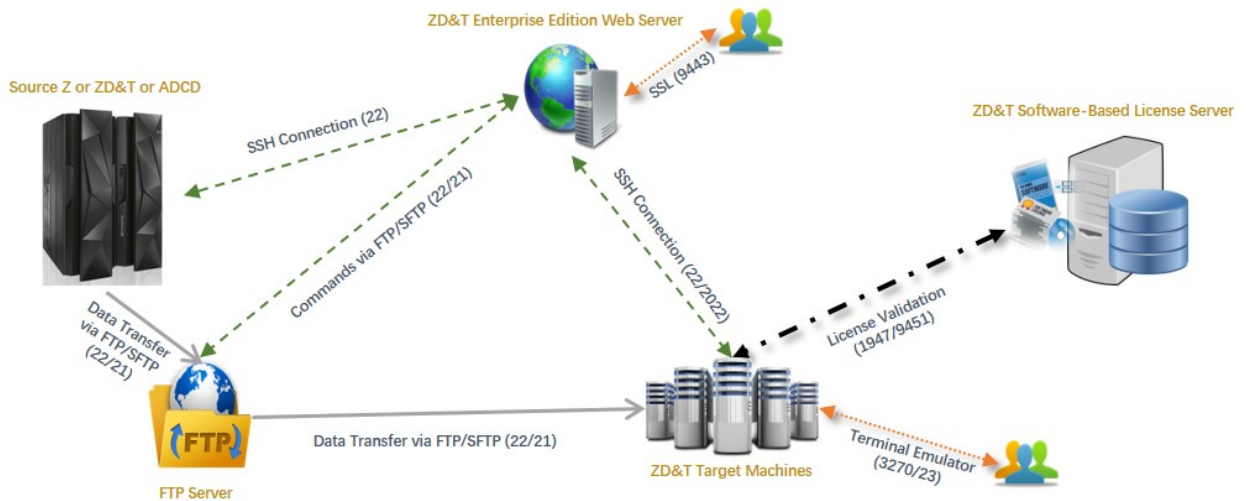


Figure 6. The RVU and various components of Enterprise Edition

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.

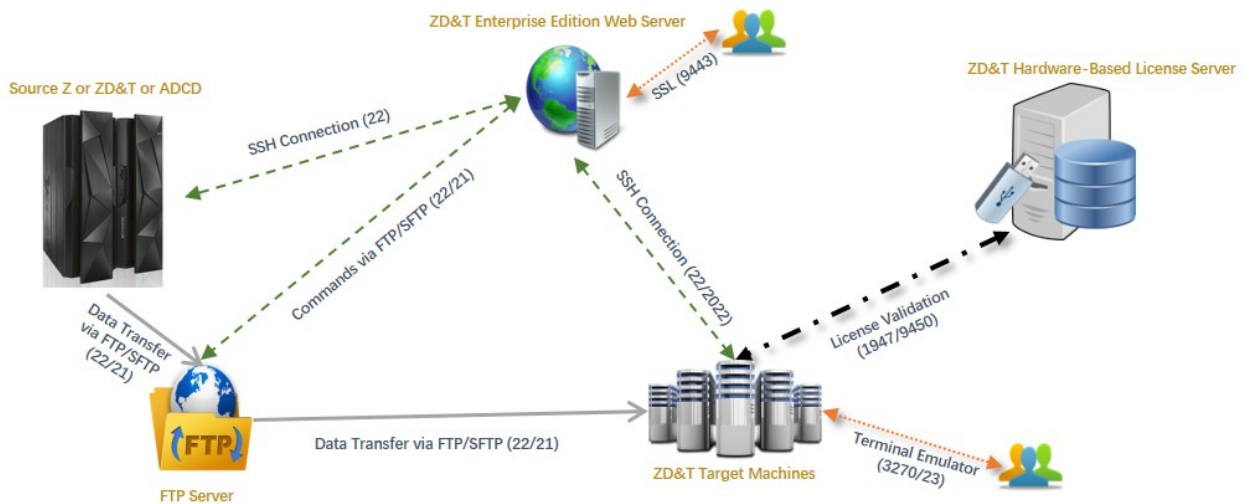


Figure 7. The Hardware-based License Server and various components of Enterprise Edition

Rational Tokens

Rational License Key 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.

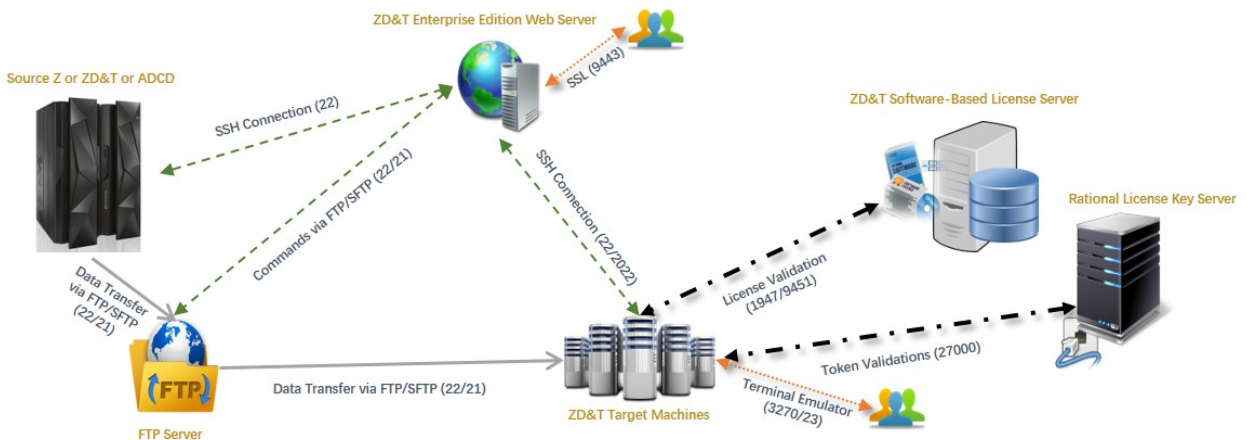


Figure 8. The Software-based License Server with Rational Tokens, and various components of Enterprise Edition

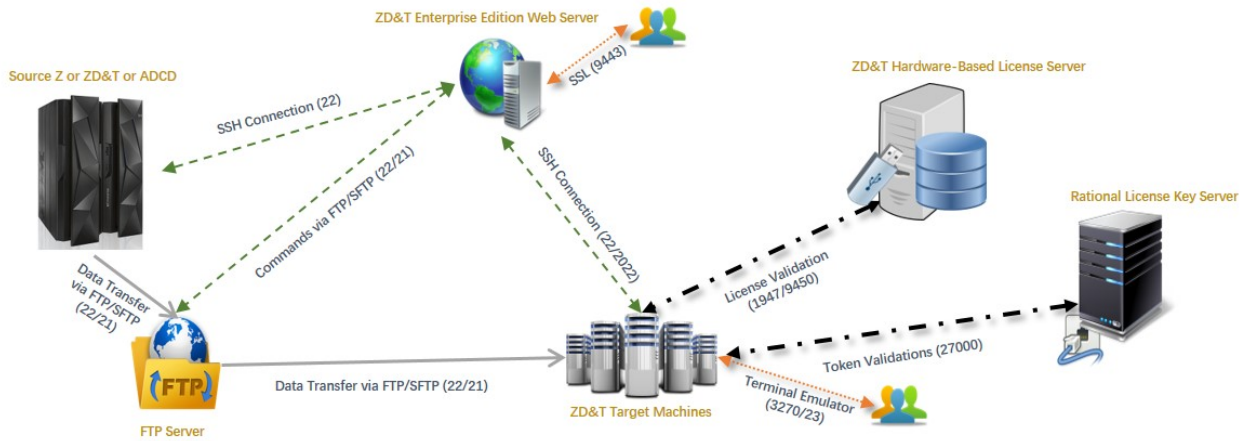


Figure 9. The Hardware-based License Server with Rational Tokens, and various components of Enterprise Edition

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 2.4, and JES2. You can select the software packages that you need, and provision the packages out of ADCD to the target environment. Alternatively, you can use Z images from source Z.

For a complete list of z/OS products in ADCD, see [Products contained in this release](#).

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.

Installing Enterprise Edition		Required/Optional	Complete
1	<p>“Prerequisites” on page 58</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"/>

Table 10. New installation checklist for Enterprise Edition (continued)

Installing Enterprise Edition		Required/ Optional	Complete
3	“Enabling the Software-based License Server” on page 90 Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
4	“Installing the Enterprise Edition web server” on page 100 Describes the steps to install the Enterprise Edition.	Required	<input type="checkbox"/>
5	“Installing the ZD&T z/OS Extraction Utilities” on page 104 If you want to create components from IBM Z platforms, install the ZD&T z/OS Extraction Utilities .	Conditional “1” on page 56	<input type="checkbox"/>
6	“Configuring the Enterprise Edition web server” on page 103 Describes the steps to configure the Enterprise Edition server environment before you start ZD&T Enterprise Edition.	Required	<input type="checkbox"/>
7	“Optional: Starting the Enterprise Edition web server” on page 102 Describes the steps to start the Enterprise Edition web server.	Required	<input type="checkbox"/>
8	“User's Guide” on page 107 <ul style="list-style-type: none"> • “Web UI User's Guide” on page 114 • “REST APIs User's Guide” on page 149. Describes the steps to create images, and provision instances from created images by using the Enterprise Edition web user interface or REST APIs.	Required	<input type="checkbox"/>
Note: 1. Required only if you want to create components from IBM Z platforms.			

Table 11. Update checklist for Enterprise Edition

Updating Enterprise Edition		Required/ Optional	Complete
1	“Prerequisites” on page 58 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"/>
3	“Updating the Enterprise Edition web server” on page 106 Describes the steps to update Enterprise Edition.	Required	<input type="checkbox"/>

Table 11. Update checklist for Enterprise Edition (continued)

Updating Enterprise Edition		Required/ Optional	Complete
4	“Configuring the Enterprise Edition web server” on page 103 Describes the steps to configure the Enterprise Edition server environment before you start ZD&T Enterprise Edition.	Required	<input type="checkbox"/>
5	“Optional: Starting the Enterprise Edition web server” on page 102 Describes the steps to start the Enterprise Edition web server.	Required	<input type="checkbox"/>
6	“User's Guide” on page 107 <ul style="list-style-type: none"> • “Web UI User's Guide” on page 114 • “REST APIs User's Guide” on page 149. Describes the steps to create images, and provision instances from created images by using the Enterprise Edition web user interface or REST APIs.	Required	<input type="checkbox"/>

Table 12. Update checklist for license server

Updating license server		Required/ Optional	Complete
1	“Returning a software-based license key” on page 96 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 98 Describes the steps to update the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Enabling the Software-based License Server” on page 90 Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>

Table 13. Update checklist for license

Updating license		Required/ Optional	Complete
1	“Returning a software-based license key” on page 96 Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	“Enabling the Software-based License Server” on page 90 Describes the steps to install, enable, update or uninstall the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>

Table 14. Uninstallation checklist for Enterprise Edition

Uninstalling Enterprise Edition		Required/ Optional	Complete
1	“Returning a software-based license key” on page 96 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 99 Describes the steps to update the Software-based License Server or Hardware-based License Server	Required	<input type="checkbox"/>
3	“Uninstalling the Enterprise Edition web server” on page 106 Describes the steps to uninstall Enterprise Edition.	Required	<input type="checkbox"/>

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](#).

For other prerequisites, see the following details.

- [“Storage server requirements” on page 58](#)
- Source environments
 - [“z/OS system requirements” on page 59](#)
 - [“Db2 extraction requirements” on page 59](#)
- Target environments
 - [“Linux target environments requirements” on page 60](#)
 - [“OpenStack Cloud target environments requirements” on page 62](#)
 - [“Docker target environments requirements” on page 62](#)
 - [“Red Hat OpenShift target environments requirements” on page 64](#)

Storage server requirements

To install and run ZD&T Enterprise Edition, a 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 storage server or to the 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 V2.4 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.

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 storage server.
- To use SFTP, Java™ 1.6 or later versions must be installed, and the PATH needs to be specified in the \$HOME/.profile and 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 components from IBM Z mainframe volumes](#) and [Creating 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.

Db2 extraction requirements

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 DSNTDP2.

Restriction:

ADCD z/OS V2R4 May Edition of 2020 is distributed with Db2 V12 at Function Level 506 (V12R1M506). Beginning at Function Level 504, applications that are bound at this function level or higher no longer support segmented (non-UTS) and partitioned (non-UTS) tablespaces. If Db2 table extraction is performed from a source system where tablespaces are deprecated types, where Db2 Admin Tool is used to create DDL for the component, and where Db2 and Db2 Admin Tools are not at levels to support Function Level 504 or higher, the provisioning4 of the Db2 component might fail when Db2 objects are created.

Linux target environments requirements

To automatically provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to a Linux target environment, the following requirements are needed.

- If you choose to install the required Linux packages during the provisioning, 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.
- FTP is only required when the storage system is set to FTP.
- The root permission is needed for the users who are responsible for provisioning.
- 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 110.

- [Accessing to software repository to run YUM or apt - get commands.](#)
- [Users and group settings.](#)
- [Sudo access configuration.](#)
- [Network configuration.](#)

- **Accessing 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
```

Note: `yum -y install ftp` is optional, enter it when the storage server is configured to use FTP.

- 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: `apt-get -y install ftp` is optional, enter it when the storage server is configured to use FTP.

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 provisioning 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 provision instances from created images, make sure to create a new group 'zpdtd' in the target environment if the group does not exist.

- If you use the root user ID to provision instances, create a user ID 'ibmsys1' if the user ID does not exist, and assign the user ID 'ibmsys1' to the group 'zpdtd'.
- If you use a non-root user ID to provision instances, assign the user ID to the group 'zpdtd'.

• 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 provisioning, 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 provisioning 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 to ensure that the emulated environment can be accessible. The only requirement is to route a port number to port 22 on the emulated z/OS. The port number to be routed is the one that you will specify when you configure the source system on the ZD&T web server.

To configure the network, complete the following steps:

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

```
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 2022 -j DNAT --to 10.1.1.2:22
iptables -A FORWARD -p tcp -d 10.1.1.2 --dport 2022 -j ACCEPT
```

3. Run the command `echo 1 > /proc/sys/net/ipv4/ip_forward`.

Note:

- `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.
- 2022 is the port number that will be routed to port 22.
- 10.1.1.2 is the IP address of network interface `tap0`, which can be found by running the command `find_io`.

OpenStack Cloud target environments requirements

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 110](#).

Docker target environments requirements

In a normal Docker setup, a container image is pushed to a remotely accessible Docker registry. Then, the container image can be pulled by instances that need to run the image by using the Docker command-line utilities such as `docker pull` or `docker run`.

ZD&T Enterprise Edition 13.0 provides a container image that is named `wazi-sandbox` to run an emulated z/OS environment in a container. To use the container image, complete the following steps.

1. Use HTTPS that is protected by using the TLS cryptographic protocol to communicate to the Docker daemon.
2. Load the image directly to the local Docker image registry of the system that is running the container.

3. Create the container from the image in privileged mode with a small set of port bindings. A Docker volume that contains all the persistence data such as volumes will be run on the emulator.

When the wazi-sandbox container is started, ZD&T Enterprise Edition can use the function ZD&T Instance Controller that is provided by the container to configure the emulator, acquire the ZD&T image, start the emulator, and finally perform the IPL of z/OS.

To complete the task, the following initial setup needs to be done before you create a provisioning.

- **Configuring the Docker daemon for HTTPS communication**

To configure the Docker daemon for HTTPS communication, refer to the instructions in the [Protect the Docker daemon socket](#).

After the Docker daemon is configured for HTTPS communication by using the TLS cryptographic protocol, save the files for the CA certificate (ca.pem), server certificate (cert.pem), and client certificate (key.pem).

- **Planning the port mapping for Docker containers**

Each Docker environment supports a maximum of five containers. In a Docker environment, each container runs their own emulated z/OS instances and will be allocated 100 ports for clients to access services on each of those emulated z/OS instances. The entire set of ports for all the containers (up to a maximum of five) must be contiguous and specified in intervals of one hundred. The ephemeral port range is recommended.

For each container, the following port mapping is in place:

- xxy00 maps to port 3270 in the container
- xxy21 maps to port 21 (FTP) in the container
- xxy22 maps to port 22 (SSH) in the container
- xxy23 maps to port 23 (Telnet) in the container
- xxy99 maps to port 8443 (ZD&T Instance controller) in the container

Where xx is the number in the thousands and y is the number in the hundreds.

For example, if you plan to use 40000 as the start port and provision two Docker containers, the first container that is provisioned will use ports 40000 - 40099 and the second container that is provisioned will use ports 40100 - 40199.

The first provision to a container has the ports 40000 - 40099 from the hosting system that is allocated to it and has the following port mappings:

- 40000 → 3270
- 40021 → 21
- 40022 → 22
- 40023 → 23
- 40099 → 8443

The second container will have the following port mappings:

- 40100 → 3270
- 40121 → 21
- 40122 → 22
- 40123 → 23
- 40199 → 8443

Red Hat OpenShift target environments requirements

Red Hat OpenShift

Sandbox is supported in customer managed and IBM Cloud® managed Classic or VPC OpenShift 4.8 clusters running on x86_64 architecture

SecurityContextConstraints

The Sandbox Operator and wazi - sandbox container use a privileged security context constraints (SCC).

```
allowHostDirVolumePlugin: true
allowHostIPC: true
allowHostNetwork: true
allowHostPID: true
allowHostPorts: true
allowPrivilegeEscalation: true
allowPrivilegedContainer: true
allowedCapabilities:
- '*'
allowedUnsafeSysctls:
- '*'
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  type: RunAsAny
kind: SecurityContextConstraints
metadata:
  annotations:
    kubernetes.io/description: 'wazi-sandbox-operator allows access to all privileged and
host
  features and the ability to run as any user, any group, any fsGroup, and with
any SELinux context.'
  name: wazi-sandbox-operator
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities: null
runAsUser:
  type: RunAsAny
seLinuxContext:
  type: RunAsAny
seccompProfiles:
- '*'
supplementalGroups:
  type: RunAsAny
users:
- wazi-sandbox-operator
volumes:
- '*'
```

Figure 10. Custom SecurityContextConstraints definition

The wazi-sandbox-volume-copy container uses the anyuid SCC or a modified definition.

```

allowHostDirVolumePlugin: false
allowHostIPC: false
allowHostNetwork: false
allowHostPID: false
allowHostPorts: false
allowPrivilegeEscalation: true
allowPrivilegedContainer: false
allowedCapabilities: null
apiVersion: security.openshift.io/v1
defaultAddCapabilities: null
fsGroup:
  ranges:
    - max: 2500
      min: 2105
  type: MustRunAs
groups: []
kind: SecurityContextConstraints
metadata:
  name: wazi-sandbox-volume-copy
priority: null
readOnlyRootFilesystem: false
requiredDropCapabilities:
  - KILL
  - MKNOD
runAsUser:
  type: MustRunAsRange
  uidRangeMax: 2500
  uidRangeMin: 2105
seLinuxContext:
  type: MustRunAs
supplementalGroups:
  ranges:
    - max: 2500
      min: 2105
  type: MustRunAs
users:
  - wazi-sandbox-volume-copy
volumes:
  - '*'

```

Figure 11. Custom SecurityContextConstraints definition

Required resources

The required cluster resources for Sandbox depend on:

- The cluster storage
- The number of expected sandbox instances
- The resource requirements from the emulator machine characteristics and device map file (devmap) for the sandbox instances
- The size of the z/OS volume files for the sandbox instance

Because the compute resources for the cluster storage depend on the specific storage driver and configuration, those compute resources are excluded from the calculations here. The cluster must have sufficient extra compute resources to manage the required storage. Contact your cluster administrator or cloud provider if you need help on these requirements.

Overall, the cluster sizing includes the following requirements:

- The base resource requirements for control nodes
- Variable requirements for worker nodes that satisfy the scheduling capacity for the expected number of instances including storage, and compute requirements for that storage

The following tables assume that the devmap specifies P processors including both CP and zIIP and M GiB of memory, and the z/OS volume files are V GiB in total.

Table 15. Cluster base resources		
	Count	Memory (GiB)
Control nodes	3	16/node

<i>Table 16. Scheduling capacity for worker nodes</i>				
Software	Memory (GiB)	CPU (cores)	Ephemeral storage (GiB)	Persistent storage (GiB)
1 single sandbox instance	$M + 2$	$P + 1$	2	$V * 1.125$

At a minimum, worker nodes must meet the scheduling requirements for a single instance. To support multiple instances, scale up the worker nodes.

For example, for the Extended ADCD image available with Sandbox, $M = 8$ GiB, $P = 3$, and $V = 270$ GiB.

To run each instance, the cluster requires worker nodes with a capacity of at least 10 GiB memory, 4 cores, 2 GiB of ephemeral storage, and approximately 304 GiB of persistent storage.

To run 5 instances, the cluster requires worker nodes with a total capacity of at least 50 GiB memory, 20 cores, 480 GiB of ephemeral storage, 1520 GiB of persistent storage, and also any extra compute capacity to support that storage. This might be 5 nodes each with a capacity for a single sandbox instance, or 2 with a capacity for 3 instances, or a single large node with a capacity for all 5.

Complete the following procedure before you provision instances to an Red Hat Open Shift target environment:

1. [“Installing Sandbox in a cluster with internet access” on page 66](#) or [“Installing Sandbox in a cluster with a restricted network” on page 67](#)
2. [“Installing the Sandbox Operator” on page 69](#)
3. [“Verifying the installation of the Sandbox Operator” on page 70](#)
4. (Optional) [“Installing and using the IBM License Service” on page 71](#)
5. (Optional) [“Configuring the cluster” on page 71](#)
6. [“Configuration reference” on page 75](#)

Notes:

- To provision an instance to a Red Hat OpenShift target environment, the protocol must be set to SFTP, and FTP is not supported now.
- Due to the requirements of setting up and running the zPDT emulator, the system container requires privileged access. This elevated access might cause security risks because the access might be knowingly or unknowingly used to affect the hosting system. To avoid the security risks, you can run development and test workloads like Sandbox, and production workloads in a separate cluster. Also, the cluster administrators need to be informed about the privileged access that is granted to the service account and system containers that are executed by the access. For more information about securing your cluster, see [Kubernetes documentation](#).

Installing Sandbox in a cluster with internet access

You can use either the OpenShift CLI or web console to install Sandbox in a cluster with access to public registries by enabling the IBM Operator Catalog in the cluster and then installing the Sandbox Operator.

Enabling the IBM Operator Catalog using the OpenShift CLI

To install the IBM Operator Catalog by using the OpenShift CLI, complete the following steps:

1. From your local computer, create a YAML file called `catalog_source.yaml` with the following content:

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
```

```
displayName: IBM Operator Catalog
publisher: IBM
sourceType: grpc
image: icr.io/cpopen/ibm-operator-catalog
updateStrategy:
  registryPoll:
    interval: 45m
```

2. From the command line, log in to your OpenShift cluster by using the `oc login` command.
3. Run the following command:

```
oc apply -f catalog_source.yaml -n openshift-marketplace
```

4. Verify the installation by running this command:

```
oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
```

If the installation was successful, you should see output that is similar to this:

NAME	DISPLAY	TYPE	PUBLISHER	AGE
ibm-operator-catalog	IBM Operator Catalog	grpc	IBM	50s

You can now install the Sandbox Operator as described in [“Installing the Sandbox Operator” on page 69](#).

Enabling the IBM Operator Catalog using the OpenShift web console

To install the IBM Operator Catalog by using the OpenShift web console, complete the following steps:

1. Log in to the OpenShift web console.
2. Click the **Import YAML** icon and copy and paste the following resource definition into the Import YAML editor.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: IBM Operator Catalog
  publisher: IBM
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog
  updateStrategy:
    registryPoll:
      interval: 45m
```

3. Click **Create**.

You can now install the Sandbox Operator as described in [“Installing the Sandbox Operator” on page 69](#).

Installing Sandbox in a cluster with a restricted network

Before you begin

The following tools and resources are needed:

- A computer with internet access on which you can run the required commands. This computer must also have access to the cluster, and is referred to as a `bastion host`.
- A private Docker registry that can be accessed by the cluster and the bastion host, and which will be used to store all images in your restricted network.
- `cloudctl`
- `skopeo`
- `oc`
- `kubectl`

- OpenSSL
- htpasswd

About this task

Use the following CASE bundle commands to install Wazi Developer Sandbox into an air gap environment.

Procedure

1. Run the following command to save and download the CASE bundle. An `images.csv` file will be generated in the specified output directory, which is needed in later steps.

```
cloudctl case save \
--case https://github.com/IBM/cloud-pak/raw/master/repo/case/ibm-wazi-developer-
sandbox-1.0.0.tgz \
--outputdir /tmp/wazi-sandbox \
--tolerance 1
```

2. Extract the case from the tar file with the following command:

```
tar -xvf /tmp/wazi-sandbox/ibm-wazi-developer-sandbox-1.0.0.tgz
```

Note: The following steps are run from the `/tmp/wazi-sandbox` directory. If you want to run from another directory, add full path to the `--case` parameter in the following commands, for example:

```
cloudctl case launch \
--case /tmp/wazi-sandbox/ibm-wazi-developer-sandbox
```

3. Set source registry credentials. Configure the credentials so that the CASE bundle can pull images from a public registry such as `icr.io` to your bastion host. Run the action `configure-creds-airgap` command to store and cache the registry credentials in a file in the `$HOME/.airgap/secrets` folder on your file system. To authenticate with `cp.icr.io`, run the following command:

```
SOURCE_REGISTRY='cp.icr.io'
SOURCE_REGISTRY_USER='cp'
SOURCE_REGISTRY_PASS='[your entitlement key]'
cloudctl case launch \
--case ibm-wazi-developer-sandbox \
--namespace namespace-title \
--inventory waziSandboxOperatorSetup \
--action configure-creds-airgap \
--args "--registry $SOURCE_REGISTRY --user $SOURCE_REGISTRY_USER --pass
$SOURCE_REGISTRY_PASS" \
--tolerance 1
```

4. Set target registry credentials. Use the following command to set credentials so that the CASE bundle can authenticate into your target bastion host docker registry.

```
TARGET_REGISTRY='[your bastion host docker registry address]'
TARGET_REGISTRY_USER='[your bastion host docker registry user]'
TARGET_REGISTRY_PASS='[your bastion host docker registry password]'
cloudctl case launch \
--case ibm-wazi-developer-sandbox \
--namespace namespace-title \
--inventory waziSandboxOperatorSetup \
--action configure-creds-airgap \
--args "--registry $TARGET_REGISTRY --user $TARGET_REGISTRY_USER --pass
$TARGET_REGISTRY_PASS" \
--tolerance 1
```

5. Mirror images to the bastion host docker registry. Use the following command to mirror images from their current location to a registry on your bastion server, using the same value `TARGET_REGISTRY` as the previous step.

```
cloudctl case launch \
--case ibm-wazi-developer-sandbox \
--namespace namespace-title \
--inventory waziSandboxOperatorSetup \
--action mirror-images \
```



```
--args "--registry $TARGET_REGISTRY --inputDir /tmp/wazi-sandbox" \  
--tolerance 1
```

6. After you have mirrored your images, you can validate that they were transferred to the local registry by using a `curl` command like the following one.

Note: The python command cleans up the JSON into a more readable format.

```
curl -k -u ${TARGET_REGISTRY_USER}:${TARGET_REGISTRY_PASS} https://${TARGET_REGISTRY}/v2/  
_catalog?n=6000 | python -m json.tool
```

7. Configure the cluster. When modifying the Sandbox custom resource definition file, ensure that you use the latest version of the images found in the `images.csv` file that is generated from step 1. After all images are copied to the internal registry on a bastion host, update the cluster such that if there is a request to pull an image from any of the external repositories specified by the CASE bundle, the cluster will try to pull them from the internal repository instead.

OCP includes a custom resource called `ImageContentSourcePolicy` that can automatically redirect image pull requests from a specified registry location to an alternative location. Additionally, create a global image pull secret so that your product can pull the images from your internal repository. Both of these actions are handled by the following command:

```
cloudctl case launch \  
--case ibm-wazi-developer-sandbox \  
--namespace namespace-title \  
--inventory waziSandboxOperatorSetup \  
--action configure-cluster-airgap \  
--args "--registry $TARGET_REGISTRY --inputDir /tmp/wazi-sandbox --dryRun" \  
--tolerance 1
```

Note: Applying `ImageContentSourcePolicy` causes cluster nodes to recycle. This process is disruptive and takes a long time to run as it requires restarting a number of processes on each of the nodes of the cluster. Take caution when running this command.

8. Install Sandbox catalog source using the following command:

```
cloudctl case launch \  
--case ibm-wazi-developer-sandbox \  
--namespace openshift-marketplace \  
--inventory waziSandboxOperatorSetup \  
--action install \  
--tolerance 1
```

Installing the Sandbox Operator

After the CASE bundle is successfully installed, you will be able to see IBM Wazi Developer Sandbox Operator inside OperatorHub. Install the Sandbox Operator and it will be managing the lifecycle of sandbox instances.

Audience: Cluster administrators

After you install and configure the requisite components, you will see that the Operator named IBM Wazi Developer for Red Hat CodeReady Workspaces Sandbox is available in the cluster catalog.

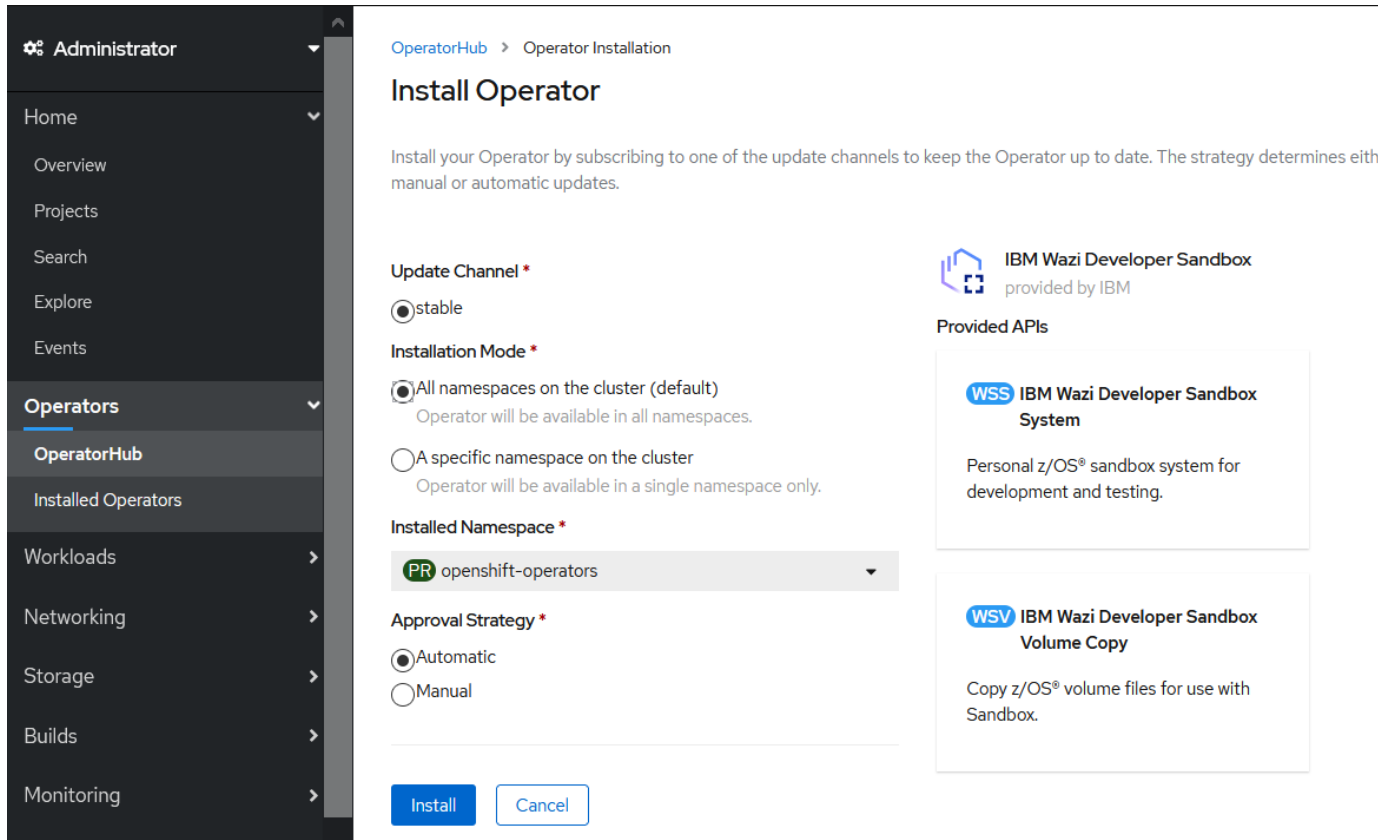
The Operator is namespace-scoped, but it can be installed in one or more namespaces in the cluster. Each instance watches resources only in its namespace.

Multiple sandbox instances can be running on the cluster at the same time, either in different namespaces or within a single namespace. Within each namespace, the name of the sandbox instance that is specified in the Custom Resource file must be unique.

To install the Operator from the OpenShift web console, follow these steps:

1. Select the Administrator perspective.
2. Choose **Operators > OperatorHub**.
3. Filter by keyword for **IBM Wazi**.
4. Click the **IBM Wazi Developer Sandbox** tile.

5. Click **Install**. And you can see the Create Operator Subscription view as follows.



6. Under **Installation Mode**, select **All namespaces on the cluster** or **A specific namespace on the cluster** based on your preference.

All namespaces on the cluster

If you choose this option, the Operator will be available in all namespaces, including the namespaces that might have no need for it.

A specific namespace on the cluster

Recommended. If you choose this option, select the target namespace. Choosing a specific namespace is recommended because Sandbox is privileged and can consume significant cluster resources. Installing only to namespaces for authorized users gives the cluster administrator better control over who can use those privileges and resources.

7. Click **Install**.

Verifying the installation of the Sandbox Operator

After the images are pulled and configured on OpenShift, you can verify that whether the Operator was deployed successfully.

After the Operator was installed from OperatorHub into a project or namespace, verify it by searching for Wazi Sandbox in the Developer Catalog in the OpenShift web console, or by running the following command to ensure that the deployment succeeded. Ensure that your `oc/kubectl` command is pointing to the namespace that the Operator was installed:

```
oc get deployments
```

If the deployment succeeded, the `ibm-wazi-developer-sandbox-operator` item is displayed and its **READY** status is `1/1`.

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
ibm-wazi-developer-sandbox-operator	1/1	1	1	9s

Installing and using the IBM License Service

Sandbox uses a third-party License Server for license validation, but IBM License Service can be used to track Virtual Processor Core (VPC) usage.

You can install IBM License Service with or without the IBM Operator Catalog:

- With the IBM Operator Catalog:

If IBM License Service is not installed in your cluster but the IBM Operator Catalog is installed, you can install the IBM Licensing Operator directly from OperatorHub. If the IBM Operator Catalog is not installed, you can install it as described in [“Installing Sandbox in a cluster with internet access”](#) on page 66.

- Without the IBM Operator Catalog:

If you do not want to use the IBM Operator Catalog, follow the instructions for https://github.com/IBM/ibm-licensing-operator/blob/master/docs/Content/Automatic_installation.md.

When IBM License Service is running, follow [Retrieving license usage data from the cluster](#) to track VPC usage.

IBM License Service tracks actual VPC resource allocation in the cluster and Sandbox might use multiple VPCs for an instance, so the number reported will be the number of emulated Z processors in the instance plus one. For example, an instance with 3 Z CPs will correctly report 4 VPC resources used, and not the license entitlement. The third-party License Server is required for reporting actual entitlement.

Configuring the cluster

You can customize the sandbox instances provisioning to override the default settings. The configurations are recommended if external access to the sandbox instance is required or if changes to the sandbox instance are expected to be persistent.

Audience: Cluster administrators

The following configurations are recommended if the mentioned use cases apply to you and the developers on your team. Configuration changes take effect upon developers creating sandbox instances from the Sandbox Operator.

- [“Configuring ingress cluster traffic”](#) on page 71
- [“Configuring cluster persistent storage”](#) on page 73

Configuring ingress cluster traffic

A sandbox instance is a virtualized z/OS environment. To use it, developers need to access it like accessing a physical system or a virtual machine. If external access to the sandbox instance is required, you must set up the mechanism that allows the routing of external traffic into the cluster.

Note: External access might not be required, for example, if developers are using IBM Wazi Developer for CodeReady Workspaces in the same cluster.

A typical use case of external access is that when developers want to access the z/OS instance that is running in the sandbox from a client outside of the cluster. See the following examples of external clients:

- External 3270 terminal emulator that is running on developers' workstations
- IBM® Wazi Developer for Eclipse that is running on developers' workstations
- IBM Wazi Developer for VS Code that is running on developers' workstations
- IBM Wazi Developer for CodeReady Workspaces that is running in a different cluster

The OpenShift® platform provides methods for communicating from outside the cluster with the services that run in the cluster. When the Sandbox Operator creates an instance, by default it creates a Service internal to the cluster of type ClusterIP that exposes secure ports in the instance to other namespaces in the cluster, and creates a Route for Wazi host components that are HTTPS services, including RSE API, the Debug Profile Service, and z/OSMF.

If this default is not appropriate, for example, if you do not have all Wazi host components installed in the instance, or you need access to unsecured or do not work with Route, you can change the way that Sandbox creates Services by using the `portProfile` and `zPorts` configuration elements.

portProfile

Use `portProfile` to configure what ports to expose and how to expose them. The `portProfile` has two sub-elements: `profile` and `scope`, which are described as follows:

Use `profile` to control what ports to expose.

<i>Table 17. profile options</i>	
Profile	Description
wazi	Default. Includes encrypted ports of Wazi Developer's host components.
wazi-all	Includes encrypted and unencrypted ports of Wazi Developer's host components.
custom	Empty profile. No ports will be exposed by default unless they are given in <code>zPorts</code> .

Use `scope` to control how to expose the ports specified in the profile. The `scope` options are as follows:

<i>Table 18. scope options</i>	
Scope	
route	Default. Exposes ports on the OpenShift cluster. It also exposes the ports that are modern, web-based protocols outside the cluster using OpenShift Routes.
nodeport	Exposes what route does, and everything else it exposes using a NodePort Service.
cluster	Exposes ports only on the OpenShift cluster using a ClusterIP Service.

zPorts

You can use the `zPorts` to expose other ports that are not covered by `portProfile`, such as custom applications on the instance, or provide alternate behavior for the Wazi ports.

`zPorts` has the following three sub-elements: `cluster`, `route`, and `nodeport`, which work similarly to the `scope`.

<i>Table 19. Sub-elements of zPorts</i>	
Elements	Description
cluster	Add ports to a ClusterIP service
route	Add ports a ClusterIP service and also add a Route
nodeport	Add ports to a NodePort service

All three sub-elements have `name` and `port` to identify the port. The `route` sub-element also has a `tls` sub-element to control how TLS is handled.

For more details, see [“Configuration reference” on page 75](#).

If you do not want to use the custom resource to create services and routes for ingress, set `portProfile.profile: custom`, leave out `zPorts`, and manually create any `Service`, `Route`, or `Ingress`.

Note: The service configuration element in Sandbox 1.3 and earlier releases was replaced with `portProfile` and `zPorts`. If you have a `service` section in your custom resource, it will create the same `NodePort` service as it did in earlier releases; but if you do not, you will notice the services exposed by default have changed slightly. If you require the old default behavior, add a `service` section with the ports you require. But `service` will be removed in a future release, so it is recommended to move to `portProfile` and `zPorts`.

Configuring network policies

OpenShift uses Kubernetes `NetworkPolicy` resources to control network traffic to pods and namespaces, which is in much the same way a firewall on a computer can block or allow traffic.

By default, when a sandbox instance is created, a `NetworkPolicy` is created that allows all traffic to and from the sandbox instance. This allows external and internal users to connect to z/OS without additional configuration.

If you want to change the default `NetworkPolicy`, specify other `NetworkPolicy` that are provided the `WaziSandboxSystem` custom resource, which are shown as follows:

Value	Description
<code>allow-all</code>	Creates the default <code>NetworkPolicy</code> .
<code>namespace-only</code>	Allows traffic to the instance only from other Pods in the same namespace.
<code>deny-all</code>	Blocks all traffic to the instance.
<code>none</code>	Creates no <code>NetworkPolicy</code> .

The access to the instance is the union of all network policies that select the instance pod. You can adjust the access by adding more `NetworkPolicy` objects manually. For example, you can start with the `deny-all` policy, and then manually add policies to specific addresses, namespaces, or pods that are on the allow list.

For more information, see [Network Policies](#) in Kubernetes documentation.

Configuring cluster persistent storage

Sandbox supports OpenShift Container Storage and was tested with the Rook-Ceph® operator, and NFS. On IBM Cloud, Sandbox was tested with `ibmc-block-gold` and `ibmc-block-custom` storage.

If using NFS, the group ID of the storage must match the group ID for Sandbox by using the `fsGroup` option.

The z/OS volumes are not encrypted on the `PersistentVolumeClaim` by Sandbox. If sensitive data is to be stored in z/OS storage, you can protect it in two ways:

- Use encryption on z/OS to encrypt the data on z/OS.
- Configure passive encryption on the cluster for the storage used by Sandbox, so the `PersistentVolumeClaim` used by Sandbox is encrypted at rest. The details of setting up encrypted storage are specific to the cluster and underlying storage, and are beyond the scope of this document. Consult your cluster administrator.

Sandbox stores z/OS volumes for each instance on a `PersistentVolumeClaim`.

Use cases

You can manage the [persistent storage](#) for sandbox instances in either of the following ways:

- Let the [Operator](#) manage the storage automatically.

Each sandbox instance has a copy of the Extended ADCD volumes, so each sandbox instance might require 300 GB of storage, which is the default size of the [claims](#) that are automatically created.

Use case: A clean environment is needed every time the instance starts, and changes to the z/OS system or data on it do not need to be kept. For example, you can use this approach to set up a build or test pipeline that always wants a clean environment and copies out any build or test artifacts before deleting the instance.

- You provision the storage for sandbox instances manually. For more information, see [How to provision the storage for sandbox instances manually](#).

Use case: If changes to the z/OS system or data stored there are important, manual provisioning is recommended.

Both options might require some preparation for the cluster to choose appropriate storage drivers, create or choose storage classes, and set the default storage classes. For more information, see [OpenShift documentation](#).

Ensure that there is sufficient storage available because many sandbox instances are expected to be created at one time.

Storage must be writable by the pod that runs the sandbox instance. By default, the Sandbox Operator mounts the storage with a default `fsGroup` value in the `securityContext` of the pod. If the storage driver that is being used does not support changing the group ID, you might need to set `fsGroup` in the custom resource definition to match the group ID of the storage. See the [OpenShift documentation](#) for information about storage group IDs, and the [“Configuration reference” on page 75](#) for using `fsGroup` in the custom resources for Sandbox.

Provision the storage for sandbox instances manually

- The storage class must support `ReadWriteOnce` [access mode](#); no other modes are supported.
- The default size of the claims that are automatically created is 300 GB. For larger or smaller images, or if you plan to add files to the image, you need to set the size when you create the claim.
- As the volumes will hold z/OS instance data, you must set up appropriate access controls and encryption of the storage.
- For ease of use, use a driver with [dynamic storage](#) capability. However, if dynamic provisioning is not available, you need to set `spec.persistence.useDynamicProvisioning: false` in the [Custom Resource](#) to prevent the Operator from using dynamic provisioning.
- For better performance, use fast storage that is close to the nodes where the sandbox instances will run.
- To allow fast and easy copying of `PersistentVolumeClaim` that has cloud-ready z/OS volume images, use a storage class and driver that supports [CSI volume cloning](#).

Optionally, you can create a custom storage class for sandbox storage, and set up a default storage class for the cluster. For example, you can use IBM Cloud storage type `ibmc-block-gold` and `ibmc-block-custom`.

What's next

- At a minimum, you need to tell developers the size to request for sandbox storage, and what [storageClassName](#) to use.
- If developers do not have authority to create a `PersistentVolumeClaim`, you need to create a `PersistentVolumeClaim` for each sandbox instance.
- Starting from version 1.2, if the backing storage requires a specific group ID, or the default group ID 2105 conflicts with another ID already in use, you can set the group ID in the Custom Resource with the `spec.fsGroup` parameter. Note that in versions earlier than 1.2, when a sandbox instance is started, the storage is initialized with correct user and group file permissions by an [init container](#). Therefore, no extra configuration of file system permissions is required.

Configuration reference

The Sandbox Operator supports two different Custom Resource file kinds. One is WaziSandboxSystem, and the other is WaziSandboxVolumeCopy.

WaziSandboxSystem

WaziSandboxSystem controls the creation of sandbox instances, and the general format is displayed as below.

```
apiVersion: sandbox.wazi.ibm.com/v1alpha1
kind: WaziSandboxSystem
metadata:
  name: your-sandbox-name
spec:
  ...
```

Where `your-sandbox-name` is the name of the sandbox instance. It must be unique within the target namespace.

The spec describes how the sandbox instance is deployed. Many fields of the spec are typically provided by the cluster administrator since they involve resources that are controlled by the cluster administrator or are common to multiple developers and sandbox instances.

The following fields are available for the spec section in the Custom Resource file:

Path	Required/Optional	Default	Description
<code>license.accept</code>	Required		Must be set to <code>true</code> to acknowledge acceptance of the Sandbox license when you create a sandbox instance. Note: This is a new field added in Sandbox 1.2. You must add this field to any existing WaziSandboxSystem Custom Resource files that are created prior to Sandbox 1.2.
<code>licensedProduct</code>	Optional	IBM Wazi Developer	Licensed product being used with Sandbox. Select one from the following products: <ul style="list-style-type: none">• IBM Wazi Developer• IBM Z Development and Test Environment (AUSS)• IBM Z Development and Test Environment (RVU)
<code>zpdt.secretName</code>	Required		Name of the Secret

Path	Required/Optional	Default	Description
zpdt.configMap.create	Optional	false	Create a ConfigMap. If false, only the parameter name is used and other parameters are ignored.
zpdt.configMap.name	Conditional “1” on page 81		The name of the ConfigMap to use.
zpdt.configMap.licenseServer	Conditional “2” on page 81		The address of the primary License Server.
zpdt.configMap.secondaryLicenseServer	Optional “2” on page 81		The address of a secondary License Server. Use this field if you want to add more than one License Server to Sandbox.
zpdt.configMap.sftpHost	Conditional “2” on page 81		The address of the SFTP server.
zpdt.configMap.sftpPort	Conditional “2” on page 81		The port of the SFTP server.
zpdt.configMap.sftpUser	Conditional “2” on page 81		The user to log in to the SFTP server.
zpdt.configMap.sftpPath	Conditional “2” on page 81		The path to the directory on the SFTP server with the z/OS volume files.
zpdt.configMap.iplZOS	Optional “2” on page 81	true	If true, z/OS will IPL when the sandbox instance is created.
zpdt.configMap.iplAddress	Optional “2” on page 81	0A80	The IPL volume for z/OS, also known as SYSRES.
zpdt.configMap.iplIODFAddress	Optional “2” on page 81	0A82	The volume that contains the IODF.
zpdt.configMap.iplLoadParm	Optional “2” on page 81	NZ	The 2-character suffix of the LOADxx member of SYS1.IPLPARM.
zpdt.configMap.devmap	Optional “2” on page 81	devmap.txt	Path to the emulator machine characteristics and device map file, relative to the mounted volumes.
zpdt.configMap.shutdownCommand	Optional “2” on page 81	%netv shutsys	The z/OS command to initiate shutdown.
zpdt.configMap.shutdownTimeout	Optional “2” on page 81	300	The length of time, in seconds, to wait for z/OS shutdown command to complete.
persistence.useDynamicProvisioning	Optional	true	Use dynamic provision for storage.

Path	Required/Optional	Default	Description
zVolumes.existingClaimName	Optional		The name of an existing claim to use. If it is set, other zVolumes options are ignored.
zVolumes.name	Optional “3” on page 81		If existingClaimName is not set, an optional name for the claim.
zVolumes.storageClassName	Conditional “3” on page 81		If existingClaimName is not set, the storage class name provided by the cluster administrator.
zVolumes.size	Optional “3” on page 81	300 GB	The size of the claim.
zVolumes.selector.label	Optional “3” on page 81		A selector label.
zVolumes.selector.value	Optional “3” on page 81		A selector value.
copyThreads	Optional		The number of threads used to copy the volume files.
copyBlockSize	Optional		The size (in bytes) of blocks when copying.
copyVolumes	Optional	true	Attempt to copy z/OS volume files before starting sandbox instance. If you are using WaziSandboxVolumeCopy to copy files, set it to false.
fsGroup	Optional	2105	Supplemental group ID to mount z/OS volume file claim.
controller.expose	Optional	route	Choose how to expose the Instance Controller: <ul style="list-style-type: none"> • cluster: Adds it to a ClusterIP Service. • route: Adds it to a ClusterIP Service and a Route. • nodeport: Adds it to a NodePort Service with no Route.

Path	Required/Optional	Default	Description
controller.tls	Optional		If exposing a Route, values here will be used for the tls configuration of the Route.
networkPolicy	Optional	allow-all	<p>Create a NetworkPolicy for the sandbox instance:</p> <ul style="list-style-type: none"> • allow-all: Allows all traffic to the sandbox. • namespace-only: Allows traffic from within this namespace only. • deny-all: Blocks all traffic and can be used as a starting point for other policies added manually. • none: No NetworkPolicy will be created.
portProfile.profile	Optional	wazi	<p>Select a port profile to use:</p> <ul style="list-style-type: none"> • wazi: Includes encrypted ports of Wazi Developer's host components. • wazi-all: Includes encrypted and unencrypted ports of Wazi Developer's host components. • custom: Empty profile. Does not include ports of Wazi Developer's host components. All ports must be given with z/OS ports.

Path	Required/Optional	Default	Description
portProfile.scope	Optional	route	<p>Choose how to expose the ports specified in the profile. If you chose the custom profile that has no ports, ignore this setting.</p> <ul style="list-style-type: none"> • <code>cluster</code>: Exposes ports only on the OpenShift cluster using a ClusterIP Service. • <code>route</code>: Exposes ports on the OpenShift cluster. It also exposes the ports that are modern, web-based protocols outside the cluster using OpenShift Route. • <code>nodeport</code>: Exposes what route does, and everything else it exposes using a NodePort Service.
zports.route	Optional		An array, which might be empty, of ports to be exposed with a ClusterIP Service and a Route.
zports.route[].name	Required		The name of this port within the service. It must be a DNS label, and unique within the service.
zports.route[].port	Required		The z/OS port number to be exposed by the service.
zports.route[].protocol	Optional	TCP	The IP protocol for this port. One of TCP, UDP, SCTP.

Path	Required/Optional	Default	Description
<code>zports.route[].appProtocol</code>	Optional		The application protocol for this port. This field follows standard Kubernetes label syntax. Unprefixed names are reserved for IANA standard http://www.iana.org/assignments/service-names . Non-standard protocols should use prefixed names such as <code>mycompany.com/my-custom-protocol</code> .
<code>zports.route[].tls</code>	Optional		Values used for the <code>tls</code> configuration of the Route.
<code>zports.cluster</code>	Optional		An array, which might be empty, of ports to be exposed with a <code>ClusterIP</code> Service
<code>zports.cluster[].name</code>	Required		The name of this port within the service. It must be a DNS label, and unique within the service.
<code>zports.cluster[].port</code>	Required		The z/OS port number to be exposed by the service.
<code>zports.cluster[].protocol</code>	Optional	TCP	The IP protocol for this port. Options: TCP, UDP, SCTP.
<code>zports.cluster[].appProtocol</code>	Optional		The application protocol for this port. This field follows standard Kubernetes label syntax. Unprefixed names are reserved for IANA standard http://www.iana.org/assignments/service-names . Non-standard protocols should use prefixed names such as <code>mycompany.com/my-custom-protocol</code> .
<code>zports.nodeport</code>	Optional		An array, which might be empty, of ports to be exposed with a <code>NodePort</code> Service

Path	Required/Optional	Default	Description
<code>zports.nodeport[].name</code>	Required		The name of this port within the service. It must be a DNS label, and unique within the service.
<code>zports.nodeport[].port</code>	Required		The z/OS port number to be exposed by the service.
<code>zports.nodeport[].protocol</code>	Optional	TCP	The IP protocol for this port. Options: TCP, UDP, SCTP.
<code>zports.nodeport[].appProtocol</code>	Optional		The application protocol for this port. This field follows standard Kubernetes label syntax. Unprefixed names are reserved for IANA standard http://www.iana.org/assignments/service-names . Non-standard protocols should use prefixed names such as <code>mycompany.com/my-custom-protocol</code> .
<code>service</code>	Optional		This parameter was deprecated starting from Sandbox 1.4.0. Use <code>portProfile</code> and <code>zPorts</code> to control what ports to expose and how to expose them. If you have an existing <code>service</code> section in your Custom Resource, it will produce the same <code>NodePort Service</code> as in earlier releases; but if no <code>service</code> section is present, that <code>Service</code> will no longer be created.
<p>Note:</p> <ol style="list-style-type: none"> 1. If <code>zpdt.configMap.create</code> is set to <code>false</code>, this value is required. 2. If <code>zpdt.configMap.create</code> is set to <code>false</code>, this field is ignored. 3. If <code>zVolumes.existingClaimName</code> is set, this field is ignored. 			

The created Service might expose the following named ports based on the settings above:

Service port name	Port	wazi	wazi-all	Routeable
debug-manager	7397	Y	Y	
debug-profile-service	7392	Y	Y	Y
dtn-cicsts56	7350		Y	
dtn-cicsts55	7360		Y	
ftp	21		Y	
local-vtam	3270		Y	
remote-debug-service	7394	Y	Y	
rse-api	7395	Y	Y	Y
rse-d	4035	Y	Y	
rse-server	4037	Y	Y	
ssh	22	Y	Y	
tn3270-secure	2023	Y	Y	
tn3270-unsecured	23		Y	
zosconnect	9443		Y	
zosmf	10443	Y	Y	Y

WaziSandboxVolumeCopy

WaziSandboxVolumeCopy allows copying ADCD z/OS volume images into Red Hat OpenShift and within OpenShift. Then, the volume images can be used by a sandbox instance.

```
apiVersion: sandbox.wazi.ibm.com/v1alpha1
kind: WaziSandboxVolumeCopy
metadata:
  name: your-volume-copy-name
spec:
  ...
```

Where `your-volume-copy-name` is descriptive name for the copy job. It must be unique within the target namespace.

The spec describes how the copy job operates. Many fields of the spec are typically provided by the cluster administrator since they involve resources that are controlled by the cluster administrator or are common to multiple developers and sandbox instances.

The following fields are available for the spec section in the Custom Resource file.

Path	Required/Optional	Default	Description
<code>source.type</code>	Optional	<code>sftp</code>	The type of the source of volume files, one of <code>sftp</code> for copying from an SFTP server or <code>pvc</code> for copying from another PersistentVolumeClaim.

Path	Required/Optional	Default	Description
source.existingConfigName	Optional		Name of the ConfigMap that has configuration information in the form of source.config. If specified, source.config will be ignored.
source.secretName	Conditional “1” on page 86		Name of the Secret that has the SFTP password in a key named sftpPassword.
source.config.sftpHost	Conditional “2” on page 86		Address of the SFTP server.
source.config.sftpPort	Conditional “2” on page 86		Port of the SFTP server.
source.config.sftpUser	Conditional “2” on page 86		User to log in to the SFTP server.
source.config.sftpPath	Conditional “2” on page 86		Path to the directory on the SFTP server with the z/OS volume files.
source.existingClaimName	Conditional “3” on page 86		Name of the PersistentVolumeClaim that has the source volumes.
persistence.useDynamicProvisioning	Optional	true	Use dynamic provision for storage.
destination.existingClaimName	Optional		The name of an existing claim to use. If set, other destination options are ignored.
destination.existingConfigName	Optional		If destination.existingClaimName is not set, the name of the ConfigMap that has configuration information for creating a PersistentVolumeClaim, containing the keys name, size, and optionally storageClassName or selector.label and selector.value. If specified, those keys in destination will be ignored.

Path	Required/Optional	Default	Description
destination.storageClassName	Conditional		If destination.existingClaimName and destination.existingConfigName are not set, the storage class name provided by the cluster administrator.
destination.size	Optional	300Gi	If destination.existingClaimName and destination.existingConfigName are not set, the size of the claim.
destination.selector.label	Optional		If destination.existingClaimName and destination.existingConfigName are not set, an optional selector label.
destination.selector.value	Optional		If destination.existingClaimName and destination.existingConfigName are not set, an optional selector value.
copyThreads	Optional		The number of threads used to copy the volume files.
copyBlockSize	Optional		The size (in bytes) of blocks used when copying.
copyTimeout	Optional		The length of time (in seconds) that the operator will wait for the copy to complete before marking the current attempt as failed and restarting the reconciliation.
fsGroup	Optional	2105	Supplemental group ID to mount z/OS volume file claim.
manifest.digests	Optional	both	Where to calculate volume digests, one of none, source, destination, and both.

Path	Required/Optional	Default	Description
manifest.digestAlgorithm	Conditional “4” on page 86	sha256	The default algorithm for volume digests if not specified with the volume. Can be md5 or sha256.
manifest.digestValidation	Optional “5” on page 86	verify	How to validate digests. One of the following values: ignore Does no validation and ignores any digests in the manifest warn Issues warnings of missing digests or issues a non-fatal error if the digests do not match verify Causes a fatal error if any digest that exists in the manifest does not match require Causes a fatal error if a volume has no digest or a digest does not match
manifest.contentsFile	Optional “6” on page 86	inventory.txt	<ul style="list-style-type: none"> To use the text or checksum manifest type: Path from source to the contents file. To use the checksum manifest type: The file is in the format produced by the md5sum or sha256sum commands. To use the JSON manifest type: In the format of a JSON array the same as in the manifest.contents object.

Path	Required/Optional	Default	Description
manifest.contents	Conditional “7” on page 86		An array that specifies the volumes to be copied. The array elements are in YAML of objects using the manifest.contents.field entries. The array fields are referred to as manifest.contents[].field below.
manifest.contents[].path	Conditional “7” on page 86		The relative path on the source to the file.
manifest.contents[].size	Optional “8” on page 86		The size in bytes of the source file.
manifest.contents[].digest	Optional “8” on page 86		The digest of the source file in the format generated by digestAlgorithm.
manifest.contents[].destination.path	Optional “8” on page 86		The relative path on the destination to the file. If not specified the destination file will be named the same as the source file, with the .gz extension removed if the source was compressed.
manifest.contents[].destination.size	Optional “8” on page 86		The size in bytes of the destination file. Should match the source size if the source was not compressed.
manifest.contents[].destination.digest	Optional “8” on page 86		The digest of the source file in the format generated by digestAlgorithm. Should match the source digest if the source was not compressed.

Note:

1. Required for source.type: sftp.
2. Applicable for source.type: sftp. However, if source.existingConfigName is set, this field is ignored.
3. Required for source.type: pvc.
4. Required for the checksum manifest type; optional otherwise
5. Ignored if manifest.digests is none.
6. Ignored for the inline manifest type.
7. Required if you want to use the inline manifest type; ignored otherwise.
8. Optional for the inline manifest type; ignored otherwise.

To provide more control over the copy job, use the following parameters.

- affinity
- fullnameOverride
- image.pullPolicy
- image.repository
- imagePullSecrets
- nameOverride
- nodeSelector
- resources
- tolerations
- persistence.useDynamicProvisioning

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.	Image Classification	File Name
1	IBM Z Development and Test Environment Enterprise Edition 13.3.0 Installation Multilingual	M03XZML	Required	zdt-install.tgz
2	IBM Z Development and Test Environment Enterprise Edition 13.3.0 Installer checksum Multilingual	M03Y0ML	Required	ee.md5
3	RLKS V8.1.5 for Windows ML	CNIV9ML	Optional	N/A
4	LKAD V8.1.5 for Windows ML	CNIW0ML	Optional	N/A
5	RLKS 8.1.5 for Linux x86 ML	CNIW1ML	Optional	N/A
6	RLKS 8.1.5 for Solaris SPARC ML	CNIW2ML	Optional	N/A
7	RLKS V8.1.5 for Solaris x86 ML	CNIW3ML	Optional	N/A
8	RLKS V8.1.5 for HP IA64 ML	CNIW4ML	Optional	N/A
9	RLKS V8.1.5 for Linux PPC ML	CNIW5ML	Optional	N/A

	Name	Part No.	Image Classification	File Name
10	RLKS V8.1.5 for AIX® ML	CNIW6ML	Optional	N/A
11	RLKS Administrator And Reporting Tool Setup ML	CNIW7ML	Optional	N/A
12	RLKS Administrator And Reporting Tool - Part 1 ML	CNIW8ML	Optional	N/A
13	RLKS Administrator And Reporting Tool - Part 2 ML	CNIW9ML	Optional	N/A
14	ADCD z/OS V2.4 for IBM Z Development and Test Environment 13.3 RSU 2103 Multilingual eAssembly	G050QML	Optional, ADCD May 2021 Edition RSU level 2103	N/A
15	z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	M03WTML	Optional, ADCD May 2021 Edition RSU level 2103	D4RES1.ZPD
16	z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	M03WVML	Optional, ADCD May 2021 Edition RSU level 2103	D4RES2.gz
17	z/OS 2.4 Part 3 of 19 - System volume Multilingual	M03WVML	Optional, ADCD May 2021 Edition RSU level 2103	D4SYS1.gz
18	z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	M03WXML	Optional, ADCD May 2021 Edition RSU level 2103	D4CFG1.gz
19	z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	M03WYML	Optional, ADCD May 2021 Edition RSU level 2103	D4USS1.gz
20	z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	M03WZML	Optional, ADCD May 2021 Edition RSU level 2103	D4USS2.gz
21	z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	M03X0ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD1.gz
22	z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	M03X1ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD2.gz
23	z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	M03X2ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD3.gz
24	z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	M03X3ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRC4.gz
25	z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	M03X4ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS1.gz
26	z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	M03X5ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS2.gz
27	z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	M03X6ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS3.gz
28	z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M03X7ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGA.gz

	Name	Part No.	Image Classification	File Name
29	z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	M03X8ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGB.gz
30	z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	M03X9ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGC.gz
31	z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	M03XBML	Optional, ADCD May 2021 Edition RSU level 2103	D4USR1.gz
32	z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	M03XCML	Optional, ADCD May 2021 Edition RSU level 2103	SARES1.ZPD
33	z/OS 2.4 Part 19 of 19 – zCX Container volume	M03XDML	Optional, ADCD May 2021 Edition RSU level 2103	D4ZCX1.gz
34	Engineering Workflow Management 7.0.1 Multilingual	M03XFML	Optional, ADCD May 2021 Edition RSU level 2103	D4BLZ1.gz
35	DB2 V12 Part 1 of 2 Multilingual	M03XGML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBC1.gz
36	DB2 V12 Part 2 of 2 Multilingual	M03XHML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBC2.gz
37	DB2 V11 Part 1 of 2 Multilingual	M03XJML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBB1.gz
38	DB2 V11 Part 2 of 2 Multilingual	M03XKML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBB2.gz
39	CICS 5.6 Multilingual	M03XLML	Optional, ADCD May 2021 Edition RSU level 2103	D4C560.gz
40	CICS 5.5 Multilingual	M03XMM L	Optional, ADCD May 2021 Edition RSU level 2103	D4C551.gz
41	IMS 14.1 Multilingual	M03XNML	Optional, ADCD May 2021 Edition RSU level 2103	D4IME1.gz
42	IMS 15.2 Multilingual	M03XPML	Optional, ADCD May 2021 Edition RSU level 2103	D4IMF1.gz
43	IBM Installation Manager 1.9 Multilingual	M03XQML	Optional, ADCD May 2021 Edition RSU level 2103	D4INM1.gz
44	DB2 Archive Logs Multilingual	M03XRML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBAR.gz
45	Tivoli Omegamon XE products 5.5.1 Multilingual	M03XSML	Optional, ADCD May 2021 Edition RSU level 2103	D4KAN1.gz
46	WAS V9.0 Part 1 of 2 Multilingual	M03XTML	Optional, ADCD May 2021 Edition RSU level 2103	D4W901.gz
47	WAS V9.0 Part 2 of 2 Multilingual	M03XVML	Optional, ADCD May 2021 Edition RSU level 2103	D4W902.gz
48	Zowe Open Source Project Multilingual	M03XWM L	Optional, ADCD May 2021 Edition RSU level 2103	D4ZWE1.gz

	Name	Part No.	Image Classification	File Name
49	ADCD Metadata for Tools Multilingual	M03XXML	Optional, ADCD May 2021 Edition RSU level 2103	ADCDTOOLS.XML
50	Extended ADCD Checksum Multilingual	M03XYML	Optional, ADCD package MD5 checksum	may2021_adcd_md5.txt

Validating the downloaded installer file

ZD&T Enterprise Edition installer now supports the signing of the installer. With the signed installer, you can validate the completeness of the downloaded package by running a command with the signing certificates that are packaged in the installer.

ZD&T Enterprise Edition installer file `zdt-install.tgz` that you download from IBM is a compressed file. This installer file contains the following multiple installers, and corresponding signature files.

zdt-install

Install the Enterprise Edition web server, and the corresponding signature file is `zdt-install.sig`.

zdt-license-server

Install the Software-based License Server, and the corresponding signature file is `zdt-license-server.sig`.

zdt-emulator

Install the ZD&T emulator, and the corresponding signature file is `zdt-emulator.sig`.

To validate the signature file for the installer, run the following command.

```
openssl dgst -sha256 -verify <PUBLICKEYNAME> -signature <signature_file> <FILETOSIGN>
```

For example, run the following command from the directory that contains the extracted installer files.

```
openssl dgst -sha256 -verify signature/public.pem -signature zdt-install.sig zdt-install
```

If the installation file is valid, you will receive the message of `Verified OK`.

Enabling the Software-based License Server

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



Warning:

- The Software-based License Server needs to be a static resource in any infrastructure configuration. In a virtualized infrastructure, the license server cannot be moved physically. If the Software-based License Server is manually or automatically moved, you need to return the license before you move the license server. Then, you need to acquire the license, and apply the license to the license server again after the movement.

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.

Host ID and host name

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

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.

To enable a Software-based License Server, follow the checklist.

<i>Table 21. Enablement checklist for software-based licensing</i>			
Software-based license		Required/Optional	Complete
1	“Installing Software-based License Server” on page 91 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"/>

Installing Software-based License Server

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

1. Open the directory that contains the installer `zdt-install.tgz`.
2. Change the authority.

```
chmod 755 zdt-install.tgz
```

3. Decompress the installer.

```
tar -xvf zdt-install.tgz
```

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

```
./zdt-license-server
```

6. Select **1** to install.
7. Press ENTER, and read the license agreements carefully. At the end of the license, enter `y` to accept or `n` to decline the terms.
8. Read the warning message, and accept or decline by entering `y` or `n`.
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 58](#). 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 commands to verify whether the installation is successful.

- RHEL operating system

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

- UBUNTU operating system

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

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

```
uim 1.10.55.12
```

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
```

11. Configure the iptables port for the License Server by opening both UDP and TCP port 1947 that are listed in the [Table 24 on page 111](#). For more information about how to set up port 1947, see [“Firewalls” on page 264](#).

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-license-server --install
```

- To install the dependencies, run the following command.

```
zdt-license-server --install --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:

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

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

```
./uimserverstart
```

Combining all the available licenses on a single license server

If you have multiple licenses available for ZD&T, you can combine all the available licenses on a single license server.

To combine all the available licenses on a single license server, complete the following steps.

1. Select all the licenses that you have, and click **Next** at the end of the page.

IBM Rational License Key Center

Sanjay Ghanshyam
Company: RDT 2016 TEST ACCOUNT
Account ID: RDT 2016 TEST ACCOUNT

Select License Keys

Below you will find a box for each product or unique sales order that contains the item you selected on the previous page. The next screen will allow you to enter the quantity and host details for this transaction.

Item ordered:	IBM Z DEVELOPMENT AND TEST ENVIRONMENT ENTERPRISE EDITION RESOURCE VALUE UNIT
Part Number:	D1QLCLL
License Key Expiration:	permanent
Original Maintenance End Date:	Dec 31, 2017
Sales Order Number:	RDT_TEST_ACCOUNT_1
Purchase Order Number:	
Order Quantity:	100
LICENSE KEYS FOR THIS ORDER	
Select the checkbox next to the key you would like to generate or view. Help	
<input checked="" type="checkbox"/>	IBM Z Development and Test Environment Enterprise Edition Resource Value Unit License 74 Available

Item ordered:	IBM Z DEVELOPMENT AND TEST ENVIRONMENT ENTERPRISE EDITION RESOURCE VALUE UNIT
Part Number:	D1QLCLL
License Key Expiration:	permanent
Original Maintenance End Date:	Nov 5, 2017
Sales Order Number:	RDT_TEST_ACCOUNT_2
Purchase Order Number:	
Order Quantity:	50
LICENSE KEYS FOR THIS ORDER	
Select the checkbox next to the key you would like to generate or view. Help	
<input checked="" type="checkbox"/>	IBM Z Development and Test Environment Enterprise Edition Resource Value Unit License 47 Available

2. Enter the total number of emulated CPs, and click **Generate**.

Required Information

Enter required information in below fields, for help see the pop-up for each input field.

IBM Z DEVELOPMENT AND TEST ENVIRONMENT ENTERPRISE EDITION RESOURCE VALUE UNIT
Total Number of Licenses Across All Orders: 150 (121 Available)
License Expiration Date: August 22, 2021

[Request file:](#) [Number of Emulated CPs:](#)

1 Choose File sg1_fyre_ibm_...99020495.zip 100

Generate

3. Download the generated update file.

View Licenses By Host

Host Name: sg1
Host ID: 951228536548436564

Generated By : Sarjaya Ghanathay (sarjaya@ibm.com) on 21-Aug-2020

Expiration Date :Aug 22, 2021

[Download](#)

4. Apply the generated update file to your license server.

```
root@sg1:~# /opt/IBM/LDK/update_license sg1_fyre_ibm_com_1598020495_update.v2c
opened sg1_fyre_ibm_com_1598020495_update.v2c
read
The license update was successful.
There were no additional results to display.
```

5. Run the following command to check whether you have all the available licenses on your license server.

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

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 with 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 96.

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

1. Open the directory that contains the installer `zdt-install.tgz`.
2. Change the authority.

```
chmod 755 zdt-install.tgz
```

3. Decompress the installer.

```
tar -xvf zdt-install.tgz
```

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

```
./zdt-license-server
```

6. Select **2** to update.
7. Press ENTER, and read the license agreements carefully. At the end of the license, enter `y` to accept or `n` to decline the terms.
8. 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 1.10.55.12
```

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

```
./zdt-license-server --update
```

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 Software-based License Server” on page 90](#).

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 with 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 96](#).
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-license-server
```

2. Select **3** to uninstall.
3. 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-license-server --uninstall
```

Installing Enterprise Edition

ZD&T Enterprise Edition provides a web server that enable users to use the browser to extract volumes or data sets, transfer the volumes or data sets to the image storage server, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to the target environment. You can use the web server to provision instances to target environment step-by-step.

Installing the Enterprise Edition web server

To install the ZD&T Enterprise Edition web server, run the installer with the root user ID.

1. Open the directory that contains the installer `zdt-install.tgz`.
2. Change the authority.

```
chmod 755 zdt-install.tgz
```

3. Decompress the installer.

```
tar -xvf zdt-install.tgz
```

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

```
./zdt-install
```

6. Select if you want to install ZD&T Enterprise Edition by default or by your own customization.
 - Select **1** for standard installation. This option will install ZD&T Enterprise Edition with the default settings, such as installation location, ports, users and so on. Also, the Software-based License Server will be automatically installed on the same system.
 - Select **2** for customized installation, and complete the following steps:
 - a. Read the license agreements carefully, and enter `y` to confirm. At the end of the license, enter `y` to accept the terms, or enter `n` to decline the terms.
 - b. Enter `y` if you want to run the Enterprise Edition web server process with a user ID called 'zdt' that is automatically created on the system. Enter `y` to continue the process. Or, enter `n` to specify another user ID.

- c. Enter y if you want to install the ZD&T Enterprise Edition web server in the default directory `/opt/ibm`. Or, enter n, and specify another directory.
 - d. Enter y if you want to use the default port to access ZD&T Enterprise Edition web server. Or, enter n, and specify your installation port.
 - e. Enter y if you want to use the default password for the user ID to log in ZD&T Enterprise Edition web server. Or, enter y, specify and verify your password.
 - f. Enter y to install the Software-based License Server on the system. Or, enter n, and install the Software-based License Server separately by following the steps in the [“Installing Software-based License Server”](#) on page 91.
 - g. Enter y to set the ZD&T Enterprise Edition web server to start and stop automatically. Or, enter n, start and stop the ZD&T Enterprise Edition web server manually by following the description in the [“Optional: Starting the Enterprise Edition web server”](#) on page 102.
 - h. If you want to use the default passphrase for the SSH private key for logging into the Linux target, enter y. Otherwise, enter y and specify your passphrase.
7. 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 13.3.0`.

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

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 58.

- To install Enterprise Edition with the default user ID and installation path, run the following command:

```
./zdt-install --install --zdtee --zdtee_port=9443
```

Note: If you install the Enterprise Edition with the default user ID and installation path, the default user ID is 'zdt', and the default installation path is `/opt/ibm`.

- To install Enterprise Edition with your own user ID and installation path, run the following command:

```
./zdt-install --install --zdtee --zdtee_port=9443 --zdtee_user=<username> --  
zdtee_path=<installationpath>
```

Installing a signed certificate on the Enterprise Edition web server

To access to the ZD&T Enterprise Edition web server on your internal server, you need to install a signed certificate that is used by the Enterprise Edition web server.

To generate your own pkcs12 keystore (`zdtkey.p12`) that contains the certificate and put the encrypted password in the `server.env` file, follow these steps:

1. Check the installation directory of your installed ZD&T Enterprise Edition. For example, `/opt/ibm/zdt` is the default installation directory, but you can specify your own installation directory during the installation process.
2. Run the following command to generate `zdtkey.p12` and put it in the `[installation directory]/zdtServer/resources/security`.

```
openssl pkcs12 -export -out zdtkey.p12 -inkey cert.key -in cert.crt -password pass:$passcert
```

3. Modify the encrypted key store password.

- a. Get the encryption key that is specified by `wlp.password.encryption.key` in the `[installation directory]/zDTSERVER/bootstrap.properties`; for example, `/opt/ibm/zDT/zDTSERVER/bootstrap.properties`
- b. Run the following command where you installed ZD&T Enterprise Edition.

```
[installation directory]/Liberty/bin/securityUtility encode --encoding=aes --key=<encryption_key_found_above> <password value>
```

Note: To ensure that the `securityUtility` command can be run successfully, the Java path must be set up. To set up the Java path, you need to add the `JAVA_HOME` in the environment variable or add Java in the Path environment variable.

- c. Modify the `[installation directory]/Liberty/usr/servers/zDTSERVER/server.env` file with your encoded password value.

```
POSTGRES_SERVER=xxx
POSTGRES_PORT=5432
POSTGRES_NAME=xxx
POSTGRES_USER=xxx
POSTGRES_PASSWORD=xxx
POSTGRES_DRIVER_DIR=xxx

KEYSTORE_PASSWORD={aes}AG6iTGAo/v3DbfEv+7FgNH4oaoanLomL5enZr86JiS0p
```

Optional: Starting the Enterprise Edition web server

When you installed the Enterprise Edition web server and selected to start the web server manually, you need to run the script to start the Enterprise Edition web server and ensure 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.

If you need to stop Enterprise Edition web server, run the command `/opt/ibm/zDT/bin/stopServer.sh`.

Configuring the Enterprise Edition web server

Before you start your Enterprise Edition web server, ensure that you configure the authentication for Enterprise Edition.

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.

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.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.

```
<!-- user authentication -->  
<basicRegistry id="zDT" realm="zDT">  
<user name="zdtadmin"  
password="*{aes}*AM1LZsnwLRNsVtYaiwqhVD09/RL+NgYthDZXZhQgARtB*" />  
</basicRegistry>
```

For more information about authentication, see [“Optional: Starting the Enterprise Edition web server” on page 102](#).

For more information about authorization, see [“Configuring authorization” on page 127](#).

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 103](#).

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 127](#).

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 127](#).

Auditing the Enterprise Edition web server

Learn about the steps to enable audit log for ZD&T Enterprise Edition.

The audit log for the ZD&T Enterprise Edition web server is defined in the <install directory>/Liberty/usr/servers/zDTSERVER/audit.xml file.

To enable the support, follow the instruction that is stated in the audit.XML file.

For more information, see [Default Audit File Handler](#).

Installing the ZD&T z/OS Extraction Utilities

This step is required only if you want to create components from IBM Z platforms, either the physical or emulated IBM Z platforms.

The ZD&T z/OS Extraction Utilities is a component to be installed on one or more IBM Z platforms to allow ZD&T to use the IBM Z platforms as source environments to extract z/OS, CICS, and Db2 resources.

To install ZD&T z/OS Extraction Utilities, complete the following steps:

1. After the installer `zdt-install.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 `HALMD20.pdf`, and other files, including the jar file `zDTMainframeSFTP.jar`.
2. Install the IBM Z components by following the instructions in `HALMD20.pdf`, specifically in section 6.0 "Installation Instructions".

Note: The data set `hlq.IBM.HALMD20.F2` obtained at step 6.1.3 in `HALMD20.pdf` is a usable program library. If you want to install the Enterprise Edition without SMP/E, complete the following steps:

- a. Choose a value for `dsnprefix`. Then, rename `hlq.IBM.HALMD20.F2` to `dsnprefix.SFEUAUTH`, and rename `hlq.IBM.HALMD20.F3` to `dsnprefix.SFEUEXEC`.
- b. Go to step 6.1.11 by skipping steps from 6.1.4 to 6.1.10.

3. Transfer `/opt/ibm/zDT/zSystem/zDTMainframeSFTP.jar` to the USS folder `/usr/lpp/IBM/zdt` that is located at the source `z/OS`.

Optional(Legacy): Installing ZD&T emulator

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

1. Open the directory that contains the installer `zdt-install.tgz`.
2. Change the authority.

```
chmod 755 zdt-install.tgz
```

3. Decompress the installer.

```
tar -xvf zdt-install.tgz
```

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

```
./zdt-emulator
```

6. Select **1** to install.
7. Press ENTER, and read the license agreements carefully. At the end of the license, enter `y` to accept or `n` to decline the terms.
8. 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 58. 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.

9. Read the warning message, and accept or decline by entering `y` or `n`.
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
```

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

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

```
./zdt-emulator --install
```

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

```
./zdt-emulator --install --dep
```

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

```
./zdt-emulator --install --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.

Updating the Enterprise Edition web server

To update Enterprise Edition web server, you need to run the installer with the root user ID.

1. Open the directory that contains the installer `zdt-install.tgz`.
2. Change the authority.

```
chmod 755 zdt-install.tgz
```

3. Decompress the installer.

```
tar -xvf zdt-install.tgz
```

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

```
./zdt-install
```

6. If the Enterprise Edition has been updated to the latest version, you can update the Enterprise Edition with the following changes.
 - Select **1** to reinstall the Enterprise Edition with existing settings except the password of the Enterprise Edition server.
 - Select **2** to refresh the installation of the Enterprise Edition by removing all existing settings and configuring new settings.
7. If the Enterprise Edition that was installed has an older version, select **1** to update the Enterprise Edition to the latest version by following the step 6 in the [“Installing the Enterprise Edition web server” on page 100](#).
8. 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 13.3.0`.

Alternatively, after you complete steps 1 - 4, you can update ZD&T Enterprise Edition silently by using the following command.

```
./zdt-install --update --zdtee
```

Uninstalling the Enterprise Edition web server

To uninstall Enterprise Edition server, you need to return the license key, and uninstall the License Server first. Then, run the installer with the root user ID.

Before you uninstall the ZD&T Enterprise Edition, 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 96](#).
2. Stop the UIM server by using the command from a non-root user ID.

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

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

1. Open the directory that contains the installer `zdt-install.tgz`.

2. Change the authority.

```
chmod 755 zdt-install.tgz
```

3. Decompress the installer.

```
tar -xvf zdt-install.tgz
```

4. Run the installer from the root user ID.

```
./zdt-install
```

5. If you installed the Enterprise Edition server with the Software-based License Server, select if want to only uninstall ZD&T or Software-based License Server, or uninstall both of them.

6. Enter y to confirm the uninstallation. Or, enter n to cancel the uninstallation.

Alternatively, you can uninstall ZD&T Enterprise Edition silently by using the following command.

```
./zdt-install --uninstall --zdtee
```

User's Guide

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

- *Enterprise Edition User's Guide* describes all detailed steps to create components and images, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image 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.

Overview of Enterprise Edition

ZD&T Enterprise Edition provides a web user interface that can create, and monitor images. Then, you can provision a z/OS instance that is running on an emulated IBM Z hardware from the created image to a target environment more quickly with the security and control you need.

By using Enterprise Edition, you can create images from the following resources. Then, you can provision a z/OS instance that is running on an emulated IBM Z hardware from the created image to your target environment.

- IBM Z components
- Application Developers Controlled Distributions (ADCD)

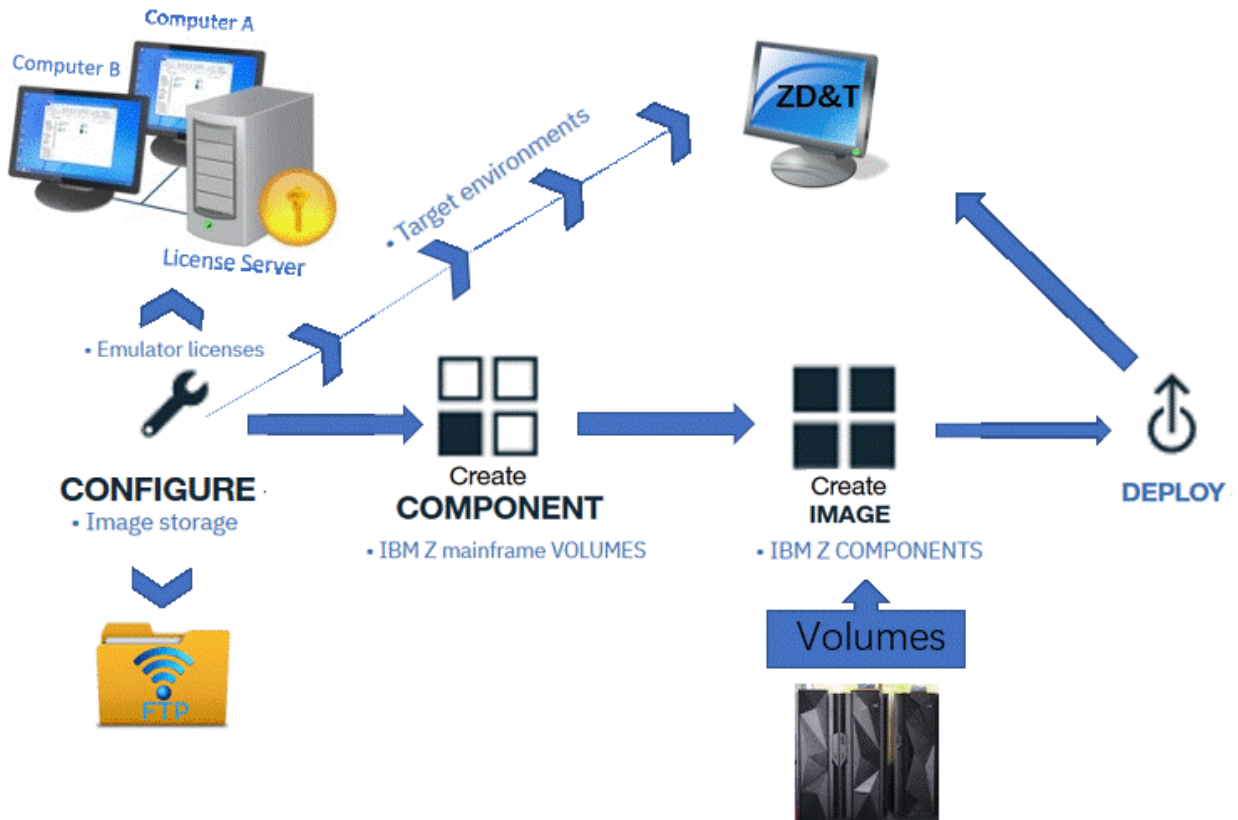
Image creation from IBM Z components

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

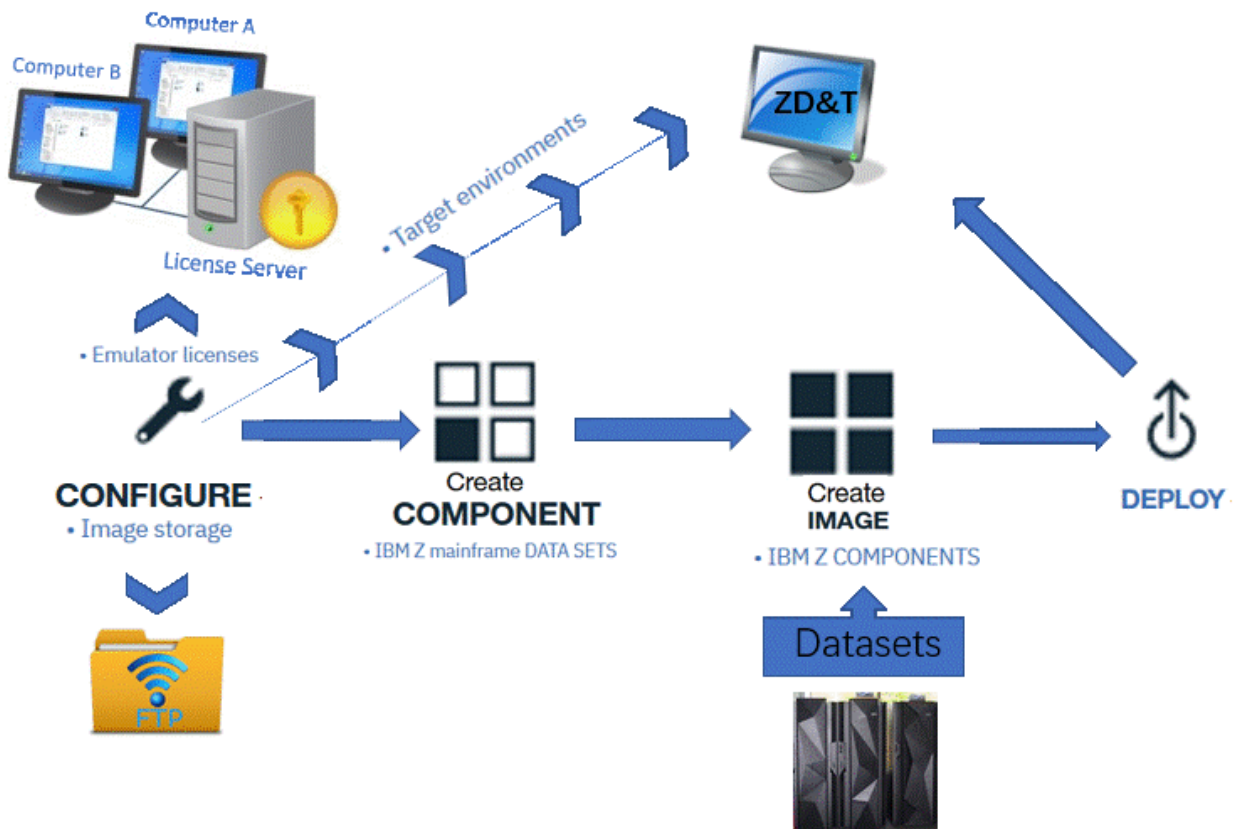
- IBM Z mainframe data sets
- IBM Z mainframe volumes
- IBM Z mainframe CICS
- IBM Z mainframe Db2
- IBM Application Discovery projects
- IBM ZD&T volumes from an existing ZD&T environment

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, you can create components by selecting the volumes from source z/OS, create the images from the created

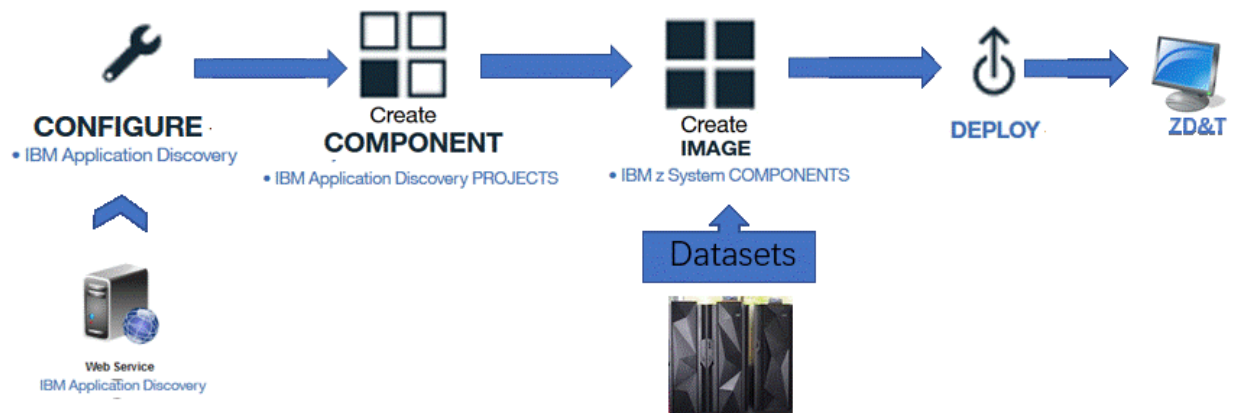
components, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to the target environment.



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, you can create components by selecting the data sets from source z/OS, create the images from the created components, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to the target environment.

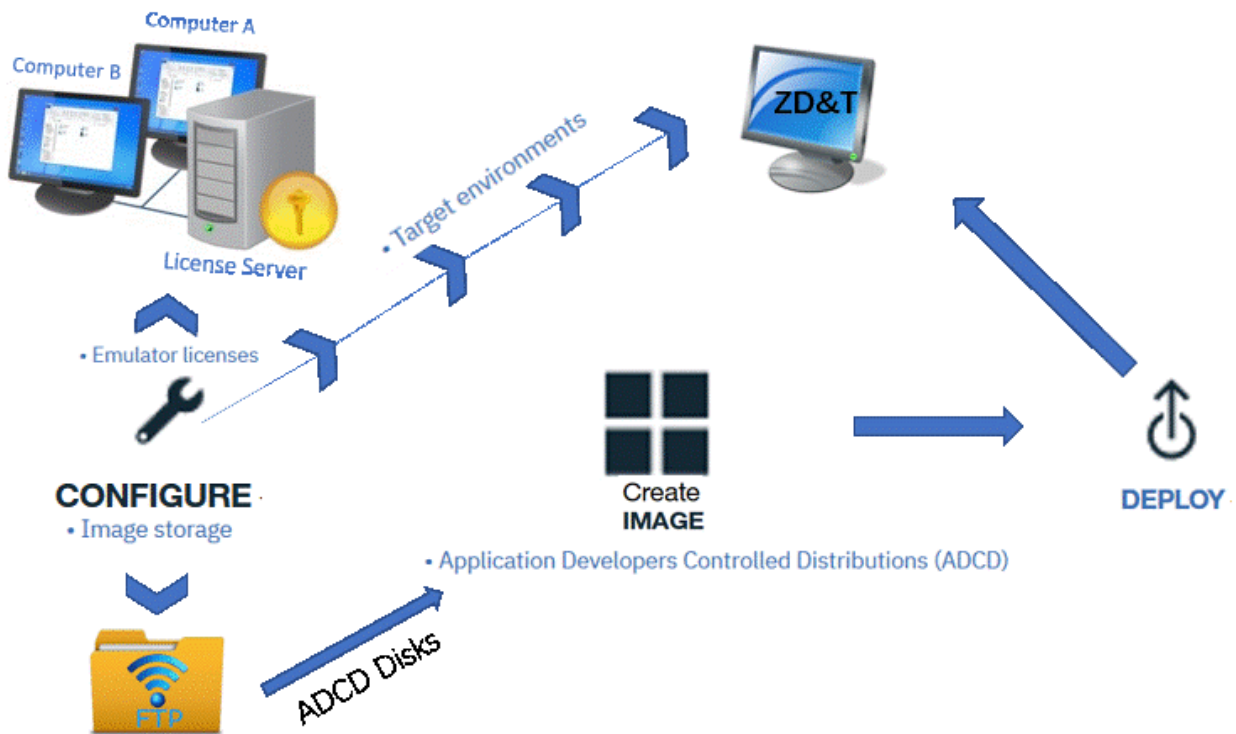


ZD&T Enterprise Edition can be integrated with IBM Application Discovery. If you have built your project in Application Discovery, and want to transfer artifacts such as data sets from source Z machine to the provision ZD&T machine, you need to configure connection with your IBM Application Discovery web services. Then, create images from the created components, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to the target environment.



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 images by selecting the subsystems, such as CICS, IMS, DB2 and so on, and the required versions of the subsystems, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to the target environment.



Cloud-ready images

A cloud-ready image is considered to be an existing image that is stored in a Persistent Volume Claim (PVC) in a Red Hat OpenShift target environment. A cloud-ready image contains only a set of volume files that can be easily cloned for use by a new ZD&T instance. To create a cloud-ready image, you must ensure that your storage class is specified to support CSI Volume Cloning when you define your Red Hat OpenShift target environment.

A cloud-ready image can be used to quickly clone a new instance to reduce the overall provisioning time. However, a cloud-ready image might cause extra costs for storing the image within the Red Hat OpenShift target environment.

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.

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

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

Table 23. Enterprise Edition web server (continued)

PORT	DIRECTION	USAGE	CONFIGURABLE
9443	BOTH	SSL	No

Table 24. 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 25. 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 provisioning 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 provisioning process starts; otherwise, an error message occurs when you provision an instance from a created image. All other ports are forwarded to the emulator Z instance.

Table 26. FTP server

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

Table 27. 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 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 images.

IBM Application Discovery (AD) integration

Before you create 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 190](#).
- 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(xy)` 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 provisioned only to ADCD-based target environments.

If the Db2 components that are created from source systems set Db2 Function Level to 504 or higher, then the Db2 components might not be able to be provisioned to the releases of ADCD that is earlier than May 2020 edition.

If the Db2 components are created by using Db2 Administration Tool, ensure that the Db2 components are created from systems that have Db2 running at function level 504 or higher in order to provision to ADCD May 2020 edition. In addition, Db2 components that are created on these systems where databases contain Materialized Query Tables might encounter errors if these tables are created in non-UTS tablespaces.

Docker container support

If you want to build your own container for ZD&T, ensure that the following requirements must be met.

- The sum of memory that is used by each 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 container. For example, rpms, Debian.
- The container must run in the Privileged mode.
- The 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 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.

If you want to use the container that is provided by ZD&T Enterprise Edition, ensure that the following requirements must be met.

- The sum of memory that is used by each container that is needed to run the base Linux system cannot exceed the Linux system memory.
- The container must run in the Privileged mode.
- The 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.

Web UI User's Guide

This document describes the steps to create components and images, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image by using the Enterprise Edition web user interface.

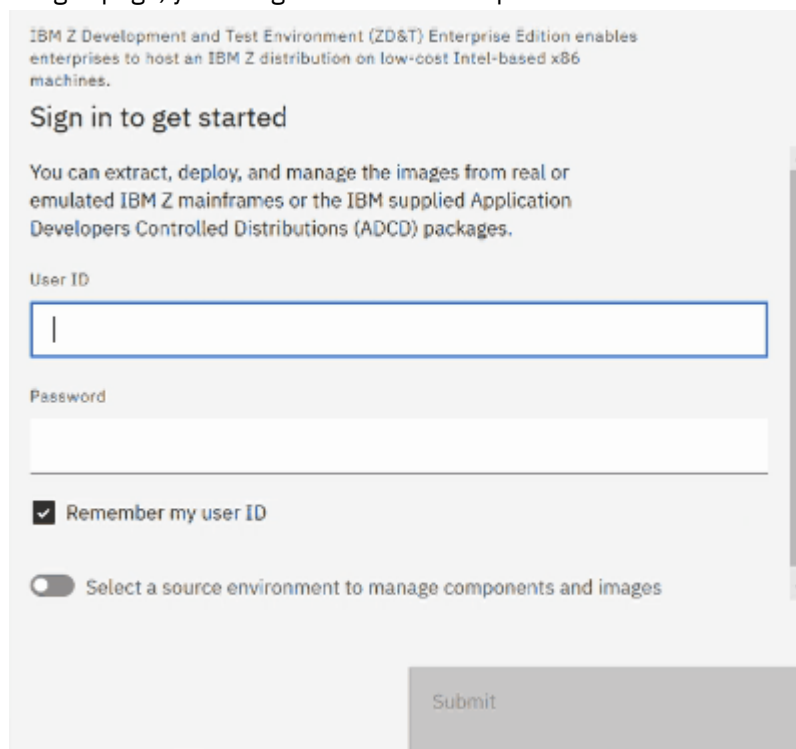
Migration to new UI from previous versions

ZD&T 13.0 provides a new UI, and the task flow is revamped in the ZD&T Enterprise Edition new UI for a streamlined and more robust user experience.

If you are using 12.0.5 or earlier versions of the ZD&T Enterprise Edition web UI, you can go through the following sections to get started with key changes.

Sign in page

If you want to manage components or images, you can additionally select an existing source environment or create a new one to work with when you sign in. Then, when you navigate to the components page and images page, you can get the list that is specific to the selected source environment right away.



IBM Z Development and Test Environment (ZD&T) Enterprise Edition enables enterprises to host an IBM Z distribution on low-cost Intel-based x86 machines.

Sign in to get started

You can extract, deploy, and manage the images from real or emulated IBM Z mainframes or the IBM supplied Application Developers Controlled Distributions (ADCD) packages.

User ID

Password

Remember my user ID

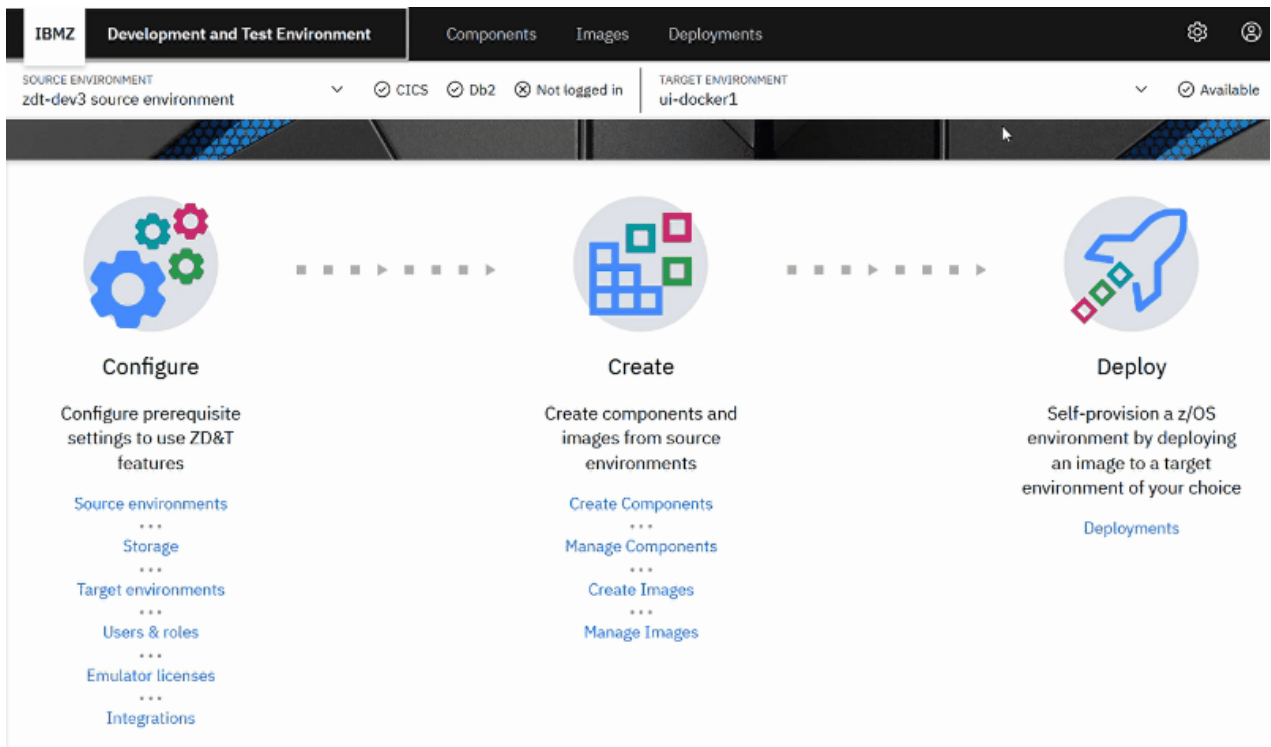
Select a source environment to manage components and images

Submit

Source and target environment selectors

You can access the environment selectors that you want from anywhere within the UI to create or manage the components, images, and instances that are specific only to the selected environment. The selectors also reduce the need to specify and authenticate at each step during the same session.

Take selecting a source environment as an example.



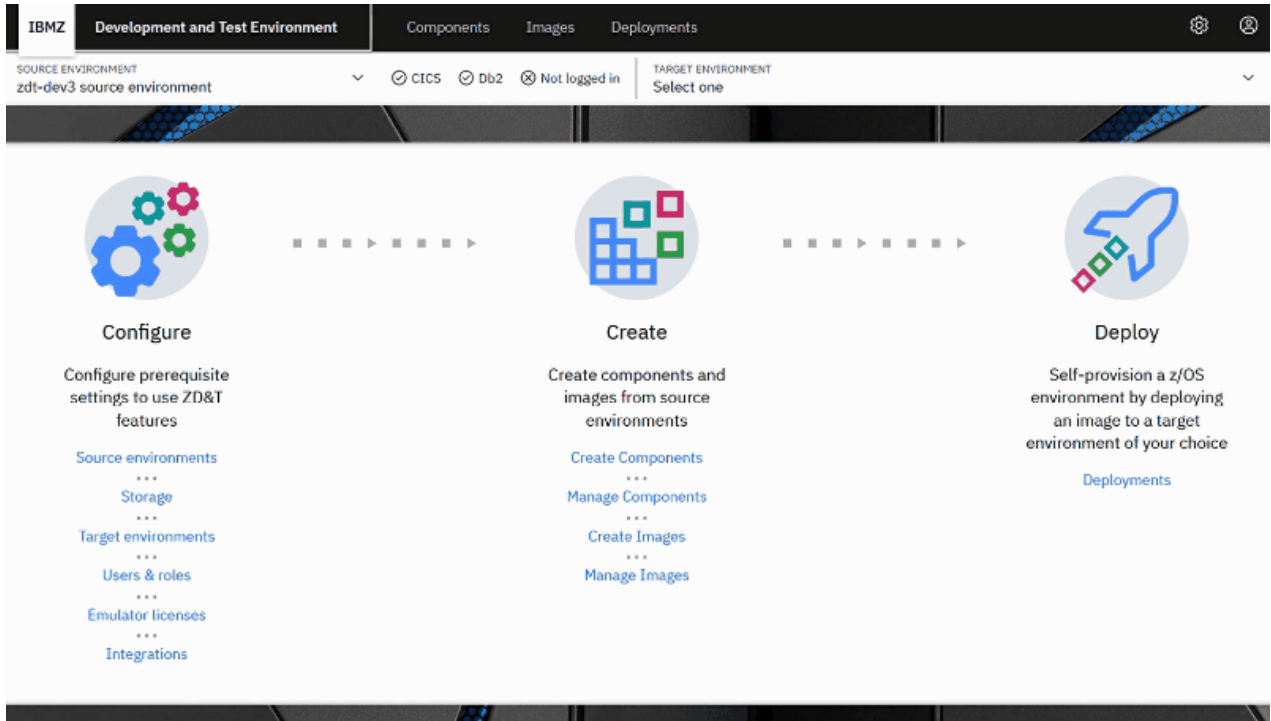
Source environment configuration

Source environment configuration now supports the following three types.

- IBM Z mainframes
- Linux system that runs an existing ZD&T emulator.
- ADCD

You can configure these types on the same source environment configuration page, and easily switch to another type of source environment. This design also consolidates the configured environments in the menu for easier retrievability.

Take configuring an IBM Z mainframe environment as an example.



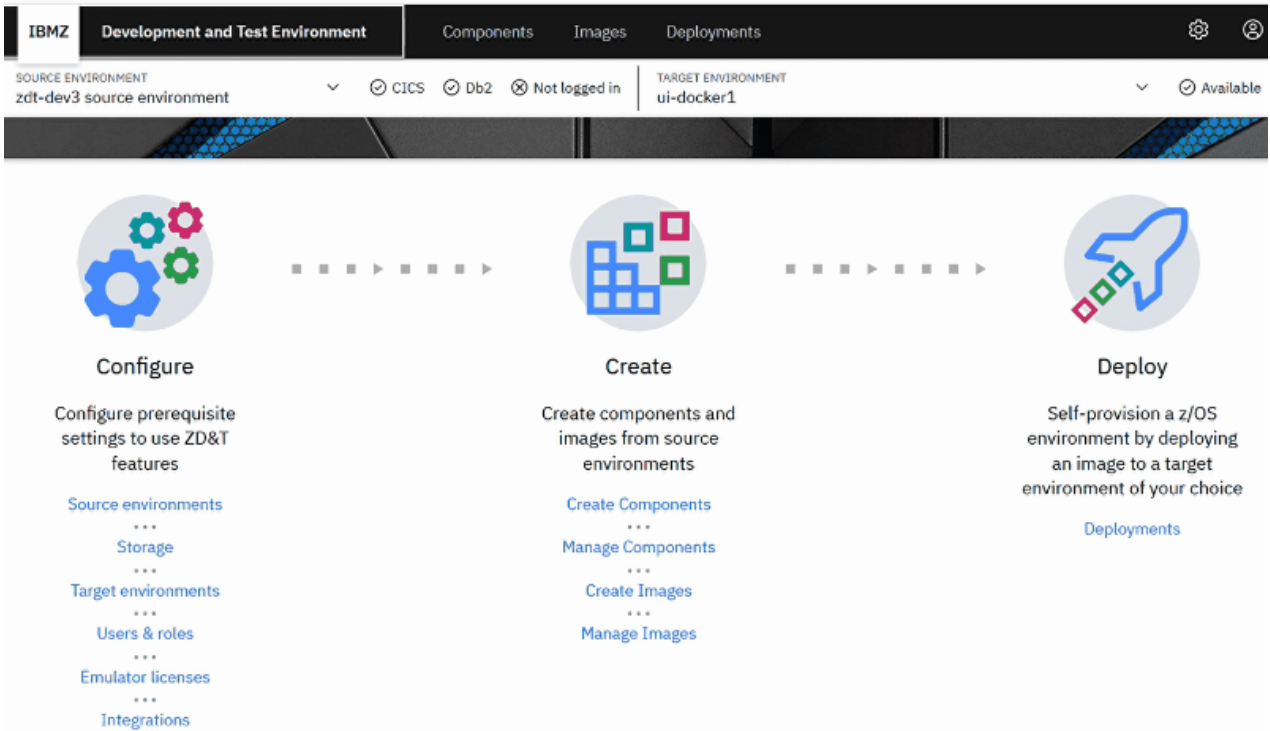
Target environment configuration

With the addition of Docker support in ZD&T 13.0.0, the target environment configuration page now provides the following three options:

-

Linux, Docker, and OpenStack.

For Linux target environments, you can now configure the Linux system as either **ZD&T managed** or **Client managed**. If you choose the **ZD&T managed** option, ZD&T configures the Linux system during instance provisioning, including installing the required Linux packages, configuring user settings, and configuring Linux network settings. Alternatively, you can choose the client managed option, and ensure that the configurations were done manually.



Component creation

The component creation for different component types is consolidated to a single page. You can now select a type of component easily from a drop-down list to create a component.

Take creating a data set environment as an example.

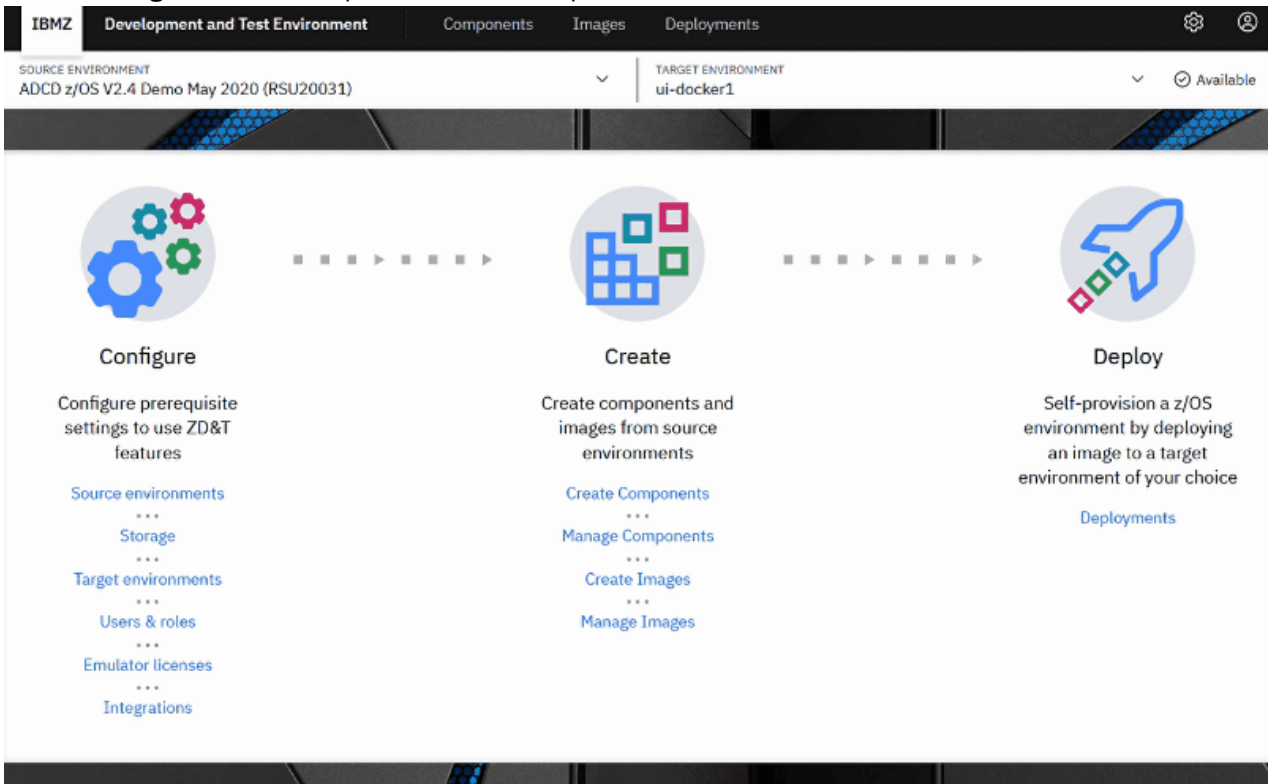
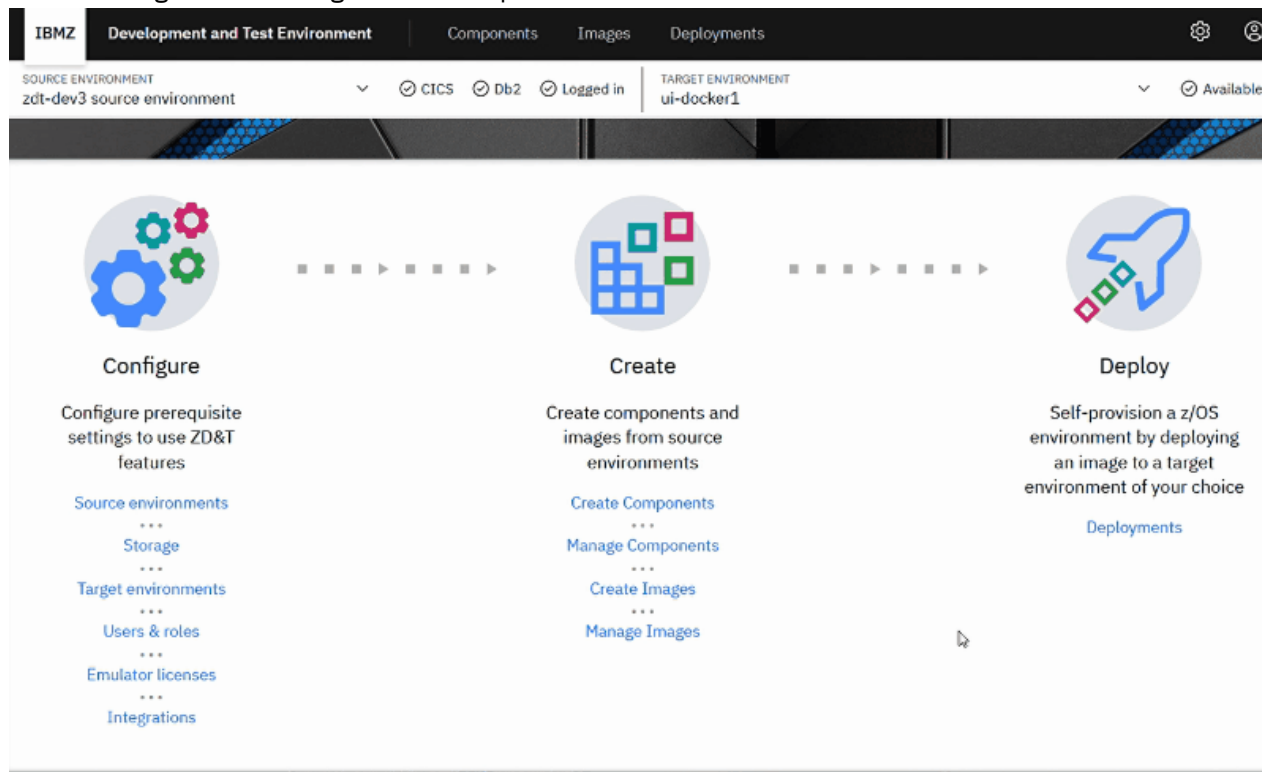


Image creation

When you create an image, the available components are displayed dynamically based on your selected source environment. For example, when you choose an ADCD package in the source environment selector, you can select all available components in the ADCD package and all extracted components from existing IBM Z mainframes. When you choose an IBM Z mainframe in the source environment selector, the page is updated in real time to display the components from the IBM Z mainframe. Then, you can include the components that are needed in your image.

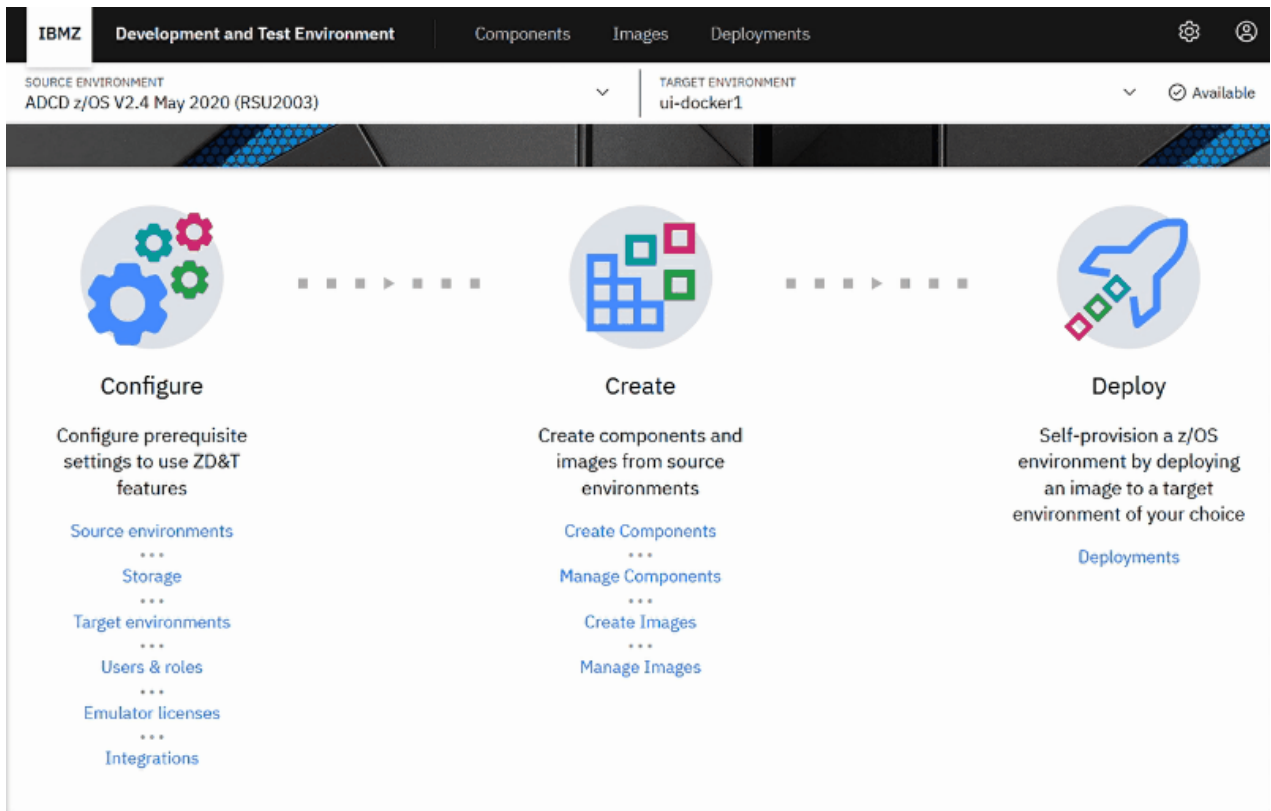
Take creating an ADCD image as an example.



Assets monitoring and management

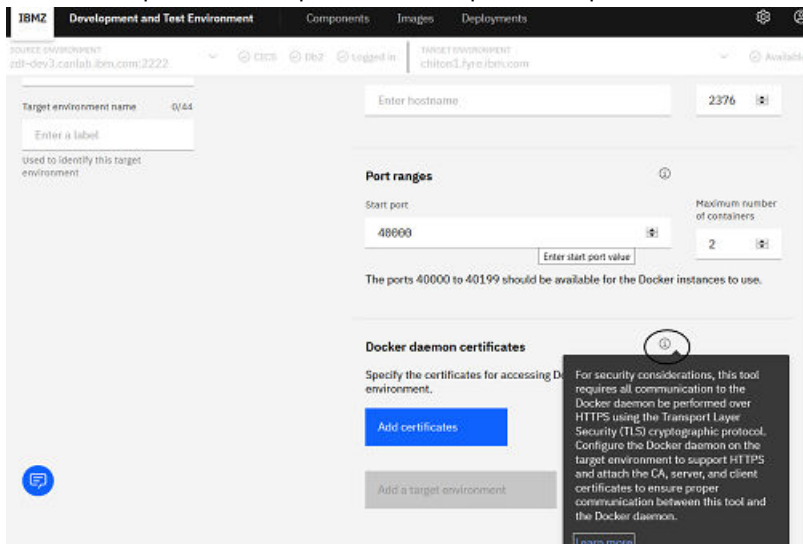
You can now easily access the created assets, including components, images, and instances that are specific to your selected source or target environments. Each type of assets now has its stand-alone page, and assets are organized for better readability and easier management.

Take monitoring and managing instances as an example.



Tooltips

ZD&T Enterprise Edition provides tooltips, which provide more context-sensitive help information.



Configuring Enterprise Edition

Learn how to configure ZD&T Enterprise Edition on the web user interface.

Configuring source environments

Source environments are configured to create components or images that are used to provision instances to target environments for development and testing.

The source environment configuration page supports the following three types:

- IBM Z mainframe system with configured CICS regions and Db2 subsystems

Supports component creation by extracting, compressing, and transferring artifacts from an existing z/OS environment on IBM Z.

- ADCD source environment

A pre-built package that contains z/OS and many z/OS products and subsystems from which you can create images.

- Linux running a ZD&T emulator

Supports faster volume component creation by compressing and transferring an existing emulated z/OS on ZD&T that runs on a host Linux system.

After you add a source environment, you can expand the source environment that is added to view the details. If you want to filter the source environment types, select the type that you want on the left.

For the IBM Z mainframe systems that are added, you can add, edit, or delete the existing source system, CICS regions, and Db2 subsystems.

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

Configuring an IBM Z mainframe

To create components from a real IBM Z environment, you need to configure an IBM Z mainframe first. Then, you can create components by extracting, compressing, and transferring artifacts from an existing z/OS environment on IBM Z.

To create a new IBM Z mainframe source environment, complete the following steps:

1. Click **Source environments** on the home page, or click the upper-right settings button to navigate to **Source environments**.
2. Click **Add source environment**.
3. Select the **IBM Z mainframe** source environment type.
4. Specify a unique source environment name that can be identified when you create components or images.
5. Enter the specified hostname and SSH port of the IBM Z mainframe system that is used to extract z/OS data.
6. If you want to create CICS components from the CICS regions on the IBM Z mainframe system, click **Add CICS region**. Then, provide the details to define a CICS region, and click **Submit**.
7. If you want to create Db2 components from the Db2 subsystems on the IBM Z mainframe system, click **Add Db2 subsystem**. Then, provide the details to define a Db2 subsystem, and click **Submit**.

Note: If you do not configure the option to use the IBM Db2 Administration Tool for z/OS, you need to enter the DDL file path when you create a Db2 component.

8. Click **Add source environment**.

For the IBM Z mainframe systems that are added, you can edit or delete the existing source system, CICS regions, and Db2 subsystems. Also, you can filter the source systems with CICS regions or Db2 subsystems by selecting the types of Db2 or CICS on the left.

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

Configuring a Linux that runs a ZD&T emulator

To create volume component quickly from an existing emulated z/OS on ZD&T that runs on a host Linux system, you need to configure an existing ZD&T source environment.

To create a ZD&T source environment, complete the following steps:

1. Click **Source environments** on the home page, or click the upper-right settings button to navigate to **Source environments**.
2. Click **Add source environment**.
3. Select the **Linux running a ZD&T emulator** source environment type.
4. Specify a unique source environment name that can be identified when you create volume components.
5. Enter the specified hostname and SSH port of the host Linux system. Then, enter the location of the ZD&T emulator devmap on the source environment.
6. If you want to define the directory where the compressed files will be stored on the ZD&T system before they are transmitted to the storage server, enable the **Compression** option. Then, enter a working directory that can temporarily store the compressed files. For example, /home/ibmsys1/zdt.
7. Enter the specified hostname and SSH port of the emulated z/OS.
8. Click **Connect ZD&T environment**.
9. Enter the credentials to the Linux that runs the ZD&T emulator, and click **Submit**.

Configuring ADCD

ADCD is a pre-built package that contains z/OS and many z/OS products and subsystems from which you can create images. ADCD is updated approximately twice a year and is provided as a part of ZD&T. To create images from ADCD, you need to configure ADCD first.

To make the ADCD provided with version 11.0.0 or later versions of ZD&T available for creating images and provisioning instances from created images, you need to copy the ADCD files to your storage by completing the following steps first.

1. Provide the required settings for the FTP storage.
2. In the base directory that is specified on the storage settings page, create a directory that is labeled `adcd`, if it does not exist.
3. In the `adcd` directory that is created in step 2, create a directory by using any label for each ADCD you want to make available in the Enterprise Edition, for example, `adcd/may2020`.
4. Transfer the `ADCDTOOLS.XML` file to the directory that is specified in step 3. The file describes the metadata of the ADCD to the product.
5. Create a `volumes` directory in the directory that is specified in step 3, for example, `adcd/may2020/volumes`.
6. Transfer all the volume files that are 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 when you create images from ADCD.

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

After the ADCD files are successfully transferred to your storage server, you need to go to the web server to configure ADCD. To create an ADCD source environment, complete the following steps:

1. Click **Source environments** on the home page, or click the upper-right settings button to navigate to **Source environments**.
2. Click **Add source environment**.
3. Select the **ADCD** source environment type.
4. Ensure that the steps 1-7 that are described on the page are completed to discover ADCD packages. Then, click **Load ADCD**.
5. After the ADCD packages are loaded successfully from the storage server, click **Back to manage page**.

If duplicated ADCD entries are found in the `adcd` directory location, you must ensure that only one copy of ADCD exists in the location.

Configuring a storage server

The storage is a crucial function of the Enterprise Edition. You can transfer and store extracted artifacts on the storage server by using SFTP (SSH File Transfer Protocol) and FTP (File Transfer Protocol).

Before you configure the storage server, 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 vsftpd.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 **Storage** on the home page, or click the upper-right settings button to navigate to **Storage**.
2. Select **SFTP** or **FTP** you need to use.
3. Enter the qualified hostname of the storage server along with the port and directory.
4. Enter the credentials of the storage server.
5. Click the **Save** button.

Configuring target environments

The target environment page is available for all users with different roles. You can provide the list of Docker, Linux, OpenStack, or Red Hat OpenShift target environments that instances will be provisioned to.

Configuring a Docker target environment

Learn about the steps to add a target environment that can run Docker to provision containers that run the ZD&T emulator.

Before you add a Docker target environment, make sure that the following prerequisite tasks were completed.

- The Docker daemon was configured for HTTPS communication by using the TLS cryptographic protocol.
- The files for the CA certificate, server certificate, and client certificate were saved, and the certificates are accessible by the browser that runs ZD&T Enterprise Edition.
- The port mappings for all the Docker containers were in place.

For more information, see [“Docker target environments requirements”](#) on page 62.

Adding a Docker target environment

To add a Docker target environment, complete the following steps:

1. Click **Target environments** on the home page, or click the upper-right settings button to navigate to **Target environments**.
2. Click **Add target environment**.
3. Click the drop-down menu of **Target environment type**, and select **Docker**.
4. Enter a unique name that can be used to identify the Docker environment.
5. In the **Service port** section, specify the hostname of the system that is running the Docker daemon and the Docker daemon HTTPS port that is set in the [Configure the Docker daemon for HTTPS communication](#). The default port for the Docker daemon that is configured for HTTPS by using TLS is 2376.
6. In the **Port ranges** section, specify the number of maximum containers that the Enterprise Edition allows provisioning and the lower port value of the contiguous port range that is available for all containers to use. Custom port mappings can be defined in the process of provisioning.

7. Click **Add certificates**, and upload the generated certificate files, including the files for CA certificate, client certificate, and server certificate. Then, click **Next**.
8. Click **Add a target environment** to save the target environment to the Enterprise Edition repository.

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 home page, or click the upper-right settings button to navigate to **Target environments**.
2. Click **Add target environment**.
3. Click the drop-down menu of **Target environment type**, and select **Linux**.
4. Enter a unique name that can be used to identify the Linux environment.
5. Select the emulator type to be **ZD&T managed** or **Client managed**. If you select **ZD&T managed**, ZD&T Enterprise Edition will automatically configure the advanced settings during the provisioning process. If you select **Client managed**, you must ensure that the following prerequisites were completed before you create a provisioning.
 - a. The ZD&T emulator is installed and configured to work with your license server.
 - b. All the required Linux operating system dependencies are installed. For more information, see [Access to software repository to run YUM or apt - get commands](#).
 - c. The user that performs the action and runs the ZD&T emulator need to have the proper permissions and privileges. For more information, see [Users and group settings](#).
 - d. The Linux networking is configured to properly route to the target emulated z/OS. Linux port 2022 is required to route to port 22 on the internal IP address, which is the value set in the target environment settings. For more information, see [Network configuration](#).
6. Enter the qualified hostname, and Secure Socket Shell (SSH) port number of the Linux environment.
7. Add the Linux target environment in one of the following ways:
 - Clear the **Advanced settings** option, and click **Add a target environment** to save the target environment to the Enterprise Edition repository.
 - Enable the **Advanced settings** option. Then, complete the following optional steps:
 - a. Modify the number of volumes that you want to extract and transfer simultaneously.
 - b. Select if you have the root access to the Linux target environment. If you select **No**, you need to 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](#).
 - c. Select if you want to install the required Linux packages during the provisioning.

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 Linux user settings during the provisioning.

If you clear the option, make sure that you manually configure the users and group settings before you provision instances from created images. For more information, see [Users and group settings](#).
 - e. Specify the emulated z/OS IP address.
 - f. Select if you want to 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 provision instances from created images. For more information, see [Network configuration](#).

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

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 home page, or click the upper-right settings button to navigate to **Target environments**.
2. Click **Add a target environment**.
3. Click the drop-down menu of **Target environment type**, and select **OpenStack**.
4. Enter a unique name that can be used to identify the cloud environment.
5. Select an authentication scope, and enter the credentials of user ID and password.
6. Enter the keystone URL, domain name, project name. Then, click **Establish connection**.
7. Select an availability zone from the drop-down list to provision an instance on the cloud platform.
8. 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.

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

9. Select a network from the drop-down list. Check the box of **Requires floating IP** if you want, and select a floating IP pool from the drop-down list. Then, specify the name of the network interface that will be used for routing the emulated z/OS network traffic.
10. Add the target environment in one of the following ways.
 - Clear **Advanced settings** option, and click **Add a target environment** to save the target environment to the Enterprise Edition repository.
 - Enable **Advanced settings** option, and complete the following optional steps:
 - a. Modify the number of volumes that you want to extract and transfer simultaneously.
 - b. Select if a snapshot of an image volume is allowed, which can significantly improve the efficiency of subsequent instances.
 - c. Select if you have the root access to the target environment. If you select **No**, you need to 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](#).

- d. Select if you want to install the required Linux packages during the provisioning.

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 Linux user settings during the provisioning.

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

- f. Specify the emulated z/OS IP address.
- g. Select if you want to 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 provision instances from created images. For more information, see [Network configuration](#).

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

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

Configuring a Red Hat OpenShift target environment

After you configure a Red Hat OpenShift target environment on the Enterprise Edition web user interface, you can provision a z/OS instance that is running on an emulated IBM Z hardware to a specific Red Hat OpenShift target environment. Also, you can use a cloud-ready image to create an instance in the Red Hat OpenShift environment.

To configure a Red Hat OpenShift target environment, complete the following steps:

1. Click **Target environments** on the home page, or click the upper-right **Configure settings** button to navigate to **Target environments**.
2. Click **Add target environment**.
3. Click the drop-down menu of **Target environment type**, and select **Red Hat OpenShift**.
4. Enter a unique name that can be used to identify the Red Hat OpenShift target environment.
5. Enter the API server URI that you obtained.

To get the API server URI, open the Red Hat OpenShift CLI, and run the `oc whoami--show-server` command. Or, you can contact your Red Hat OpenShift administrator for this information if needed.

6. Enter the API token that you obtained.

To find the API token, log in to the Red Hat OpenShift Container Platform web console, click your ID on the upper right, and select **Copy Login Command**. Or, open the Red Hat OpenShift CLI, run the `oc whoami--show-token` command. Also, you can contact your Red Hat OpenShift administrator for this information if needed.

7. Click **Establish connection**.

Note: After the connection is successful, you can edit the existing API server URI or the API token if you need. However, if you confirm to edit the values, all other selections that you made will be discarded.

8. Click the drop-down menu, and select a namespace that you want to work with. If you do not know which namespace to select, confirm with your Red Hat OpenShift administrator.
Note: To provision z/OS instances to your Red Hat OpenShift target environment successfully, you must ensure that the Sandbox Operator was installed in the target namespace.
9. Click the drop-down menu, and select a storage class that you want to work with.
10. If the selected storage class supports Container Storage Interface (CSI) volume cloning, you can enable the option **Enable fast provisioning** to store the cloud-ready images or faster provisioning.
11. Add the Red Hat OpenShift target environment in one of the following ways:
 - Clear the **Advanced settings** option, and click **Add a target environment** to save the target environment to the Enterprise Edition repository.
 - Enable the **Advanced settings** option, and modify the number of volumes that you want to extract and transfer simultaneously. Then, click **Add a target environment** to save the target environment to the Enterprise Edition repository.

Configuring authorization

The **Users&roles** page requires administrator privileges. To configure the **Users & roles** page, you need to configure the administrators first before you configure the users.

Configure a user registry

To add a user to the ZD&T Enterprise Edition, you must ensure that the user ID exists in your connected user registry.

For more information about configuring your user registry, see [“Authentication for the Enterprise Edition application” on page 103](#).

Add a user

To add a new who exists in the user registry to the ZD&T Enterprise Edition, complete the following steps:

1. Click **Users & roles** on the home page, or click the upper-right settings button to navigate to **Users & roles**.
2. Click **Add user**.
3. Enter a user ID that exists in the user registry, and select the corresponding roles for the user.
 - If you want to extract your own z/OS volumes for use on ZD&T, select the **Builder** role. Builder role includes the capabilities of developer/tester.
 - If you want to create your own z/OS image or provision a z/OS image to a target environment, select the **Developer/Tester** role.
4. Click **Save**.

Configuring emulator licenses

The license servers that are used by each target ZD&T instance needs to be configured within the Enterprise Edition before the target instances are created. The emulator licenses page also requires administrator privileges.

Configure a License Server

To configure a Hardware-based License Server or a Software-based License Server, complete the following steps:

1. Click **Emulator licenses** on the home page, or click the upper-right settings button to navigate to **Emulator licenses**.
2. Click the blank field under **License servers**. Then, specify a unique license label that can be used to recognize the license server, and enter the hostname of the license server.

3. Select the type of the license server.
4. Optional: Select if you want to set the license server as the primary license server.
5. Click **Save**.

Configure Rational license key servers

If you want to use Rational Tokens, you must add at least one license server before you can add a Rational License Key Server. To configure a Rational license key server, complete the following steps:

1. Click **Emulator licenses** on the home page, or click the upper-right settings button to navigate to **Emulator licenses**.
2. Click the blank field under **Rational license key servers**. Then, specify a unique license label that can be used to recognize the Rational license key server.
3. Enter the hostname and port of the Rational license key server.
4. Click **Save**.

After you add an emulator license, you can also edit or delete the license at any time.

Configuring AD integrations

To make ZD&T Enterprise Edition use the knowledge base to locate the data sets of a specific application, configure AD integrations by providing the details of your IBM Application Discovery web server. The AD integrations configuration also requires administrator privileges.

To create the image from IBM Application Discovery project, you need to set up 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.

Hostname

The hostname 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 ID and Password

If your HTTP or HTTPS server requires basic authentication, you must enter the password; otherwise, an "Invalid credential" error message might 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 might occur.

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

1. Click **Integrations** on the home page, or click the upper-right settings button to navigate to **Integrations**.
2. Click **Define AD server**.
3. Select the protocol type of the AD server. Then, enter the hostname and port of the AD server.
4. Enter the user ID and password to connect to the AD server.
5. 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 189 for reference.

Related information

[IBM AD Build User Guide](#)

Creating components from an existing IBM Z mainframe

A component is a collection of assets from an IBM Z mainframe. A component can be reusable with other components from the same IBM Z mainframe in a mix and match fashion. You can use the created components to create images, and provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to target environments.

ZD&T Enterprise Edition supports creating six types of components:

- Volume component
- Data set component
- CICS component
- Db2 component
- IBM Application Discovery projects
- IBM ZD&T volumes from an existing ZD&T instance

Restriction: TCP/IP profile data sets that are sequential or PDSE are not supported. Only PDS is supported now.

Creating a data set component

Learn how to create a data set component.

To create a data set component, 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 data set component, 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 a data set component, 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 a data set component, complete the following steps:

1. Select an IBM Z mainframe that you want to connect from the drop-down menu in the **SOURCE ENVIRONMENT** pane.

2. Click **Create components** on the home page. Or, click the **Components** tab on the header, and click **Create component**.
3. Enter the credentials of the IBM Z mainframe if the source environment is not logged in.
Note: Contact your administrator if you do not have the information of the mainframe system.
4. Click the drop-down menu of the **Component type**, and select **Data set**.
5. Enter a unique name that can be used to recognize the data set component.
6. Optional: Enable the **Adjust advanced settings** option, 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 components with multiple data sets. However, if the number is too large, the system might be overloaded, and the overall process might be slow.
7. Optional: Enter some notes that can help you identify the component.
8. Select the data set type that you want to create under **Filter**.
9. Enter a qualifier, for example, ADCD . ** . Then, select all the data sets that are needed for the component.
10. Click **Show number items selected** to verify the data sets that are selected to create the component.
11. Click **Create component** directly. Or, click **Schedule extraction** to specify the date and time that you want to create the component.

When you create a component, the information about IBM Z mainframe and the selected data sets will be saved and extracted to the storage server. Then, the created component can be used to create an image.

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

Creating a volume component

Learn how to create a volume component.

If you create a volume component, 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 a volume component, complete the following steps:

1. Select an IBM Z mainframe that you want to connect from the drop-down menu in the **SOURCE ENVIRONMENT** pane.
2. Click **Create components** on the home page. Or, click the **Components** tab on the header, and click **Create component**.
3. Enter the credentials of the IBM Z mainframe if the source environment is not logged in.
Note: Contact your administrator if you do not have the information of the mainframe system.
4. Click the drop-down menu of the **Component type**, and select **Volume**.
5. Enter a unique name that can be used to recognize the volume component.
6. Optional: Select if you want the system residence files to be contained in the component.
7. Optional: Enable the **Adjust advanced settings** option, 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 components with multiple volumes. However, if the number is too large, the system might be overloaded, and the overall process might be slow.
8. Optional: Enter some notes that can help you identify the component.
9. Select all volumes that are needed to create the component.
10. Click **Show number items selected** to verify the volumes that are selected to create the component.

11. Click **Create component** directly. Or, click **Schedule extraction** to specify the date and time that you want to create your component.

When you create a component, all extracted artifacts will be stored on the system that you configured for storage.

Creating a CICS component

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

To create a CICS component from an IBM Z mainframe, complete the following steps:

1. Select an IBM Z mainframe that you want to connect from the drop-down menu in the **SOURCE ENVIRONMENT** pane.
2. Click **Create components** on the home page. Or, click the **Components** tab on the header, and click **Create component**.
3. Enter the credentials of the IBM Z mainframe if the source environment is not logged in.
Note: Contact your administrator if you do not have the information of the mainframe system.
4. Click the drop-down menu of the **Component type**, and select **CICS**.
5. Select a configured CICS region.
6. Enter a unique name that can be used to recognize the CICS component.
7. Optional: Enable the **Adjust advanced settings** option, 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 components with multiple artifacts. However, if the number is too large, the system might be overloaded, and the overall process might be slow.

8. Optional: Enter some notes that can help you identify the component.
9. Select the CICS groups that are needed to create the component. If you want to specify the load modules of the selected CICS groups, enable the option, and enter a qualifier. Then, select the data sets that need to be included in the component.
10. Click **Show number items selected** to verify the CICS groups that are selected to create the component.
11. Click **Create component** directly. Or, click **Schedule extraction** to specify the date and time that you want to create the component.

After you create a CICS component from IBM Z mainframe, all extracted artifacts will be stored on the system that you configured for storage.

Creating a Db2 component

After you configured an IBM Z mainframe system with Db2 subsystems, you can create Db2 components.

Before you create Db2 components from an IBM Z mainframe, you must ensure that you are aware of the prerequisites and known limitations for creating Db2 components. For more information, see [“Prerequisites” on page 58](#) and [“Limitations” on page 112](#).

To create a Db2 component from an IBM Z mainframe, complete the following steps:

1. Select an IBM Z mainframe that you want to connect from the drop-down menu in the **SOURCE ENVIRONMENT** pane.
2. Click **Create components** on the home page. Or, click the **Components** tab on the header, and click **Create component**.
3. Enter the credentials of the IBM Z mainframe if the source environment is not logged in.
Note: Contact your administrator if you do not have the information of the mainframe system.
4. Click the drop-down menu of the **Component type**, and select **Db2**.
5. Select a configured Db2 subsystem.

6. Enter a unique name that can be used to recognize the Db2 component.
7. 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 DDL file path.
8. Optional: Enable the **Adjust advanced settings** option, 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 components with multiple artifacts. However, if the number is too large, the system might be overloaded, and the overall process might be slow.
9. Optional: Enter some notes that can help you identify the component.
10. Select the Db2 artifacts from the schemas that are displayed for the component.

Note: If you check the box of a schema, all the artifacts are automatically selected. If you want to select only some artifacts of a schema or clear some selected artifacts, click the schema name, then select or clear the artifacts.
11. Click **Show number items selected** to verify the Db2 schemas and artifacts that are selected to create the component.
12. Click **Create component**. Or click **Schedule 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 a Db2 component from IBM Z mainframe, all extracted artifacts will be stored on the system that you configured for 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 instances 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 an AD component

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

Before you create 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 components from IBM Application Discovery projects. For more information, see [“Limitations” on page 112](#).

- The connection is set up and tested through the **Integrations** page. For more information, see [“Configuring AD integrations” on page 128](#).

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

1. Click **Create components** on the home page. Or, click the **Components** tab on the header, and click **Create component**.
2. Click the drop-down menu of the **Component type**, and select **AD project**.
3. Select an IBM AD project that is configured in the AD environment integrations.
4. Enter a unique name that can be used to recognize the AD component.
5. Optional: Enable the **Adjust advanced settings** option, 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 components with multiple artifacts. However, if the number is too large, the system might be overloaded, and the overall process might be slow.

6. Optional: Enter some notes that can help you identify the component.
7. Select the data sets you need to transfer from your IBM Z machine to the target ZD&T machine. Alternatively, you can search the terms that 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 might occur when you provision instances from created images.

8. Click **Show number items selected** to verify the data sets that are selected to create the component.
9. Click **Create component** directly. Or, click **Schedule extraction** to specify the date and time that you want to create the component.

When you create components from IBM Application Discovery projects, if errors occur, see [“Troubleshooting the components creation from IBM Application Discovery projects” on page 189](#) for reference.

Creating a volume component from an existing ZD&T environment

After you configure an existing emulated z/OS on ZD&T that runs on a host Linux system, you can create a volume component from the source environment with faster speed.

Before you create a volume component from an existing ZD&T environment, ensure that the following two requirements are met.

- The emulated z/OS on the source ZD&T environment is up and running.
- The latest ZD&T z/OS Extraction Utilities is installed in the source environment. For the detailed instructions, see [“Installing the ZD&T z/OS Extraction Utilities” on page 104](#).

To create a volume component from an existing emulated z/OS on ZD&T that runs on a host Linux system, complete the following steps.

1. Select a ZD&T environment that you want to connect from the drop-down menu in the **SOURCE ENVIRONMENT** pane.
2. Click **Create components** on the home page. Or, click the **Components** tab on the header, and click **Create component**.
3. Enter the credentials of the z/OS that is running on the ZD&T.

Note: Contact your administrator if you do not have the information.
4. Enter a unique name that can be used to recognize the volume component.
5. Optional: Select if you want the system residence files to be contained in the component.
6. Optional: Enable the **Adjust advanced settings** option, and enter an integer in the box of **Concurrent extractions** to define the number of volumes that you want to copy simultaneously.

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

7. Optional: Enter some notes that can help you identify the component.
8. Select the volumes that are needed to create your component.
9. Click **Show number items selected** to verify the volumes that are selected to create the component.
10. Click **Create component** directly. Or, click **Schedule extraction** to specify the date and time that you want to create your component.

When you create a component, all the volumes that are copied will be stored on the system that you configured for storage.

11. Enter the credentials to log in the Linux that runs the ZD&T emulator.

Creating images

An image is a collection of components that are packaged together as a single deliverable. An image contains the z/OS core system and user-selected z/OS subsystems and applications. An image must contain one and only one 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. After you create an image, you can provision a z/OS instance that is running on an emulated IBM Z hardware from a created image.

To create an image from a different IBM Z mainframe, or from a different ADCD, change your source environment in the source environment drop-down list. You can use ZD&T Enterprise Edition to create images from various sources.

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



Warning: Since ZD&T 13.0, the function of importing images from an existing ZD&T is replaced by creating a volume component from an existing ZD&T. Any image that was imported from an existing ZD&T in previous versions is deprecated. You can replace the deprecated images by creating images from ZD&T components. For more information about creating ZD&T components, see [“Creating a volume component from an existing ZD&T environment” on page 133.](#)

Creating z/OS images

After you created components from IBM Z mainframe system or ZD&T instance, you can create z/OS images by combining the created components into a single package.

To create images, complete the following steps:

1. Select an IBM Z mainframe or a ZD&T instance that you want to connect from the drop-down menu in the **SOURCE ENVIRONMENT** pane.
2. Click **Create Images** on the home page. Or, click the **Images** tab on the header, and click **Create image**.
3. Enter a unique name that can be used to recognize the z/OS image.
4. Optional: Enable the **Adjust advanced settings** option, and enter an integer in the box 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](#).

5. Optional: Enter some notes that can help you identify the image.
6. Select the components and versions that you want to include in the image.

Note: An image must contain at least one and only one 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.

7. Click **Show number items selected** to verify the components that are selected to create your image. Then, click **Create image**.

Creating ADCD images

Learn how to create images from ADCD packages.

ZD&T Enterprise Edition supports creating an image by selecting the existing ADCD (Application Developers Controlled Distribution) components. You can create an image and select the pre-packaged components that contain IBM z/OS software, such as CICS 5.2 or 5.3, Db2 V11 or V12, and the required z/OS 2.4 components.

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

1. Select an ADCD source environment that from the drop-down menu in the **SOURCE ENVIRONMENT** pane.
2. Click **Create Images** on the home page. Or, click the **Images** tab on the header, and click **Create image**.
3. Enter a unique name that can be used to recognize the ADCD image.
4. Optional: Select if you want to include IBM Z components in the ADCD image.
5. Optional: Enable the **Adjust advanced settings** option, and enter an integer in the box of **Number of cryptographic coprocessor** and **Domains**.

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](#).

6. Optional: Enter some notes that can help you identify the image.
7. Select all the components from ADCD and the extracted IBM Z components that you want to include in the ADCD 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.


8. Click **Show number items selected** to verify the components that are selected to create the image. Then, click **Create image**.

Converting an image to a cloud-ready image

After you create an image, you can convert the image to a cloud-ready image if the Container Storage Interface (CSI) volume cloning is enabled in your Red Hat OpenShift target environment. Then, the cloud-ready image can be reused for faster subsequent provisioning to this target environment.

To convert an image to a cloud-ready image, complete the following steps:

1. From the drop-down menu in the **SOURCE ENVIRONMENT** pane, select an IBM Z mainframe or a ZD&T instance that you want to connect.

2. From the drop-down menu in the **TARGET ENVIRONMENT** pane, select a Red Hat OpenShift target environment.
3. Click the **Images** tab on the header.
4. From the **Image (from source environment)** section, select an image that you want to convert.
5. Click the **Convert to a cloud-ready image** () button.
6. Enter a unique name that can be used to recognize the cloud-ready image.
7. Select if you want to change the image that is used to convert. If the image has multiple versions, then select a version.
8. Optional: Enter some notes that can help you identify the image.
9. Enter the credentials that are used to log in to the final emulated IBM Z environment.

Note: If you provision an instance from a z/OS image that requires the credentials, you must enter the initial password of the source environment. However, if you provision an instance from an ADCD image that requires the credentials, you can set the initial password for the user ID 'IBMUSER'. This password will be used when you log in to the final emulated z/OS system.

10. Configure the provisioning in one of the following ways.

- Clear the **Advanced settings** option.
- Enable the **Advanced settings** option. Then, complete the following optional steps:
 - a. Define the central processors (CPs), integrated information processors (zIIPs), and available system memory size that are allocated to the emulator.

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 that the emulator performs well, the total quantity of CPs and zIIPs need to 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 volume device address, and LOADxx suffix.
- c. Specify the port of the service that is running on the z/OS that you want to expose.

11. Click **Convert image**.

Provisioning instances to target environments

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

The provisioning 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 image.

Note: To ensure that a provisioning is shown as success after IPL is complete, SSHD must be running. The provisioning operation for ZD&T Enterprise Edition relies on being able to SSH to the target z/OS to determine when the IPL is complete. It is also used as a means of being able to initiate data set restorage after IPL. If SSHD is not started with the IPL process, the Enterprise Edition web server will time out after ten minutes because a successful SSH connection to the target z/OS is not established. Even the IPL can be completed successfully, the web server determines that the IPL fails, and shows that the provisioning operation has failed.

Pre-provisioning validation

Before you provision instances from images to a Linux target environment, you must ensure that the following items are valid. Otherwise, when you provision instances, the pre-provisioning validation function will automatically check the items, and some error messages might occur.

Note: The pre-provisioning 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 no 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 or 9451.
- If you use a Hardware-based License Server, the port number must be 9450 or 9451.

6. Storage server connectivity

- If you use FTP as the storage server, the port number must be 21.
- If you use SFTP as the 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-provisioning validation function will check the users and group configurations by using `zdt_congfig_user_byRoot.sh`.

- Group

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

- Users

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

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`.

- RHEL 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 provision instances from created images successfully, a full application size and an extra space of the maximum decompressed volume size are needed.

11. Privilege commands permission

If you use a non-root user ID to provision instances from created 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 provision instances from the created images, if you clear the option to issue the IPL command manually, complete the following step after the provisioning.

1. Click **Manage instances** on the home page. Or, click the **Instances** tab on the header. Then, expand the image that is provisioned to the system.
2. Find the section that is called Initial Program Load under the 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 provisioning.

Note: If you choose a Linux target environment, and provision instances from images by using the root user ID, login with the user ID `ibmsys1`. If you choose an OpenStack cloud environment, login with your default cloud-init user ID. The private key and passphrase can be downloaded from instances 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`.

a. `awsstop`

Wait for few minutes for a running ZD&T to stop.

b. `awsstart aprof1`

Wait for few minutes for ZD&T to get ready.

c. `ipl 0a80 parm 0a82CS`

Monitor the console for any outstanding message.

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.

Provisioning instances to a Docker environment

Learn about the steps to provision instances from an image to a Docker environment, or resume a failed provisioning.

To provision instances from an image to a Docker environment, complete the following steps:

1. From the drop-down menu in the **TARGET ENVIRONMENT** pane, select a Docker environment to provision instances.
2. Click **Manage instances** on the home page. Or, click the **Instances** tab on the header, and click **Provision instance**.
3. Enter a unique name that can be used to recognize your instance.
4. Select an image and the version you want to provision to the container.
5. Optional: Enter some notes that can help you identify the instance.
6. If you provision an instance from an image that contains data set components. Enter the credentials to access the source environment.

The credentials are needed to restore data sets in a more flexible and efficient way. The ZD&T Enterprise Edition needs to log in to the source environment to run the job or program.

Note: If you provision an instance from a z/OS image that requires the credentials, you must enter the initial password of the source environment. However, if you provision an instance from an ADCD image that requires the credentials, you can set the initial password for the user ID 'IBMUSER'. This password will be used when you log in to the final emulated z/OS system.

7. Configure the provisioning in one of the following ways.
 - Clear the **Advanced settings** option, and click **Provision instance** to start the provisioning.
 - Enable the **Advanced settings** option. 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.
- Notes:**
- 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 that the emulator performs well, the total quantity of CPs and zIIPs should be less than the number of real cores on the target environment.
- a. Override the IPL command if you want. Specify the attributes of the device address, IODF address, and LOADxx suffix.
 - c. Specify the private port of your Docker environment.

8. Click **Provision instance**.

Provisioning instances to a Linux target environment

Learn about the steps to provision instances to a Linux target environment.

To provision an instance to a Linux target environment, complete the following steps:

1. From the drop-down menu in the **TARGET ENVIRONMENT** pane, select a Linux target environment to provision instances.
2. Click **Manage instances** on the home page, or click the **Instances** tab on the header, and click **Provision instance**.
3. Enter a unique name that can be used to recognize the instance.
4. Select an image that you want to provision. If the image has multiple versions, then select a version.
5. Optional: Enter some notes that can help you identify the instance.
6. Provide the required credentials of the target environment in one of the following ways.

- Select **Credentials**, and enter the user credentials that can access the target environment.

Note: If you provision instances by using a non-root user ID, make sure that the user ID has sudo access. For more information, see [Sudo access configuration](#).

- Select **Certificate**, and enter the user ID along with your OpenSSH private key certificate file. If your key file requires a passphrase, enter the passphrase.

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`.

7. Optional: Provide the deployment directory.

The deployment directory is a directory where ZD&T Enterprise Edition stores installation files and volume files during the provisioning 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-provisioning validation step will automatically check whether the target environment meets the requirements of a provisioning. For more information about the items that will be checked, see items 1 - 8 in [“Pre-provisioning validation”](#) on page 136.

8. If you provision an instance that contains data set components. Then, enter the credentials to access the source environment.

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 source environment to run the job or program.

Note: If you provision an instance from a z/OS image that requires the credentials, you must enter the initial password of the source environment. However, if you provision an instance from an ADCD image that requires the credentials, you can set the initial password for the user ID 'IBMUSER'. This password will be used when you log in to the final emulated z/OS system.

9. Configure the provisioning in one of the following ways.

- Clear the **Advanced settings** option, and click **Provision instance** to start the provisioning.
- Enable the **Advanced settings** option. 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.

Notes:

- 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 **Provision instance**.

After you click **Provision instance**, the pre-provisioning 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-provisioning validation”](#) on page 136.

Provisioning instances to an OpenStack environment

Learn about the steps to provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to an OpenStack environment, or resume and update a provisioning. 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 provision instances from created images to an OpenStack environment, complete the following steps:

1. From the drop-down menu in the **TARGET ENVIRONMENT** pane, select a cloud target environment to provision instances.
2. Click **Manage instances** on the home page, or click the **Instances** tab on the header, and click **Provision instance**.
3. Enter a unique name that can be used to recognize the instance.
4. Select an image that you want to provision. If the image has multiple versions, then select a version.
5. Optional: Enter some notes that can help you identify the instance.
6. Choose the level of your authentication, **Project** or **Domain**, and enter the credential of your cloud account.
7. Enter the credentials that are used to log in to the final emulated IBM Z environment.

Note: If you provision an instance from a z/OS image that requires the credentials, you must enter the initial password of the source environment. However, if you provision an instance from an ADCD image that requires the credentials, you can set the initial password for the user ID 'IBMUSER'. This password will be used when you log in to the final emulated z/OS system.

8. Configure the provisioning in one of the following ways.
 - Clear the **Advanced settings** option, and click **Provision instance** to start the provisioning.
 - Enable the **Advanced settings** option. 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.

Notes:

- 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 that the emulator performs well, the total quantity of CPs and zIIPs need to 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.
 - c. Then, click **Provision instance**.

Provisioning instances to a Red Hat OpenShift environment

Learn about the steps to provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to a Red Hat OpenShift environment, or use a cloud-ready image to create an instance in the Red Hat OpenShift environment.

To provision instances from created images to a Red Hat OpenShift environment, complete the following steps:

1. From the drop-down menu in the **TARGET ENVIRONMENT** pane, select a Red Hat OpenShift target environment to provision instances.

2. Click **Manage instances** on the home page, or click the **Instances** tab on the header, and click **Provision instance**.
3. Enter a unique name that can be used to recognize the instance.
4. Select an image that you want to provision in one of the following ways.
 - Select a created image by following these steps:
 - a. [“Creating z/OS images” on page 134](#) or [“Creating ADCD images” on page 135](#).
 - b. If the image has multiple versions, then select a version.
 - c. Enter the credentials that are used to log in to the final emulated IBM Z environment.

Notes:

- If you provision an instance from a z/OS image that requires the credentials, you must enter the initial password of the source environment. However, if you provision an instance from an ADCD image that requires the credentials, you can set the initial password for the user ID 'IBMUSER'. This password will be used when you log in to the final emulated z/OS system.
 - If you provision an instance from a cloud-ready image, this step is skipped.
 - Select a cloud-ready image by following the steps at [“Converting an image to a cloud-ready image” on page 135](#).
5. Configure the provisioning in one of the following ways.
 - Clear the **Advanced settings** option.
 - Enable the **Advanced settings** option. Then, complete the following optional steps:
 - a. Define the central processors (CPs), integrated information processors (zIIPs), and available system memory size that are allocated to the emulator.

Notes:

- 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 that the emulator performs well, the total quantity of CPs and zIIPs need to 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 volume device address, and LOADxx suffix.
 - c. Specify the port of the service that is running on the z/OS that you want to expose.
6. Click **Provision instance** to start the provisioning.

Monitoring and managing all created assets and target environments

After you create components or images, or provision a z/OS instance that is running on an emulated IBM Z hardware from a created image, you can monitor and manage all of the created assets and provisioning. Also, you can take actions directly for the created assets and provisioning.

Monitoring and managing all created components

After you created a component, you can check the status of the extraction, and the details of your created components.

Editing created components

After you created a component, you can edit the contents of the component by completing the following steps:

1. Click the **Edit component** icon.
2. Change the component name if you want.
3. Enable the **Adjust advanced settings** option, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.

4. Modify the volumes or data sets that you selected. Or, reselect the volumes or data sets for the component.
5. Click **Update component**. Or, click **Schedule extraction** to specify the date and time when you want to update the component.

Note:

- A component that is being extracted, scheduled, or provisioned cannot be edited. However, you can create a new version or duplicate the component to edit and work with it.
- The component type cannot be edited.

Resuming a failed extraction of a component

If the extraction of a component failed, you can resume the component by completing the following steps:

1. Click the **open and close list of options** icon, and select **Resume component**.
2. Verify the details of the component, and click **Yes, perform extraction**.

Refreshing the extraction of a component

To refresh the extraction of a component, complete the following steps:

1. Click the **open and close list of options** icon, and select **Refresh component**.
2. Verify the details of the component, and click **Yes, refresh**.

Creating a new version of a component

When you create a 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 **open and close list of options** icon, and select **Create new version**.
2. Log in to the specified IBM Z mainframe if required.
3. Change the component name if you want.
4. Enable the **Adjust advanced settings** option, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.
5. Add contents, or reselect new contents that you need for the new version.
6. Click **Create component**. Or, click **Schedule extraction**.

Note:

- When you create a new version of a component, you can add, remove, or reselect the volumes or data sets for the new version.
- The **New version** button is available only for the newest version of a component.
- The component type cannot be edited.

Duplicating a created component

If you want to copy a created component with a separate lifecycle that starts from version 1, complete the following steps:

1. Click the **open and close list of options** icon, and select **Duplicate component**.
2. Log in to the specified IBM Z mainframe if required.
3. Create a new component name.
4. Enable the **Adjust advanced settings** option, and enter an integer in the box of **Concurrent extractions** to define the number of artifacts that you want to extract simultaneously.

5. Duplicate the same contents, or reselect the volumes or add data sets that you need for the duplicated component.
6. Click **Create component**. Or, click **Schedule extraction**.

Deleting created components

To delete a component, complete the following steps:

1. Click the **open and close list of options** icon.
2. Select **Delete component**, and click **Delete**.

Note:

- Only the components that were created from IBM Z can be deleted separately.
- a component that is being created cannot be deleted.
- a component that is used in an image cannot be deleted. To delete a component, you must delete all of the images that use the components first.
- If you delete a component and image, the volume files that are located at the storage server will be deleted automatically.

Canceling a scheduled extraction

If you want to cancel the extraction of a component that is scheduled to be created, complete the following steps:

1. Click the **open and close list of options** icon, and select **Cancel scheduled component extraction**.
2. Click **Yes, cancel scheduled extraction** to confirm cancellation.

Stopping the extraction of a component

To stop a component that is being extracted, complete the following steps:

1. Click the **open and close list of options** icon, and select **Stop extraction**.
2. Click **Yes, stop extraction** to confirm it.

Migrating 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 provisioned.

To migrate a component, complete the following steps:

1. Click the **Migrate** icon, and select **Perform extraction**.
2. Enter the credentials of the IBM Z mainframe that you create component from.
3. Click **Complete**.

In a migration, the contents of a component might be changed if the original contents that came from the mainframe system changed.

Downloading log

To check the details of the created components, click the **open and close list of options** icon. Then, select **Download log**.


Note: Only the 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 images

After you create an image, you can stop, resume, or delete the image. Also, you can retrieve logs to check the details of the image.

Modifying selected components for a created image

After you created an image, you can edit the components that you selected for the image on the **Images** page by completing the following steps:


1. Click the **Edit** icon  , and then select **Edit image**.
2. Optional: Change the image name if you want.
3. Optional: Enable the **Adjust advanced settings** option. After this option is enabled, you can specify the number of cryptographic coprocessors or edit image properties.
4. Modify the components that are selected, or reselect components for the image.
5. Click **Update image**.

Note:

- Only the images that were created from IBM Z mainframe and ADCD can be edited.
- An image that is being extracted, scheduled, or provisioned cannot be edited. However, you can create a new version or duplicate the image to edit and work with it.

Editing image properties

After you created an image, you can edit the image properties on the **Images** page by completing the following steps:

1. Click the **Edit** icon  , and then select **Edit image properties**.
2. Modify the system configurations, and then click **Next**.
3. Modify the networking configurations, and then click **Save**.


You can also edit image properties when modifying selected components for a created image.

Note:

- Only the images that were created from IBM Z mainframe and ADCD can be edited.
- An image that is being extracted, scheduled, or provisioned cannot be edited. However, you can create a new version or duplicate the image to edit and work with it.


Duplicating a created image

To copy a created image with a separate lifecycle that starts from version 1, complete the following steps:

1. On the **Images** page, select **Duplicate image** from the action menu .
2. Log in to the specified IBM Z mainframe if required.
3. Create a new image name.
4. Optional: Enable the **Adjust advanced settings** option, and enter an integer in the box of **Cryptographic coprocessor** and **Domain**.
5. Duplicate the same components, or reselect the components that you want for the duplicated image.
6. Click **Create image**.

Creating a new version of an image

When you create an 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 that you selected but do not want to modify the existing image, you can create a new version by completing the following steps:


1. On the **Images** page, select **Create new version** from the action menu .
2. Log in to the specified IBM Z mainframe if required.
3. Change the image name if you want.
4. Optional: Enable the **Adjust advanced settings** option, and enter an integer in the box of **Cryptographic coprocessor** and **Domain**.
5. Add new components, or reselect the components that you need for the new version.
6. Click **Create image**.

Note:

- The **Create new version** button is available only for the newest version of an image.
- When you create a new version of an image, you can add, remove, or reselect the components for the new version.

Deleting a created image

To delete an image that was created, complete the following step:

1. On the **Images** page, select **Delete image** from the action menu .
2. Click **Yes, delete**.

Note:

- An image that is being extracted, scheduled, or provisioned cannot be deleted. To delete an image that was provisioned, you must remove the image from all target environments first.
- If you delete an image that was created from IBM Z components or ADCD, the components that are used in the image will not be deleted. However, if you delete an image that was created from an exiting ZD&T, the components that are used in the image will also be deleted.

Migrating images to the current version

To ensure the 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. Go to the **Images** page to verify whether an image shows the migration icon. If the icon exists, migrate the images, otherwise the images cannot be provisioned.

To migrate images that can be used in the current version, complete the following steps:

1. Click the **Migrate** icon, and select **Migrate image to current version**.
2. Log in to the specified IBM Z mainframe that you create images from.
3. Click **Complete**.

The migration collects only the information about a list of offline direct access storage device (DASD) that is needed from the mainframe systems, which will not change the contents of images.

Downloading log

To check the details of a created image, go to the **Images** page, and then select **Download log** from the action menu .

Note:

- Only the 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 provisioning to target environments

After you provision a z/OS instance that is running on an emulated IBM Z hardware from a created image to a target environment, you can use the **Instances** page to check the status of the instance. Also, you can remove or resume the provisioned instances, retrieve logs or the private key, and start or stop the z/OS emulator.

Checking the status of the provisioning process

After you provisioned an image, the **Instances** page shows all the instances with an overall percentage of the provisioning process.

If you want to check the details of a provisioning process, complete the following steps.

1. Click the drop-down menu of **TARGET ENVIRONMENT** on the upper-right, and select a target environment that you provisioned instances.
2. Click **Manage instances** on the home page, or click **Instances** on the top tab.
3. Click the percentage of the provisioning you need to check. Then, you can view the overall progress, current operation, and the provisioning detail that shows each particular step in the provisioning with a specific percentage.

Resuming a failed provisioning

To resume a failed provisioning of an image, complete the following steps:

1. Click the action icon beside the instance, and select **Resume provision**.
2. Provide the required credentials.
3. Select if you want to configure the **Advanced settings**, then configure the central processors (Cps), system memory (GB), the automatic IPL command and so on.
4. Click **Provision instance**.

Removing provisioned instances from Red Hat OpenShift environments

To remove provisioned instances from Red Hat OpenShift environments, complete the following steps:

1. Click the action icon beside the instance, and select **Deprovision**.
2. Check the box of **Force removal** if you want.
3. Click **Deprovision** to confirm.

Removing provisioned instances from Docker environments

To remove provisioned instances from Docker environments, complete the following steps:

1. Click the action icon beside the instance, and select **Deprovision**.
2. Click **Deprovision** to confirm.

Removing provisioned instances from Linux environments

To remove provisioned instances from Linux environments, complete the following steps:

1. Click the action icon beside the instance, and select **Remove image**.
2. Select or clear the following options. Then, click **Next**.
 - Remove ZD&T and revert network configurations.

- Remove the image.

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

3. Enter the required credentials, and click **Submit**.

Removing provisioned instances from cloud environments

To remove provisioned instances from cloud environments, complete the following steps:

1. Click the action icon beside the instance, and select **Remove image**.
2. Enter the required credentials, and click **Submit**.

Updating a successful provisioning

If a successful provisioning contains an image that has multiple versions, you can update the provisioning by completing the following steps:

1. Click the action icon beside the instance, and select **Update image**.
2. Select a version of the image that is contained in the provisioning.
3. Complete steps 6 - 9 in [“Provisioning instances to a Linux target environment”](#) on page 140.
4. Click **Provision instance**.

Starting the emulator z/OS

If the emulator z/OS is offline or stopped, start the emulator z/OS by completing the following steps:

1. Click the action icon beside the instance, and select **Start emulator z/OS**.
2. Enter the required credentials, and click **Submit**.

Stopping the emulator z/OS

To stop a running emulator z/OS, complete the following steps:

1. Click the action icon beside the instance, and select **Stop emulator z/OS**.
2. Enter the required credentials, and click **Submit**.

Updating the emulator to the latest version

After an instance is provisioned from a created image, the emulator version number displays under the instance. 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.

Note: An instance that is being provisioning cannot be removed or resumed. Also, you cannot start or stop the z/OS emulator currently.

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 components and images, provisioning instances from created images, and monitoring images and instances.

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/api/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 171](#) 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.encryption.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+NgYthDZXhQgARtB"/>  
</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_V13.1.0.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 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/api/explorer/>.

Connecting to target environments

To connect to target environments, you can use 'curl' to call REST APIs.

About this task

To connect to target environments by using 'curl' to call REST APIs, 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/api/explorer/>

Procedure

1. Configure the connection to the target environments if you want to provision instances from created images to Linux environments.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "your_target_system_host_name", "sshPort": 22, "label": "zdtForTesting", "installOSPackages": true, "concurrentVolumeTransferNumber": 3, "configNetwork": true, "configUser": true, "syntax": "sudo ${command}", "syntaxForValidation": "sudo -n -l", "networkInterface": "eth1", "tcpPortRules": "0:21 \n 23:2021 \n 2022>22 \n 2023:3269 \n 3271:9449 \n 9452:65535", "udpPortRules": "111 \n 514 \n 1023 \n 1044:1049 \n 2049", "zdtManaged": "true", "hasRootAccess": "true"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystem'
```

2. Configure the connection to the cloud platform if you want to provision instances from created images to OpenStack environments.

```
curl -k -u zdtadmin: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'
```

Creating and managing target environments

You can call REST APIs to create and manage target environments.

Creating and managing Linux target environments

You can call REST APIs to create and manage Linux target environments.

About this task

Starting with V13.3.0, you can create a Linux target environment with a UUID contained in the response by using the REST API. This UUID represents the Linux target environment that you create. You can manage the Linux target environment by using the UUID. However, limitations exist. Migration is not supported yet.

To create or manage Linux target environments by calling REST APIs, you need to use your own setup variables to replace the *italic* text in the examples. For more information about the descriptions and available values of each field, see the REST API specification at the following URL:

`https://<your-tools-server-host-name>:9443/api/explorer/`

Procedure

1. Use the following REST API to create a Linux target environment. After a Linux target environment is created, the response contains a UUID that represents the Linux target environment.

```
curl -X POST -k -u zdtadmin:password "https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/v1/linux" -H "accept: application/json;charset=utf-8" -H "Content-Type: application/json;charset=utf-8" -d '{"hostname":"your_linux_target_system_host_name","ssh-port":22,"label":"your_label","install-os-packages":true,"concurrent-volume-transfer-number":3,"config-network":true,"config-user":true,"syntax":"sudo ${command}","syntax-for-validation":"sudo -n -l","network-interface":"eth1","zos-ip-address":"your_z-system_ip_address","tcp-port-rules":"0:21 \ 23:2021 \ 2022>22 \ 2023:3269 \ 3271:9449 \ 9452:65535","udp-port-rules":111,"zdt-managed":true,"has-root-access":true,"description":"string"}'
```

2. After a Linux target environment is created, you can use the following REST APIs to get, update, or delete the target environment.

Getting a Linux target environment

UUID is required to get a Linux target environment. To get a Linux target environment, use the following REST API.

```
curl -X GET -k -u zdtadmin:password "https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/v1/linux/your-linux-target-system-uuid" -H "accept: application/json;charset=utf-8"
```

Updating a Linux target environment

UUID is required to update a Linux target environment. To update a Linux target environment, use the following REST API.

```
curl -X PUT -k -u zdtadmin:password 'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/v1/linux' \ -H 'accept: application/json;charset=utf-8' \ -H 'Content-Type: application/json;charset=utf-8' \ -D '{ "label": "updatedLabel", "uuid": "your_target_linux_target_system_uuid" }'
```

Deleting a Linux target environment

UUID is required to delete a Linux target environment. To delete a Linux target environment, use the following REST API. After a Linux target environment is deleted, the response displays 200 (OK) status code.

```
curl -X DELETE -k -u zdtadmin:password 'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/v1/linux/<UUID>?revert-network-config=false&delete-volume-files=false' \ -H 'accept: application/json;charset=utf-8' \ -H 'Content-Type: application/json;charset=utf-8'
```

The following two query parameters are optional:

revert-network-config

Specifies whether the network configuration must be reverted to its original state before the provision was done. The default value is false.

delete-volume-files

Specifies whether the volume files must be deleted from the Linux target environment. The default value is false.

Creating and managing Docker target environments

You can call REST APIs to create and manage Docker target environments.

About this task

To create or manage Docker target environments by calling REST APIs, you need to use your own setup variables to replace the *italic* text in the examples. For more information about the descriptions and available values of each field, see the REST API specification at the following URL:

<https://<your-tools-server-host-name>:9443/api/explorer/>

Procedure

1. Use the following REST API to create a Docker target environment.

```
curl -X POST "https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/docker-system" -H "accept: application/json;charset=utf-8" -H "Content-Type: multipart/form-data" -F "docker-system={ \"hostname\": \"your_docker_system_host_name\", \"port\": 2376, \"lower-port-value\": 40000, \"max-containers\": 3, \"label\": \"docker-test\", \"description\": \"description\", \"concurrent-transfers\": 4 }" -F "ca-certificate=@ca.pem;type=application/x-x509-ca-cert" -F "server-certificate=@cert.pem;type=application/x-x509-ca-cert" -F "client-key=@key.pem;type=application/x-x509-ca-cert"
```

After a Docker target environment is created, all the information of the Docker target environment, including a UUID that represents the target environment, is shown in the response. See the following example:

```
{
  {
    "uuid": "c1642b32-7b51-4551-9479-de185a344ca5",
    "creation-time": 1637167044195,
    "deletable-details": null,
    "label": "docker-test",
    "description": "description",
    "hostname": "your_docker_system_host_name",
    "port": 2376,
    "concurrent-transfers": 4,
    "type": "DOCKER",
    "online": false,
    "user": {
      "uuid": "0c07f3e6-4a1a-41a0-8a02-d56291ee30be",
      "creation-time": 1636575518386,
      "name": "zdtadmin",
      "show-welcome": false,
      "login-time": 1637162131998,
      "login-count": 41,
      "roles": [
        {
          "uuid": "61be84e5-1589-4b41-a123-b311e1d3906c",
          "creation-time": 1636575518381,
          "name": "Administrator"
        }
      ]
    },
    "lower-port-value": 40000,
    "upper-port-value": 40300,
    "max-containers": 3
  }
}
```

2. After a Docker target environment is created, you can use the following REST APIs to get, update, or delete the target environment.

Getting a Docker target environment

UUID is required to get a Docker target environment. To get a Docker target environment, use the following REST API.

```
curl -X GET
'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/docker-system/<UUID>' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8'
```

If the call completes successfully, all the information of the Docker target environment is shown in the response. See the following example:

```
{
  {
    "uuid": "c1642b32-7b51-4551-9479-de185a344ca5",
    "creation-time": 1637167044195,
    "deletable-details": null,
    "label": "docker-test",
    "description": "description",
    "hostname": "your_docker_system_host_name",
    "port": 2376,
    "concurrent-transfers": 4,
    "type": "DOCKER",
    "online": false,
    "user": {
      "uuid": "0c07f3e6-4a1a-41a0-8a02-d56291ee30be",
      "creation-time": 1636575518386,
      "name": "zdtadmin",
      "show-welcome": false,
      "login-time": 1637162131998,
      "login-count": 41,
      "roles": [
        {
          "uuid": "61be84e5-1589-4b41-a123-b311e1d3906c",
          "creation-time": 1636575518381,
          "name": "Administrator"
        }
      ]
    }
  },
  "lower-port-value": 40000,
  "upper-port-value": 40300,
  "max-containers": 3
}
}
```

Updating a Docker target environment

UUID is required to update a Docker target environment. To update a Docker target environment, use the following REST API.

```
curl -X PUT 'https://<hostname>:9443/ZDTREST/zdtrts/targetEnvServices/docker-system' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8' \
-D '{
  "label": "updatedLabel",
  "uuid": "<UUID>"
}'
```

If the call completes successfully, the updated information of the Docker target environment is shown in the response. See the following example:

```
{
  {
    "uuid": "c1642b32-7b51-4551-9479-de185a344ca5",
    "creation-time": 1637167044195,
    "deletable-details": null,
    "label": "updatedLabel",
    "description": "description",
    "hostname": "your_docker_system_host_name",
    "port": 2376,
    "concurrent-transfers": 4,
    "type": "DOCKER",
    "online": false,
    "user": {
      "uuid": "0c07f3e6-4a1a-41a0-8a02-d56291ee30be",
      "creation-time": 1636575518386,
      "name": "zdtadmin",
      "show-welcome": false,
      "login-time": 1637162131998,
      "login-count": 41,
      "roles": [
        {
          "uuid": "61be84e5-1589-4b41-a123-b311e1d3906c",
          "creation-time": 1636575518381,
          "name": "Administrator"
        }
      ]
    }
  }
}
```

```

    },
    "lower-port-value": 40000,
    "upper-port-value": 40300,
    "max-containers": 3
  }s
}

```

Deleting a Docker target environment

UUID is required to delete a Docker target environment. To delete a Docker target environment, use the following REST API. After a Docker target environment is deleted, the response displays 200 (OK) status code.

```

curl -X DELETE
'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/docker-system/<UUID>' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8'

```

Creating and managing Red Hat OpenShift target environments

You can call REST APIs to create and manage Red Hat OpenShift target environments.

About this task

To create or manage Red Hat OpenShift target environments by calling REST APIs, you need to use your own setup variables to replace the *italic* text in the examples. For more information about the descriptions and available values of each field, see the REST API specification at the following URL:

<https://<your-tools-server-host-name>:9443/api/explorer/>

Procedure

1. Call the REST API to create a Red Hat OpenShift target environment.

Note: To retrieve the API URL of a Red Hat OpenShift target environment, log in to a Red Hat OpenShift Environment, navigate to **Overview** page, and copy the URL that is shown under **Cluster API Address**.

```

curl -X POST
'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/openshift-system' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8' \
-D '{
  "url": "https://api.<openshift-api-hostname>:6443",
  "token": "tJ2H4kThisIsASampleTokenepg0Arcxd88BmUTrce_2ho0YFaASImDsJZLZ0",
  "label": "openshiftTE",
  "storageClass": "rook-ceph-block",
  "namespace": "Sample-NS"
}'

```

If the REST API call is successful, the response that includes all the information of a newly created Red Hat OpenShift target environment will be shown as follows.

```

{
  "uuid": "8f3e7512-516f-4f22-974a-2f68f4ae7bee",
  "creation-time": 1615827720907,
  "deletable-details": null,
  "label": "testMachine1",
  "description": null,
  "hostname": "api.sample.openshift.ibm.com",
  "port": 0,
  "passphrase": null,
  "concurrent-transfers": 1,
  "type": "OPENSHIFT",
  "token": null,
  "namespace": "sampleNS",
  "storage-class": "sample-storage",
  "cloning": false
}

```

2. Call the REST API to get a Red Hat OpenShift target environment by UUID.

Note: UUID is required to get the Red Hat OpenShift target environment.

```
curl -X GET
'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/openshift-system/<UUID>' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8'
```

If the REST API call is successful, the response that includes all the information of the requested Red Hat OpenShift target environment will be shown as follows.

```
{
  "uuid": "8f3e7512-516f-4f22-974a-2f68f4ae7bee",
  "creation-time": 1615827720907,
  "deletable-details": null,
  "label": "testmachine",
  "description": null,
  "hostname": "api.sample.openshift.ibm.com",
  "port": 6443,
  "passphrase": null,
  "concurrent-transfers": 1,
  "type": "OPENSIFT",
  "token": null,
  "namespace": "sampleNS",
  "storage-class": "sample-storage",
  "cloning": false
}
```

3. If you need to update a Red Hat OpenShift target environment, call the following REST API.

Note: UUID is required to update the Red Hat OpenShift target environment.

```
curl -X PUT
'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/openshift-system' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8' \
-D '{
  "label": "updatedLabel",
  "uuid": "960065ea-5035-4a26-aa77-de799ccd7dcd"
}'
```

If the REST API call is successful, the response that includes the updated information of the requested Red Hat OpenShift target environment will be shown as follows.

```
{
  "uuid": "8f3e7512-516f-4f22-974a-2f68f4ae7bee",
  "creation-time": 1615827720907,
  "deletable-details": null,
  "label": "updatedLabel",
  "description": null,
  "hostname": "api.sample.openshift.ibm.com",
  "port": 0,
  "passphrase": null,
  "concurrent-transfers": 1,
  "type": "OPENSIFT",
  "token": null,
  "namespace": "sampleNS",
  "storage-class": "sample-storage",
  "cloning": false
}
```

4. If you need to delete a Red Hat OpenShift target environment, call the following REST API by using a UUID.

Note: The following two query parameters are required.

- Cleanup: If the Boolean value is set to *true*, the role and role-binding for the service account in the Red Hat OpenShift environment will be deleted. This query parameter is required.
- Admin-token: The optional token of an account can delete the role and role-bindings. This token is not needed if the cleanup parameter is set to false.

UUID path parameter is required to delete a Red Hat OpenShift target environment.

```
curl -X DELETE
'https://<hostname>:9443/ZDTREST/zdtrs/targetEnvServices/openshift-system/<UUID>?admin-
token=<admintoken>&cleanup=true' \
-H 'accept: application/json;charset=utf-8' \
-H 'Content-Type: application/json;charset=utf-8'
```

If the REST API call is successful, the response displays 200 (OK) status code.

Creating components

If you want to extract data from z/OS Systems, you must create components before you create images.

Creating volumes components from z/OS system

Learn about the steps to create components from IBM Z volumes 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 '{ "system-uuid": "systemuuid", "username":
"zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/
auth-services/v3/authenticate-ssh'
```

2. Call the REST APIs to get the information on all volumes for the 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/z-services/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":18920109
60}, {"id":0,"name":"A27CCC","extractionTime":"1960-01-01
23:03:20","deviceNumber":"04BE","deviceType":"3390","progress":0,"compressed":0,"size":28380164
40}, {"id":0,"name":"A45SYS","extractionTime":"1960-01-01
23:03:20","deviceNumber":"0492","deviceType":"3390","progress":0,"compressed":0,"size":85140493
20},...]
```

3. Create the component by including your selected volumes that are returned from above call.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/
json' --header 'Accept: application/json' --header 'SSHAuthorization:
{"token": "VVNFUjM1NTE4MTkwMzg4MzI=", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22} ' -d
'{ "name": "testComp-fromLinux", "version": 1, "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 components from IBM Z data sets 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 '{ "system-uuid": "systemuuid", "username":
"zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/
auth-services/v3/authenticate-ssh'
```


The following response is displayed.

```
{ "expiration": 1516202935577, "username": "ZOSUSER", "password": "ZOSPW", "token": "VVNFUjM1NTE4ODczMjY3NjU=", "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": "cm9vdC0xMzM5NzAzMjQ2", "hostname": "YOUR_Z_OS_HOST_NAME", "port": 22}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/z-services/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 your selected data sets returned from 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_USER", "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 an IBM Z mainframe” on page 120](#).
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 '{"system-uuid": "systemuuid", "username": "zosuser", "password": "zospw"}' https://your_tools_server_host_name:9443/ZDTREST/zdtrs/auth-services/v3/authenticate-ssh'
```

The following response is displayed.

```
{ "expiration": 1516202935577, "username": "ZOSUSER", "password": "ZOSPW", "token": "VVNFUjM1NTE4ODczMjY3NjU=", "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 z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json'
--header 'Accept: application/json' -d '{ "system-uuid": "systemuuid", "username":
"zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/
auth-services/v3/authenticate-ssh'
```

The following response is displayed.

```
{ "token": "cm9vdC02NjQ3ODgxMTY=" }
```

2. Obtain the list of Db2 subsystems from the z/OS system by passing the uuid of the z/OS system.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --
header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/
source-env-services/db2-subsystem?uuid=abb1b85a-c148-4628-bfe8-749c6751c5b8'
```

After the request is complete, the following JSON response with a list of subsystems is displayed.

```
[
  {
    "uuid": "4c52d3e0-1a0b-4b6e-898f-38f1db4286a0",
    "creation-time": 1563540882793,
    "subsystem-id": "DC1A",
    "subsystem-location": "DC1A",
    "sdsnext": "SYS1.DC1A.SDSNEXIT",
    "sdsnload": "SYS1.DC1A.SDSNLOAD",
    "run-lib": "",
    "use-admin": true,
    "admin-program": "ADB2GEN",
    "admin-plan": "adb2gen",
    "admin-program-lib": "DSN.TOOLS.ADB1201.SADBLLIB"
  },
  {
    "uuid": "e729459a-2540-46f2-ab3d-cfcf5f20bb76",
    "creation-time": 1602103833447,
    "subsystem-id": "DB1E",
    "subsystem-location": "DB1E",
    "sdsnext": "SYS1.DB1E.SDSNEXIT",
    "sdsnload": "SYS1.DB1E.SDSNLOAD",
    "run-lib": null,
    "use-admin": true,
    "admin-program": null,
    "admin-plan": "ADB2GEN",
    "admin-program-lib": "DSN.TOOLS.ADB1201.SADBLLIB"
  }
]
```

3. Obtain the list of schemas by using the subsystem ID with the hostname, and port of the z/OS system, providing the authentication to 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_zsystem_host_name", "port": 22}'
'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/db2/v1/schemas?
hostName=your_zsystem_host_name&sshPort=22&subsystemName=DC1A'
```

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, and providing the authentication to 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_zsystem_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, providing the authentication to 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_zsystem_host_name", "port": 22}' -d '{"name": "test", "version": 1, "systemType": "COMP_SYSTEM_TYPE_ZSYSTEM_USER", "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 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.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{"system-uuid": "systemuuid", "username": "zosuser", "password": "zospw"}' https://your_tools_server_host_name:9443/ZDTREST/zdtrs/auth-services/v3/authenticate-ssh
```

The following response is displayed.

```
{"token": "cm9vdC02NjQ3ODgxMTY="}
```

2. Call the REST API to get the list of AD servers that are defined in ZD&T.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' https://your_tools_server_host_name:9443/ZDTREST/zdtrs/integration-services/ad-servers
```

3. Call the REST API to get the list of AD project that is defined in AD by providing the uuid of the selected AD server.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' https://your_tools_server_host_name:9443/ZDTREST/zdtrs/integration-services/ad-projects/e3c9166d-3d2f-4c4f-8a88-a440826b2973
```

4. Call the REST API to get the list of data sets for a specific project that is defined in a specific AD server, by providing the uuid of the AD server and project.

```
curl -k -u zdtadmin:password -X GET --header 'Accept: application/json' https://your_tools_server_host_name:9443/ZDTREST/zdtrs/integration-services/ad-project-datasets/ad/{uuid}/projects/{project-uuid}
```

5. Create the component by including your selected data sets returned from 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_USER", "description": "ds_comp", "dataSets": [ { "name": "SYS1.A39AAA.LPALIB" }, { "name": "SYS1.AF0MMOD1" }, { "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 volume components from existing ZD&T instances

Learn about the steps to create volume components from existing ZD&T instances.

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 '{ "system-uuid": "systemuuid", "username": "zosuser", "password": "zospw"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/auth-services/v3/authenticate-ssh'
```

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/zdt-services/volumes?uuid=systemuuid'
```

The following response is displayed.

```
{"zosAllVolumes": [{"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}, ...], "zosInfo": {"sysinfo": {"loadsuf": "AU", ...}, "disks": [...], ...}}
```

3. Call the REST API to authenticate the SSH connection to the Linux system that is running the ZD&T instance.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "hostname": "YOUR_LINUX_HOST_NAME", "port": 22, "username": "LinuxUSER", "password": "LinuxPW"}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/auth-services/v2/authenticateSSH'
```

4. Create the component by providing the authentication to the Linux system. The authentication needs to include your selected volumes, and the whole "zosInfo" structure in the "zsystemInformation" section that is returned from step 2 GET VOLUME call.

```
curl -k -u zdtadmin:password -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header 'SSHAuthorization: {"token": "VVNFUjM1NTE4MTkwMzg4MzI=", "hostname": "YOUR_LINUX_HOST_NAME", "port": 22} ' -d '{ "name": "testComp-fromLinux", "zSystem": { "hostname": "YOUR_Z_OS_HOST_NAME", "port": 2022 }, "systemType": "COMP_SYSTEM_TYPE_ZDT_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"}, "zsystemInformation": {"sysinfo": {"loadsuf": "AU", ...}, "disks":
```

```
[...],...} } }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/  
zdt-volume'
```

Checking the status of components creation

After the components are created, you can call the REST API to check the status of the created components.

Procedure

Run the following command, and use the variables of your ZD&T instances to replace the following *italic* text.

```
curl -k -u ztdadmin:password -X GET --header 'Accept: application/  
json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/components/  
YOUR_Z_OS_HOST_NAME?sourcePort=YOUR_Z_OS_HOST_PORT'
```

Check the field in the component level. If the message "status": "COMP_SCHEDULED" is displayed, it means that the component is still being extracted.

When the message "status": "COMP_EXTRACTED" is displayed, it means that the component extraction is done, and the component is ready to be used.

Creating images

After the images are created, you can store the image data in the specified storage server.

Creating images from z/OS volumes or data sets

Learn about the steps to create images from z/OS volumes or data sets.

Procedure

1. Create the images by including some of the components that you have created earlier.

```
curl -k -u ztdadmin:password -X POST --header 'Content-Type: application/json' --  
header 'Accept: application/json' -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,  
  "message": "These data sets cannot be extracted because their containing volumes cannot be  
found: [SYS1.A39AAA.LPALIB , SYS1.AFOMMOD1 "],  
  "level": {  
    "name": "SEVERE"  
  },  
  "cause": "",  
  "resolution": ""  
}
```

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 171.

2. Optional: Try again when an error code is displayed at step 1.

```
curl -k -u ztdadmin:password -X POST --header 'Content-Type: application/  
json' --header 'Accept: application/json' -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 images is returned.

Creating an image from ADCD package

Learn about the steps to create images from ADCD package by using REST API.

Procedure

1. Copy your ADCD package to the 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 provision an instance from the created image.
2. Get the ADCD package information before you create any ADCD application for the first time.

```
curl -k -u zdtadmin:password -X GET 'https://your_tools_server_host_name:9443/ZDTREST/zdtis/adcd-services/packages' --header 'accept: application/json;charset=utf-8'
```

The following response is displayed.

```
[{
  "id": 708,
  "uuid": "3d3e891b-f961-45c8-9a0e-cc030906dca0",
  "name": "z/OS V2.4 May 2020",
  "rsu": "2003",
  "restoreDataset": true,
  "zsystemInformation": null,
  "zdtVolumeSms": true,
  "restoreDb2": true
}]
```

3. Get the component names that are 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 'https://your_tools_server_host_name:9443/ZDTREST/zdtis/componentServices/components?host=may2020' --header 'accept: */*'
```

The following response is displayed.

```
[ { "id": 8,
  "uuid": "1db5685d-9b25-4cb7-80a5-0de1cd7f7850",
  "name": "Customer Information Control System (CICS) - 5.4",
  "systemType": "COMP_SYSTEM_TYPE_ADCD_USER",
  ... },
  {
    "id": 9,
    "uuid": "26daeb0f-7429-45b9-a776-67cdc0a6eb28",
    "name": "z/OS - 2.4",
    "systemType": "COMP_SYSTEM_TYPE_ADCD_ENV",
    "componentType": "COMP_WITH_VOLUME",
    "zSystem": {
      "id": 101,
      "hostname": "may2020",
      "type": "SYSTEM_TYPE_ADCD",
      "port": 22,
      "adcdPackage": {
        "id": 1,
        "uuid": "3d3e891b-f961-45c8-9a0e-cc030906dca0",
        "name": "z/OS V2.4 May 2020",
        "rsu": "2003",
        ...
      }
    },
    "volumes": [
      {
        "id": 27,
        "name": "B4SYS1",
        ...
      }
    ]
  }
]
```

4. Create an image.

- **Creating an image from ADCD package**

If you want to create an image from ADCD 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": "may2020", "port": 22 }, "ftpSystem": { "hostname": "your_storage_server_host_name" }, "components": [ { "name": "z/OS - 2.4", "version": 1 } ] }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/ADCDApplicationImage'
```

Then, the generated information of image is returned.

- **Creating an image from ADCD package and data sets from other z Systems**

If you want to create an image from ADCD 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": "may2020", "port": 22 }, "ftpSystem": { "hostname": "your_storage_server_host_name" }, "components": [ { "name": "z/OS - 2.4", "version": 1 }, { "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/ADCDApplicationImage'
```

- **Creating an image from ADCD package and CICS components**

If you want to create an image from ADCD 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": "may2020", "port": 22 }, "ftpSystem": { "hostname": "your_storage_server_host_name" }, "components": [ { "name": "z/OS - 2.4", "version": 1 }, { "name": "CICS-component1", "zSystem": { "hostname": "z-system1", "port": 22 }, "version": 1 } ] }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/ADCDApplicationImage'
```

Checking the status of images creation

After the images are created, you can call the REST API to check the status of the created images.

Procedure

Run the following command, and use the variables of your ZD&T instances to replace the following *italic* text.

```
ccurl -k -u zdtadmin:password -X GET --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/application/YOUR_Z_OS_HOST_NAME?applname=testApp1-fromLinux'
```

Check the field in the application level. If the message `"status": "APPL_IN_PROGRESS"` is displayed, it means that the image is still being extracted.

When the message `"status": "APPL_AVAILABLE"` is displayed, it means that the application extraction is done, and the image is ready to be used.

Editing created components

After you create volumes or data set components from z/OS systems, you can edit the component, or create a new version and duplicate for the component.

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 '{ "system-uuid": "systemuuid", "username":
"zosuser", "password": "zospw"}' https://your_tools_server_host_name:9443/ZDTREST/zdtrs/
auth-services/v3/authenticate-ssh'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVNFUjM1NTE4ODczMj
Y3NjU=","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 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 created images

After you create images, you can edit the image, or create a new version or a duplicate of the image.

Procedure

1. To edit an image that is not provisioned, run the following command.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":
{"domain":0,"ccp":0},"newInfo":{"name":"zdt-dev1.wholeimage.without.LOAD$1"},"addItems":
[{"name":"zdt-dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":[{"name":"zdt-
dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":false,"newVersion":fals
e,"sysResComponent":{"name":"zdt-dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}' https://
your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/application
```

2. To create a new version of an image, run the following command.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":
{"domain":0,"ccp":0},"newInfo":{"name":"zdt-dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":[{"name":"zdt-
dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":false,"newVersion":true
,"sysResComponent":{"name":"zdt-dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}' https://
your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/application
```

3. To duplicate an image to a new image, run the following command.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header
'Accept: application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":
{"domain":0,"ccp":0},"newInfo":{"name":"zdt-
dev1.wholeimage.with.LOAD$1.duplicated"},"addItems":[{"name":"zdt-dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":[{"name":"zdt-
dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":true,"newVersion":false
,"sysResComponent":{"name":"zdt-dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}' https://
your_tools_server_host_name:9443/ZDTREST/zdtrs/appServices/application
```


Provisioning instances from created images

After the images are created, you can provision a z/OS instance that is running on emulated IBM Z hardware from the created image to your Linux, OpenStack, or Docker target environment.

Provisioning instances to a Linux target environment

You can provision a z/OS instance that is running on emulated IBM Z hardware to a specific Linux target environment by using the UUID or by authenticating with a password or an ssh certificate key file.

About this task

Starting with V13.3.0, after a Linux target environment is created by using REST APIs, the response contains a UUID that represents the Linux target environment. You can use this UUID to provision instances to the target environment. However, limitations exist. Migration is not supported yet. Also, comparing with the REST APIs that are provided in earlier versions, some features are not supported, for example, updating a provisioning.

For more information about creating Linux target environments by using REST APIs, see [“Creating and managing Linux target environments”](#) on page 150.

Procedure

1. To provision instances to a Linux target environment by using the UUID, follow the steps:
 - a. Create a provisioning to a Linux target environment by using the following REST API. In this example, optional fields are added to specify the IPL parameters or to specify the z/OS user credential that is required to provision an instance from a created image with data sets.

```
curl -X POST -k -u ztdadmin:password "https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/linux" -H "accept: application/json;charset=utf-8" -H "Content-Type: application/json;charset=utf-8" -d '{"emulator":{"cp":3,"ram":6442450944,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"label":"testZdtAppl","description":"","target-uuid":"your_target_env_uuid","resume":false,"update":false,"application-uuid":"your_application_uuid"},"zos-creds":{"username":"uer","password":"password"},"deployment-directory":"/home/ibmsys1","linux-creds":{"username":"user","password":"password"}}'
```

- b. After you create a provisioning, you can use the following REST API to monitor the provisioning progress.

```
curl -N --http2 -H "Accept:text/event-stream" -k -u ztdadmin:password 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/monitor-services/v1/provision-status/{provision-uuid}'
```

If the response is "status": "SUCCEEDED", the provisioning is completed. To resume a failed provisioning, use the following REST API:

```
curl -k -u ztdadmin:password -X POST -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{"emulator":{"cp":1,"ram":3,"ziip":0},"ipl":{"device-address":"a80","iodfaddress":"0a82","load-suffix":"au"},"general":{"label":"docker1","provision-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":true},"zoscreds":{"username":"zosUsername","password":"zosPassword","private-key":"privateKey"}}' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/linux'
```

- c. If you need to deprovision from a Linux target environment, use the following REST API. After the deprovisioning is completed, the response displays 200 (OK) status code.

```
curl -X DELETE -k -u ztdadmin:password "https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/linux/your_provision_uuid?force=false&resume=false&preserve=false" -H "accept: application/json;charset=utf-8"
```

2. To provision instances to a Linux target environment by authenticating with a password or an ssh certificate key file, follow the steps:

- a. 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 }
```

- b. Create a provisioning or update a successful provisioning to a Linux target environment.

- If you create a new provisioning to a Linux target environment, specify the images and target environment that 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'
```

- If you need to update a successful provisioning with a new version of the image to a specific Linux target environment, specify the new version of the image, the image that is included in the provisioning, and target Linux environment. Put the token from the first call into the authorization header.

Three optional fields were added to specify the IPL parameters. Two fields were added to specify the z/OS user credential that is required to provision an instance from a created image 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": "cm9vdC0xMzZM5NzAzMjQ2", "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", "iplIODFAddress": "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'
```

- c. After you provision an instance to a target environment, run the following command to monitor the provisioning progress.

```
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'
```

- d. Run the following command to check the status of the provisioning.

```
curl -k -u ztdadmin: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 provisioning process is complete. If the provisioning fails, you can obtain the provisioning logs by completing the following steps:

- i) Enter the following command.

```
curl -k -u ztdadmin: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.

- ii) Extract the file to read the logs.

Provisioning instances to an OpenStack target environment

You can provision a z/OS instance that is running on an emulated IBM Z hardware to a specific OpenStack platform, or use the snapshot of an image to create a ZD&T instance in the OpenStack platform.

Procedure

1. Call the REST API by specifying the application and the OpenStack platform that you created.

```
curl -k -u ztdadmin: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": "ztdadmin", "cloudUser": "user_to_login_cloud", "password": "cloud_user_password", "scope": "project", "cp": 3, "ram": 16106127360, "doIPL": true, "deploymentDirectory": "/zdtee", "iplDeviceAddress": "0A80", "iplIODFAddress": "0A81", "iplLoadSuffix": "WS", "targetZosUsername": "TSOUSR1", "targetZosPassword": "TSOPASS1" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/deployCloud'
```

2. After you provision an instance to a target environment, run the following command to monitor the provisioning progress.

```
curl -k -u ztdadmin: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/monitor-services/getDeployStatus'
```

3. Run the following command to check the status of the provisioning.

```
curl -k -u ztdadmin: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 provisioning process is complete.

If the provisioning fails, you can obtain the provisioning logs by completing the following steps.

- a. Enter the following command.

```
curl -k -u ztdadmin: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.

4. To resume a failed or interrupted provisioning, specifying the application and the OpenStack 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": "/zdtee", "iplDeviceAddress": "0A80", "iplIODFAddress": "0A81", "iplLoadSuffix": "WS", "targetZosUsername": "TSOUSR1", "targetZosPassword": "TSOPASS1" }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtris/deployServices/deployCloud'
```

Then, use the step 2 to check the provisioning.

5. If you need to deprovision a OpenStack instance, run 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/zdtris/deployServices/deprovisionCloudInstance'
```

Provisioning instances to a Docker target environment

You can provision a z/OS instance that is running on emulated IBM Z hardware to a specific Docker target environment by authenticating with `ca.pem`, `cert.pem`, and `key.pem` certificate files.

Before you begin

Before you provision instances to a Docker environment, you must generate `ca.pem`, `cert.pem`, and `key.pem` certificate files. For the detailed instruction, see [Protect the Docker daemon socket](#).

Procedure

1. Call the REST API to provision an instance a Docker environment.

Note:

- General variables

If the `resume` is set to `false`, the parameters **application-uuid**, and **target-uuid** are the required. The parameter `provision-uuid` is mandatory only if `resume` is set to `true`; otherwise, the parameter is optional.

- Ports variables

If you want extra ports to be exposed on the Docker container, specify the ports. By default, the ports 21, 22, 23, 3270 and 8443 are exposed.

- To create a new provisioning to a Docker environment, run the following command.

```
curl -k -u zdtadmin:password -X POST -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{ "emulator": { "cp": 1, "ram": 3, "ziip": 0 }, "ipl": { "device-address": "a80", "iodf-address": "0a82", "load-suffix": "au" }, "general": { "label": "docker1", "target-uuid": "11111111-aaaa-bbbb-cccc-222222222222", "resume": false, "application-uuid": "11111111-aaaa-bbbb-cccc-222222222222" }, "zos-creds": { "username": "zosUsername", "password": "zosPassword", "private-key": "privateKey" }, "ports": [ 7000 ] }' 'https://your_tools_server_host_name:9443/ZDTREST/zdtris/provision-services/v1/docker'
```

- To resume a failed provisioning to a Docker environment, run the following command.

```
curl -k -u zdtadmin:password -X POST -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{ "emulator": { "cp": 1, "ram": 3, "ziip": 0 }, "ipl": { "device-address": "a80", "iodf-address": "0a82", "load-suffix": "au" }, "general": { "label": "docker1", "provision-uuid": "11111111-aaaa-bbbb-cccc-222222222222", "resume": true }, "zos-creds": { "username": "zosUsername", "password": "zosPassword", "private-key": "privateKey" }, "ports":
```

```
[7000]} 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/docker'
```

2. After you provision an instance to a Docker environment, run the following command to monitor the provisioning.

```
curl -k -u zdtadmin:password -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/monitor-services/v1/provision-status/{provision-uuid}'
```

Provisioning instances to a Red Hat OpenShift target environment

You can provision a z/OS instance that is running on an emulated IBM Z hardware to a specific Red Hat OpenShift target environment, or use a cloud-ready image to create a ZD&T instance in the Red Hat OpenShift environment.

Before you begin

Before you provision a z/OS instance that is running on an emulated IBM Z hardware, you need to create a Red Hat OpenShift target environment. For more information, see [“Creating and managing Red Hat OpenShift target environments”](#) on page 154.

Provisioning instances from a created image

After you create an image, you can provision a z/OS instance that is running on an emulated IBM Z hardware from the created image.

Procedure

1. Call the REST API to provision an instance from a created image to a Red Hat OpenShift environment.

Note:

- General variables

If the *resume* is set to false, the parameters **application-uuid**, and **target-uuid** are the required. The parameter **provision-uuid** is mandatory only if *resume* is set to true. Otherwise, the parameter is optional.

- Cloud-ready-image variable

The cloud-ready-image variable is used to indicate that if you want to create a cloud-ready image during the process of provisioning.

- To create a new provisioning to a Red Hat OpenShift environment, run the following command.

```
curl -X POST 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/openshift/zdt-image' -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{"emulator":{"cp":3,"ram":4000,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"label":"string","description":"string","target-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":false,"application-uuid":"11111111-aaaa-bbbb-cccc-222222222222"},"zos-creds":{"username":"zosUsername","password":"zosPassword","private-key":"string"},"cloud-ready-image":true}'
```

- To resume a failed provisioning to a Red Hat OpenShift environment, run the following command.

```
curl -X POST 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/openshift/zdt-image' -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{"emulator":{"cp":3,"ram":4000,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"provision-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":true},"zos-creds":{"username":"zosUsername","password":"zosPassword","private-key":"string"},"cloud-ready-image":true}'
```

2. After you provision an instance to a Red Hat OpenShift environment, run the following command to monitor the provisioning progress.

```
curl -k -u zdtadmin:password -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/monitor-services/v1/provision-status/{provision-uuid}'
```

Converting a created image to a cloud-ready image

To provision an instance from a cloud-ready image, you need to convert a created image to a cloud-ready image first.

Procedure

1. Call the REST API to provision an instance from a created image to a Red Hat OpenShift environment.

Note: If the *resume* is set to false, the parameters **application-uuid**, and **target-uuid** are the required. The parameter **provision-uuid** is mandatory only if *resume* is set to true. Otherwise, the parameter is optional.

- To create a new provisioning to a Red Hat OpenShift environment, run the following command.

```
curl -X POST 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deploy-services/v1/openshift/zdt-image' -H 'accept: application/json; charset=utf-8' -H 'Content-Type: application/json; charset=utf-8' -d '{"emulator":{"cp":3,"ram":4000,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"label":"string","description":"string","target-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":false,"application-uuid":"11111111-aaaa-bbbb-cccc-222222222222"},"zos-creds":{"username":"zosUsername","password":"zosPassword","private-key":"string"}}'
```

- To resume a failed provisioning to a Red Hat OpenShift environment, run the following command.

```
curl -X POST 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deploy-services/v1/openshift/zdt-image' -H 'accept: application/json; charset=utf-8' -H 'Content-Type: application/json; charset=utf-8' -d '{"emulator":{"cp":3,"ram":4000,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"provision-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":true},"zos-creds":{"username":"zosUsername","password":"zosPassword","private-key":"string"}}s {'
```

2. After you provision an instance to a Red Hat OpenShift environment, run the following command to monitor the provisioning progress.

```
curl -k -u zdtadmin:password -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/monitor-services/v1/provision-status/{provision-uuid}'
```

Provisioning instances from a cloud-ready image

After a cloud-ready image is created, you can provision a z/OS instance that is running on an emulated IBM Z hardware from the cloud-ready image.

Procedure

1. Call the REST API to provision an instance from a cloud-ready image to a Red Hat OpenShift environment.

Note:

- General variables

If the *resume* is set to false, the parameters **application-uuid**, and **target-uuid** are the required. The parameter **provision-uuid** is mandatory only if *resume* is set to true. Otherwise, the parameter is optional.

- pvc-name variable

The *pvc-name* variable indicates that the name of the Persistent Volume Claim that represents the cloud-ready image.

- To create a new provisioning to a Red Hat OpenShift environment, run the following command.

```
curl -X POST 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/openshift/cr-image' -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{"emulator":{"cp":3,"ram":4000,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"label":"string","description":"string","target-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":false},"zos-creds":{"username":"zosUsername","password":"zosPassword","private-key":"string"},"pvc-name":"name_of_cloud_ready_image_pvc"}'
```

- To resume a failed provisioning to a Red Hat OpenShift environment, run the following command.

```
curl -X POST 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-services/v1/openshift/cr-image' -H 'accept: application/json;charset=utf-8' -H 'Content-Type: application/json;charset=utf-8' -d '{"emulator":{"cp":3,"ram":4000,"ziip":0},"ipl":{"device-address":"a80","iodf-address":"0a82","load-suffix":"au"},"general":{"provision-uuid":"11111111-aaaa-bbbb-cccc-222222222222","resume":true},"zos-creds":{"username":"zosUsername","password":"zosPassword","private-key":"string"},"pvc-name":"name_of_cloud_ready_image_pvc"}'
```

2. After you provision an instance to a Red Hat OpenShift environment, run the following command to monitor the provisioning progress.

```
curl -k -u zdtadmin:password -X GET --header 'Content-Type: application/json' --header 'Accept: application/json' 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/monitor-services/v1/provision-status/{provision-uuid}'
```

ZD&T response error codes

If an error occurs when you use REST APIs, refer to the response error codes for detailed description.

Note: The italic text such as *{0}*, *{1}*, *{2}* indicates that they are the variables, the variables will be replaced by the contents that are displayed in the messages.

Error ID	Description
403	This account is locked out for <i>{0}</i> minutes.
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 storage server already exists.
10006	The storage server does not exist.
10007	The Licensing system <i>{0}</i> already exists.
10008	The Licensing system <i>{0}</i> 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.
10014	The application image does not exist.
10015	The component already exists.

Table 28. Response error codes (continued)

Error ID	Description
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 {0} is not reachable.
10028	Create entity {0} failed.
10029	Retrieve entity {0} failed.
10030	Update entity {0} failed.
10031	Delete entity {0} failed.
10101	Failed to connect to the remote host {0} on port {1}.
10102	Token expired.
10200	Failed to connect to the storage server {0} on port {1}.
10201	Connection refused.
10202	Invalid user ID or password.
10203	Invalid base directory {0} specified for server {1} on port {2}.
10204	Invalid hostname for server {0}.
10205	Connection closed without indication.
10206	Failed to set the file transfer type.
10207	A failure occurred when storing content on the storage server.
10208	A failure occurred when retrieving content from the storage server.
10209	Failed to create directory {0} on the storage server.
10210	Unable to access the directory {0} on the storage server.
10211	Failed to delete the file {0} from the image storage server.
10213	Internal Error.
10214	The old base directory is needed.
10215	This feature is supported only for SFTP storage system.
10216	This source system cannot be deleted because there exists components or applications related to it.
10300	Failed to connect to the IBM Application Discovery server {0} on port {1}.
11001	Installation files download failed.
11003	Deploy request parameter not passed.
11007	Invalid Linux platform of target system.
11011	zPDT installation script error: Parameters invalid.
11012	zPDT installation script error: Primary licensing server is required.

Table 28. Response error codes (continued)

Error ID	Description
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 startup message not found.
11026	Deploy script error: Parameter device number required.
11027	Deploy script error: Z is not on.
11028	Application images cannot be deployed to the same system that is running the IBM Z Development and Test Environment Enterprise Edition web server.
11029	Target server is not available for deployment.
11030	Failed to create the directory {0} on the target system.
11033	IBM Z Development and Test Environment did not start 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 to be deployed.
11101	Changing the owner of volume image files failed.
11102	Executing command {0} on target environment {1} 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.
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: {0}.

Table 28. Response error codes (continued)

Error ID	Description
11112	Updating the emulator is not authorized.
11113	Downloading the z-volume files failed.
11114	Unzipping the z-volume files failed.
11115	User has no root or sudo access, please change the target system settings to bypass the emulator installation.
11116	User has no root or sudo access, please change the target system settings to bypass the dependencies installation.
11117	User has no root or sudo access, please change the target system settings to bypass the network configuration.
11118	User has no root or sudo access, please change the target system settings to bypass the user configuration.
11119	User has no root or sudo access, please change the target system settings to bypass the user configuration.
12000	A failure occurred when reading the devmap file: {0}.
12001	A failure occurred when parsing the devmap file.
12002	A failure occurred when writing the devmap file.
12003	Create application failed.
12004	The devmap file does not exist in the specified location: {0}.
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.
12008	The compression directory does not exist in the specified location: {0}.
12009	The user {0} does not have write access to the compression directory {1}.
12010	Should not specify the compression directory {0} because the compression option is off.
20001	Authorization token has expired.
20002	Authorization token was not supplied.
20003	User already exists.
20004	User does not exist.
21001	The machine running the ZD&T Enterprise Edition server experienced a network error.
21002	The ssh command on {0} failed to be executed.
21003	The sftp command on {0} failed to be executed.
21004	Remote command execution at {0} has been running for too long.
21005	Sending/receiving files to/from {0} failed when {1}.
21006	The target environment TCP port routing rules must contain 2022>22.
21007	The target environment port routing rules have syntax error.

Table 28. Response error codes (continued)

Error ID	Description
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 component schedule date is invalid.
30009	The component schedule time is invalid.
30010	The component schedule date is required.
30011	The component schedule time is required.
30012	The component schedule time zone is invalid.
30013	The component schedule time zone is required.
30014	Only one component that contains the system residence volume can be selected.
30015	The extraction cannot be resumed as the application image is already extracted.
30016	Only an application created from an IBM Z mainframe supports extraction being resumed or stopped.
30017	The number of simultaneous volume extractions cannot be less than 1.
30018	The extraction cannot be stopped as the application is not being extracted.
30019	The z/OS user does not match the user who created the application.
30020	Component name is required.
30021	Component name is invalid.
30022	Volumes are required.
30023	Volumes are invalid.
30024	A data set name filter string is required.
30025	The data set name filter string cannot be * or **.
30026	The selected data set list is required.
30027	Only z/OS components can be deleted.
30028	This application component is currently being deleted.
30029	This application component cannot be deleted as it is being used by the application image {0}.
30030	This application component is currently being extracted.
30031	The data set name filter string contains invalid characters.

Table 28. Response error codes (continued)

Error ID	Description
30032	The application component type cannot be changed from {0} to {1}.
30033	Only application components containing volumes can have their type changed.
30034	The version of the component is invalid.
30035	Only components created from an IBM Z mainframe support editing.
30036	The same volume or data set cannot be processed in multiple requests.
30037	The extraction cannot be stopped as the component is not being extracted.
30038	The extraction cannot be resumed as the component is already extracted.
30039	Creating a new version or duplication cannot occur at the same time.
30040	Deploy preference is required.
30041	Deploy preference is invalid.
30042	The image storage information is required to be defined on the <a href
30043	Component is required.
30044	The emulator licensing information is required to be defined on the <a href
30045	Target environment is required.
30046	Number of <abbr title.
30047	Memory size is invalid.
30048	Deployment label is required.
30049	Not enough <abbr title
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 log in the cloud is required.
30055	The password of the user to log in 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.
30061	The request to validation an existing ZD&T image failed due to missing required input.
30063	The compression directory is required.
30064	The comment is longer than the allotted maximum of 225 characters.
30065	The component's system type is invalid.
30066	You cannot create {0} component on the source system of type {1}.

Table 28. Response error codes (continued)

Error ID	Description
30067	Valid z/OS system information is required.
30068	Please login {0} instead of {1}.
30080	Target environment is required.
30081	Target environment is invalid.
30082	A source system is required.
30083	The source system is invalid.
30084	A storage server is required.
30085	The storage server is invalid.
30086	The IBM Application Discovery system is required.
30087	The IBM Application Discovery system information is invalid or empty.
30088	The port is invalid.
30089	A 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	A User ID is required.
30109	Password is required.
30110	Your password has expired.
30111	Target environment label is required.

Table 28. 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 {0} failed.
30115	The file size cannot exceed 50KB.
30116	All IPL parameters need 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	Storage server hostname is required.
30125	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.
30152	The UUID parameter for this API is required.
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 {0}.
31003	This application image is being deprovisioned from {0}, please wait for it to finish.
31004	This application image is being deployed to {0} but failed, please confirm to delete it.
31005	This selected application component {0} is not in this application.
31006	This selected application component {0} is found in other applications as shown below: {1}.
31007	This application image is being deleted.
31008	This application image is not available.

Table 28. 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 {0} 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: {0}.
31013	Only an ADCD or Z system application needs to be migrated.
31014	This {0} does not need migration.
31015	This {0} 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 {0}.
31022	This application image contains duplicate data set {0}.
31023	This ADCD version doesn't support including data set 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 {1} is found in schema {0}.
31029	Duplicate database name {0} is found.
31030	Only one ADCD Db2 component can be selected.
31031	The version of Db2 from component {0} does not match the version of ADCD Db2 component {1}.
31032	The Db2 files failed to be transferred.
31033	The specific ADCD package could not be found.
31034	The image of the specific ADCD package had been deployed.
31035	The SYSRES component {0} needs to be migrated.

Table 28. Response error codes (continued)

Error ID	Description
31050	Wait for the application component {0} to finish processing.
31051	The application component {0} has been deleted.
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 {0}.
31059	This component doesn't need to be migrated.
31100	The related records for the given application image and target environment cannot be found.
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 {0} 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	The application image cannot be deployed to cloud by using this REST API.
31111	A target cloud instance cannot be deprovisioned while it is being provisioned.
31112	A target cloud instance cannot be deprovisioned while it is being deployed.
31113	Deprovision of {0} from target cloud platform {1} failed with error return code {2}.
31114	A target cloud instance cannot be deprovisioned 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 log in to z/OS for deploying {0}.
31117	The password for user {0} to log in to z/OS {1} has expired and needs to be changed.
31118	Cannot deploy to a target environment that is being deprovisioned.

Table 28. Response error codes (continued)

Error ID	Description
31119	This application cannot be deployed because its data set component {0} has not been extracted.
31120	The user ID must not be longer than 8 characters.
31121	Authentication failed.
31122	The Linux group zpdt does not exist.
31123	TheLinux user ID ibmsys1 does not exist.
31124	Unsupported platform.
31125	{0} is not a supported CPU architecture.
31126	The user performing the deployment does not belong to Linux group zpdt.
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 {0} could not resolve to a valid IP address from the system {1}.
31130	The connection to hostname {0} on port {1} from the system {2} failed.
31131	The deployment directory {0} does not exist on the target environment.
31132	Write access is required to the deployment directory {0}.
31133	The deployment directory does not have enough available disk space.
31143	The required software 'ftp' is not installed on the system {0}.
31144	The required 32-bit library 'stdlib' is not installed on the system {0}.
31145	The required software 'perl' is not installed on the system {0}.
31146	The required software 'zip' is not installed on the system {0}.
31147	The required software 'unzip' is not installed on the system {0}.
31148	The required software 'gzip' is not installed on the system {0}.
31149	The required software 'bc' is not installed on the system {0}.
31150	The required library 'libasound2' is not installed on the system {0}.
31151	You do not have permission to run the command {0} on the system {1}.
31152	You do not have permission to change the owner of file {0} on the system {1}.
31153	Failed to list all of the privilege management commands.
31154	{0} is not a supported Linux distribution.
31155	The IBM ZD&T emulator needs to be running in order to deploy data sets.
31159	Cannot run command 'nc' to validate the connectivity to hostname {0} on port {1} from the system {2}.
31160	Cannot run command 'getent' to resolve hostname {0} from the system {1}.
31162	This application cannot be deployed because it contains component {0} that needs to be migrated.

Table 28. Response error codes (continued)

Error ID	Description
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 {0} with version {1} is not found to be deployed on host {2}.
31169	Reading devmap failed.
31170	No spare 3390 volume available for restoring data set.
31171	Volume cannot be found at {0}.
31172	The emulator information cannot be retrieved from the target environment {0}.
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 {0}, with output {1}.
31302	An error occurred when allocating a file when executing command {0}, with output {1}.
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 {0} with output {1}.
31307	The z/OS TCP/IP settings cannot be acquired when executing command {0}, with output {1}.
31308	No SYSPRINT/stdout when executing command {0}, with output {1}.
31309	The load library for the z/OS extraction tool is not APF-authorized.
31310	READ access is not granted to the SAF profile in the class DASDVOL.
31311	The Volume image file cannot be opened: {0}.
31312	Compression initialization failed - internal error with output {0}.
31313	Cannot open the volume - internal error without {0}.
31314	Command {0} invoked cannot execute.
31315	Linux command {0} was not found.
31316	Invalid argument given to the exit statement when executing command {0}.
31317	Fatal error signal with return code {0} when executing command {1}.

Table 28. Response error codes (continued)

Error ID	Description
31318	Data sets allocated failed.
31319	Volume {0} extraction failed.
31320	Failed to list data set.
31321	A failure occurred while retrieving IBM Z mainframe information, return code: {0}
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 mainframe with the following output.
31325	These data sets cannot be extracted because your user ID has insufficient access to read them: {0}.
31326	These data sets cannot be extracted because their containing volumes cannot be found: {0}.
31327	These data sets cannot be extracted because of unknown errors: {0}.
31328	APF authorization failed.
31329	The CICS region information is required.
31330	The CICS region name is required.
31331	The CICS region {0} was already defined.
31332	The CICS region {0} was not found.
31333	The LOADLIB of REXX script in command {0} was not found.
31334	Parsing CICS CSD got format error.
31335	You must select some GROUPs defined in the CICS CSD.
31336	Got unexpected exception when trying to retrieve CICS CSD definition for {0}.
31337	The input LIST name {0} is not a valid CSD LIST name in {1}.
31338	The input GROUP name {0} is not a valid CSD GROUP name in {1}.
31339	Components are created using the CICS region {0}. Need to delete those CICS components first.
31340	Vary Online ZD&T created volumes failed with error code {0}.
31341	The Db2 subsystem information is required.
31342	The Db2 subsystem ID is required.
31343	The Db2 subsystem ID {0} was already defined.
31344	The Db2 subsystem ID {0} 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.

Table 28. Response error codes (continued)

Error ID	Description
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.
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 are set to \$1.
31361	Insufficient parameters or error reading parameter file.
31362	No tables were found in the table lists 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.
31370	Db2 subsystem is in use.
31371	Unable to connect to the Db2 subsystem with the given parameters.
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.
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.

Table 28. Response error codes (continued)

Error ID	Description
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.
31654	The image is not active.
31655	The image is not found.
31656	The flavor is not public accessed.
31657	The flavor is not found.
31700	The target environment could not be deleted as an application image is deployed to it.
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 <i><a href</i>).
31752	ID is invalid.
31753	Privilege command option does not exist.
31754	Cannot find network interfaces from server {0}:{1}.
31755	The deployment uuid is required.
31756	The deployment does not exist.
31757	The deployment has not finished yet.
50010	Configuring TCPIP on the wazi-sandbox container failed.
50011	Invalid original z/OS IP address.
50012	Invalid TCP volume request parameter.
50013	Invalid data set name.
50014	Invalid TCP member name.
50015	Invalid z/OS IP.
50016	Invalid gateway IP.
50017	Configuring ADCD IPL on the wazi-sandbox container failed.
50018	Invalid RSU request parameter.
50019	Invalid ADCD IPL request parameter.
50026	Querying license on the wazi-sandbox container failed.
50027	Generating the parm file for the IEASYM script failed.
50028	File name for IEASYM parm file is either invalid or empty.
50029	Symbolic information for IEASYM parm file is either invalid or empty.
50031	The licensing server request is required.

Table 28. Response error codes (continued)

Error ID	Description
50032	The licensing server hostname is required.
50033	The licensing server UUID is required.
50034	The token server request is required.
50035	The token server hostname is required.
50036	The token server UUID is required.
50037	The UUID is of an invalid format.
50038	A label is required.
50039	The label must be under 44 characters long.
50040	The label is already in use by another system.
50041	The label cannot be empty.
50042	There already exists a primary licensing system.
50081	Running command of getting the number of ZPD files failed.
50082	Decryption is not done yet in the container.
50100	The application UUID is required.
50101	The application UUID is invalid.
50102	The target system UUID is required.
50103	The target system UUID is invalid.
50104	The provision UUID is required.
50105	The provision UUID is invalid.
50106	The user name is more than 8 characters.
50107	The general properties are required.
50108	The credential of the target z/OS is required.
50109	The emulator properties are required.
50110	The IPL properties are required.
50111	The provision request string is invalid.
50112	The target system does not exist.
50113	The provision object does not exist.
50114	The list of ports contains invalid value.
50115	Failed to read the content of <code>/etc/resolve</code> .
50116	Failed to read the content of <code>/etc/hosts</code> file in container.
50117	Could not resume provision for <code>{0}</code> because it is not a failed provision.
50118	The <code>{0}</code> cannot be deprovisioned with status <code>{1}</code> .
50119	The target system cannot exceed the maximum number of allowed provisions.
50120	The number of containers that a target system has must be between 1 and 5.

Table 28. Response error codes (continued)

Error ID	Description
50121	Failed to send temp file to storage server location {0}.
50122	Failed to read the content of /zdt/zsystem/FEUCUTIL file in container.
500128	Failed to create PVC {0}.
500129	Failed to create custom resource of volume copy {0}.
500130	Container does not exist for the given provision UUID {0} and container ID {1}.
50900	Unable to establish an SSH connection to the SSH service listening on port {1} of host {0}.
50920	Unable to establish an HTTPS connection to the Docker daemon listening on port {1} of host {0}.
50940	Unable to establish an HTTPS connection to the ZD&T instance controller service listening on port {1} of host {0}.
50941	A failure occurred while executing an Instance Controller API.
50942	An unsupported Instance Controller HTTP method was called.
50943	An attempt was made to call an Instance Controller API with invalid credentials (HTTP status code 401).
50944	An attempt was made to call an Instance Controller API for which the user is not authorized to access (HTTP status code 403).
50945	An attempt was made to call an Instance Controller API that does not exist (HTTP status code 404).
50946	An attempt was made to call an Instance Controller API with an incorrect HTTP method (HTTP status code 405).
50947	An attempt was made to call an Instance Controller API that is not capable of generating a response for the request Accept headers (HTTP status code 406).
50948	An attempt was made to call an Instance Controller API using a media type that is not supported by this API (HTTP status code 415).
50949	An attempt was made to call an Instance Controller API due to a conflict (HTTP status code 409).
50950	An attempt was made to call an Instance Controller API that no longer exists (HTTP status code 410).
50951	An attempt was made to call an Instance Controller API and caused an unknown error.
50980	An oprmsg command failed.
50981	The requested oprmsg command is invalid.
50982	The requested oprmsg command is empty.
60000	This target environment has not been provisioned.
60001	This entity {0} is being used by another user, please retry after some time.
60215	The cloning option is disabled on target environment {0}.
60218	Failed to clone Persistent Volume Claim {0} in namespace {1}.
60220	Failed to create ConfigMap {0} in namespace {1}.

Table 28. Response error codes (continued)

Error ID	Description
60221	Failed to create Secret {0} in namespace {1}.
60222	Failed to delete ConfigMap {0} in namespace {1}.
60223	Failed to delete Secret {0} in namespace {1}.
60224	Failed to delete Persistent Volume Claim {0} in namespace {1}.
60225	Failed to update Persistent Volume Claim {0} in namespace {1}.
60228	Persistent Volume Claim {0} already exists in namespace {1}.
70001	The ZD&T extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS returned an unexpected result.
70991	Checking {0} version for the ZD&T extraction utility SFTP module zDTMainframeSFTP.
70992	A Java 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 version.
70999	Internal error acquiring information from z/OS.
79999	An error occurred when connecting to z/OS: {0}.
99998	Unexpected exception.
99999	Unknown error.

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 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.
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.

Table 30. Troubleshooting the components creation from IBM Application Discovery projects (continued)

Symptom	Problem Determination
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 AD integrations” on page 128](#).
3. Create the component from IBM Application Discovery, and select the project that is built to analyze the CICS startup JCL. For more information, see [“Creating an AD component” on page 132](#).
4. Create the image. For more information, see [“Creating z/OS images” on page 134](#).
5. Provision an instance from a created image to the target environment. For more information, see [“Provisioning instances to target environments” on page 136](#).

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, LPAALSTCI 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.

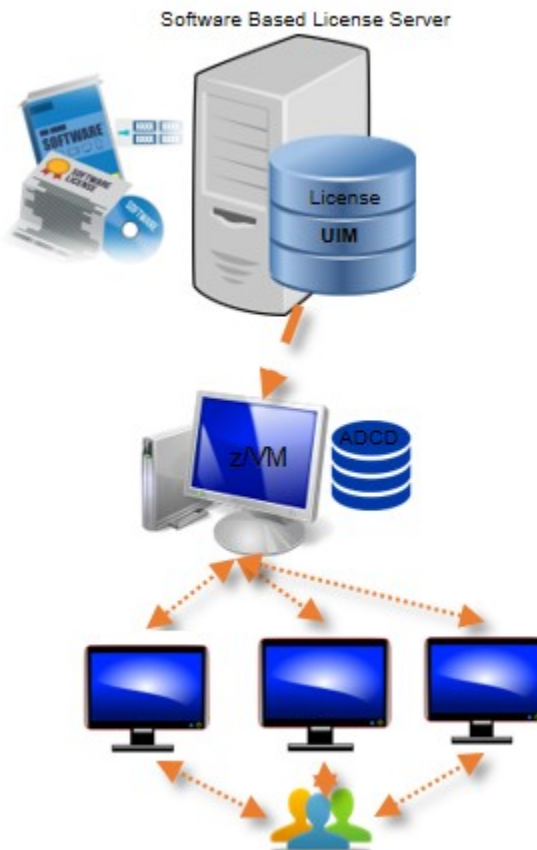


Figure 12. The Software-based License Server and various components of Parallel Sysplex

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 13. 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.

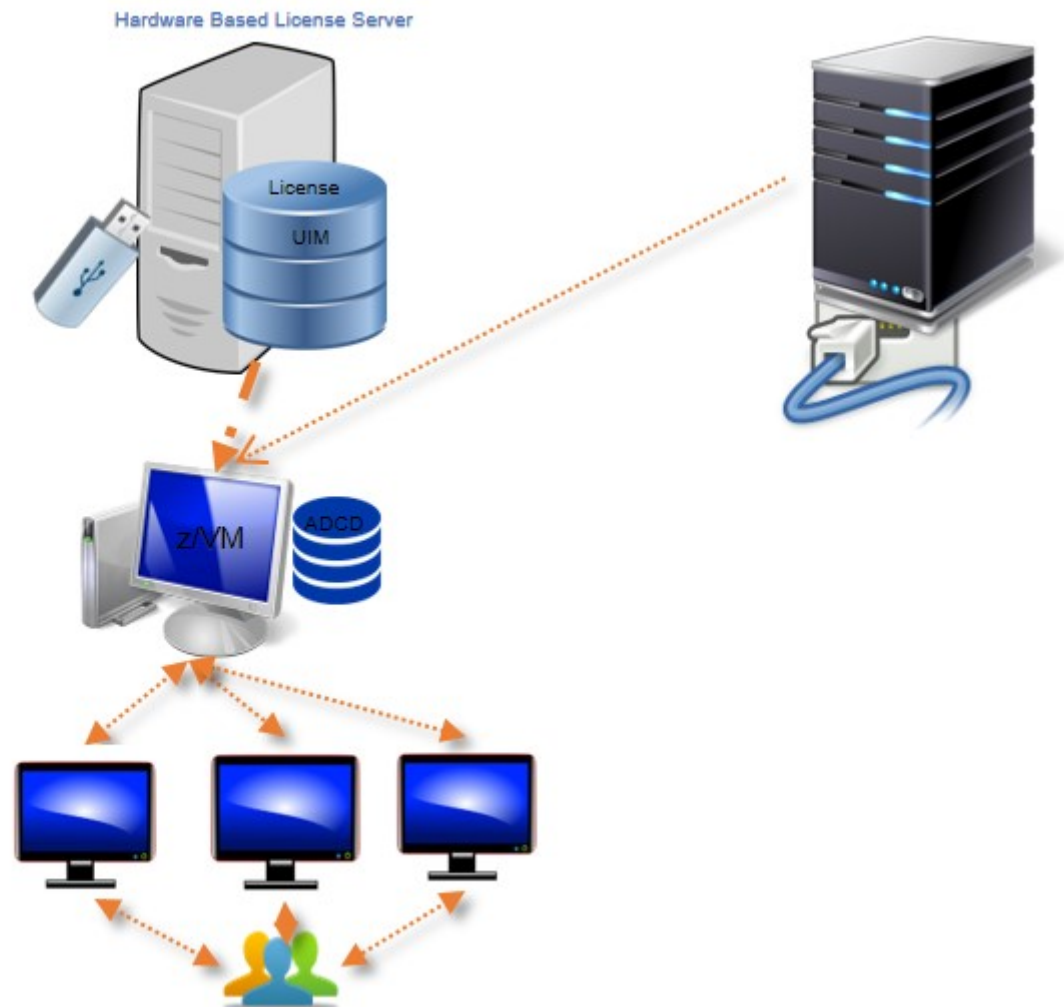


Figure 14. The Hardware-based License Server with Rational Tokens, and various components of Parallel Sysplex

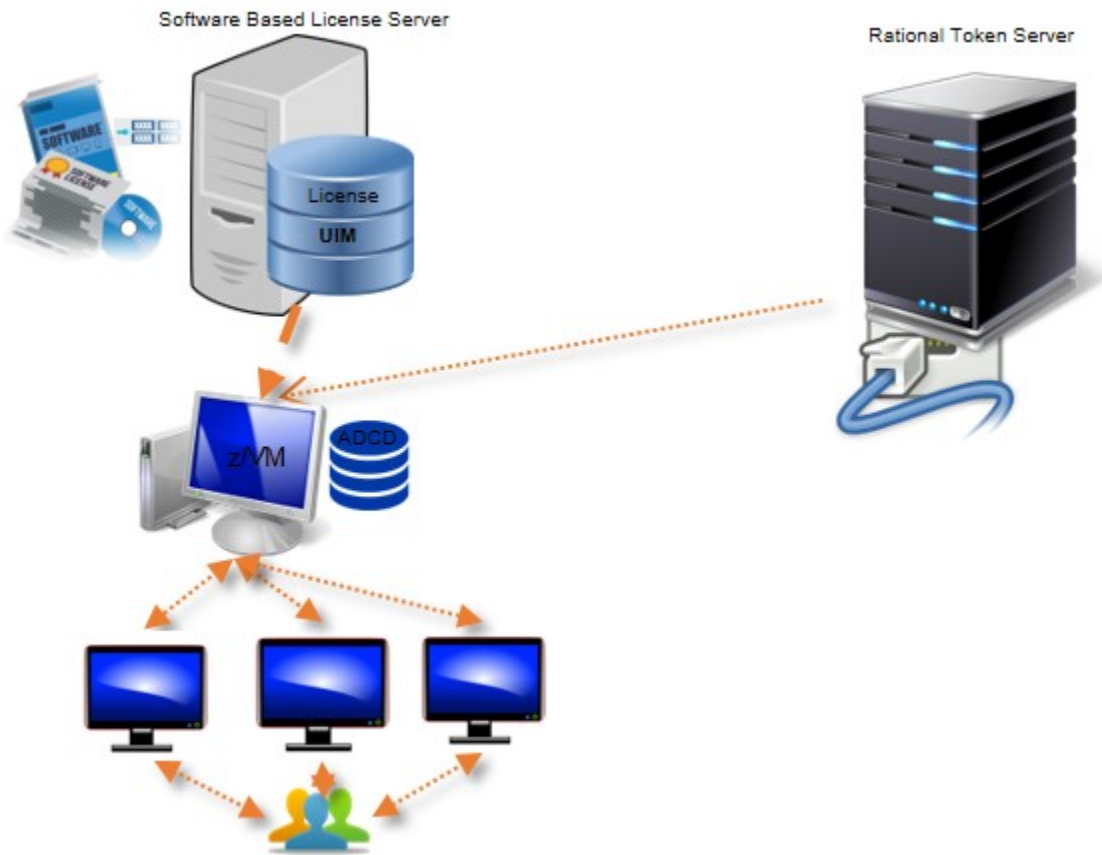


Figure 15. The Software-based License Server with Rational Tokens, and various components of Parallel Sysplex

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 [Products contained in this release](#).

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 31. New installation checklist for Parallel Sysplex

Installing Parallel Sysplex		Required/ Optional	Complete
1	<p>“Prerequisites” on page 197</p> <p>Describes the software and hardware requirements, and the steps to set up the basic Linux system.</p>	Required	<input type="checkbox"/>

<i>Table 31. New installation checklist for Parallel Sysplex (continued)</i>			
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 servers” on page 202</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 218</u> Describes the steps to install Parallel Sysplex.	Required	<input type="checkbox"/>
5	<u>“Activating ZD&T Parallel Sysplex instances” on page 219</u> Describes the steps to activate ZD&T instances	Required	<input type="checkbox"/>
6	<u>“Configuring Z instances” on page 221</u> Describes the steps to configure ADCD to start Z instances.	Required	<input type="checkbox"/>
7	<u>“Starting z/VM” on page 223</u> Describes the steps to start z/VM.	Required	<input type="checkbox"/>
8	<u>“Starting Z instances” on page 224</u> Describes the steps to start Z instances.	Required	<input type="checkbox"/>
9	<u>“Stopping Z instances and z/VM” on page 225</u> Describes the steps to stop Z and z/VM.	Optional	<input type="checkbox"/>

<i>Table 32. Update checklist for Parallel Sysplex</i>			
Updating Parallel Sysplex		Required/ Optional	Complete
1	<u>“Returning a software-based license key” on page 207 or “Returning a hardware-based license key” on page 216</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 197</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 32. 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 227. 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 226 and “Installing Parallel Sysplex” on page 218. 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 servers” on page 202 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 219 Describes the steps to activate ZD&T instances	Required	<input type="checkbox"/>
7	“Configuring Z instances” on page 221 Describes the steps to configure ADCD to start Z instances.	Required	<input type="checkbox"/>
8	“Starting z/VM” on page 223 Describes the steps to start z/VM.	Required	<input type="checkbox"/>
9	“Starting Z instances” on page 224 Describes the steps to start Z instances.	Required	<input type="checkbox"/>

Table 33. Update checklist for license server

Updating license server		Required/ Optional	Complete
1	“Returning a software-based license key” on page 207 or “Returning a hardware-based license key” on page 216 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 210 or “Updating the Hardware-based License Server” on page 217 Describes the steps to update the Software-based License Server or Hardware-based License Server.	Required	<input type="checkbox"/>
3	“Enabling the license servers” on page 202 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 34. Update checklist for license</i>			
Updating license		Required/Optional	Complete
1	“Returning a software-based license key” on page 207 or “Returning a hardware-based license key” on page 216 Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	<input type="checkbox"/>
2	“Enabling the license servers” on page 202 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 35. Uninstallation checklist for Parallel Sysplex</i>			
Uninstalling Parallel Sysplex		Required/Optional	Complete
1	“Returning a software-based license key” on page 207 or “Returning a hardware-based license key” on page 216 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 211 or “Uninstalling the Hardware-based License Server” on page 218 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 228 Describes the steps to uninstall Parallel Sysplex.	Required	<input type="checkbox"/>

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.	Image Classification	File Name
1	IBM Z Development and Test Environment Parallel Sysplex V13.3.0 Installation Multilingual	M03Y1ML	Required	ZDT_Install_PS_V13.3.0.tgz
2	IBM Z Development and Test Environment Parallel Sysplex V13.3.0 Installation checksum	M03Y2ML	Required	ps.md5
3	z/VM 7.1 #1 of 5 - RES volume	G00PPZX	Required	M01RES.gz
4	z/VM 7.1 #2 of 5 – Page volume	G00PQZX	Required	M01P01.gz
5	z/VM 7.1 #3 of 5 – System volume	G00PRZX	Required	M01S01.gz
6	z/VM 7.1 #4 of 5 – Common volume	G00PSZX	Required	VMCOM1.gz
7	z/VM 7.1 #5 of 5 – Release volume	G00PTZX	Required	710RL1.gz
8	z/VM 7.1 - Checksum	G00PVZX	Required	adcd_zvm71_checksum.txt
9	RLKS V8.1.5 for Windows ML	CNIV9ML	Optional	N/A
10	LKAD V8.1.5 for Windows ML	CNIW0ML	Optional	N/A
11	RLKS 8.1.5 for Linux x86 ML	CNIW1ML	Optional	N/A
12	RLKS 8.1.5 for Solaris SPARC ML	CNIW2ML	Optional	N/A
13	RLKS V8.1.5 for Solaris x86 ML	CNIW3ML	Optional	N/A
14	RLKS V8.1.5 for HP IA64 ML	CNIW4ML	Optional	N/A

	Name	Part No.	Image Classification	File Name
15	RLKS V8.1.5 for Linux PPC ML	CNIW5ML	Optional	N/A
16	RLKS V8.1.5 for AIX ML	CNIW6ML	Optional	N/A
17	RLKS Administrator And Reporting Tool Setup ML	CNIW7ML	Optional	N/A
18	RLKS Administrator And Reporting Tool - Part 1 ML	CNIW8ML	Optional	N/A
19	RLKS Administrator And Reporting Tool - Part 2 ML	CNIW9ML	Optional	N/A
20	ADCD z/OS V2.4 for IBM Z Development and Test Environment 13.3 RSU 2103 Multilingual eAssembly	G050QML	Optional, ADCD May 2021 Edition RSU level 2103	N/A
21	z/OS 2.4 Part 1 of 19 - RES volume 1 Multilingual	M03WTML	Optional, ADCD May 2021 Edition RSU level 2103	D4RES1.ZPD
22	z/OS 2.4 Part 2 of 19 - RES volume 2 Multilingual	M03WVML	Optional, ADCD May 2021 Edition RSU level 2103	D4RES2.gz
23	z/OS 2.4 Part 3 of 19 - System volume Multilingual	M03WVML	Optional, ADCD May 2021 Edition RSU level 2103	D4SYS1.gz
24	z/OS 2.4 Part 4 of 19 - Configuration volume Multilingual	M03WXML	Optional, ADCD May 2021 Edition RSU level 2103	D4CFG1.gz
25	z/OS 2.4 Part 5 of 19 - UNIX System Services volume 1 Multilingual	M03WYML	Optional, ADCD May 2021 Edition RSU level 2103	D4USS1.gz
26	z/OS 2.4 Part 6 of 19 - UNIX System Services volume 2 Multilingual	M03WZML	Optional, ADCD May 2021 Edition RSU level 2103	D4USS2.gz
27	z/OS 2.4 Part 7 of 19 - z/OS Products volume 1 Multilingual	M03X0ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD1.gz
28	z/OS 2.4 Part 8 of 19 - z/OS Products volume 2 Multilingual	M03X1ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD2.gz
29	z/OS 2.4 Part 9 of 19 - z/OS Products volume 3 Multilingual	M03X2ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRD3.gz
30	z/OS 2.4 Part 10 of 19 - z/OS Products volume 4 Multilingual	M03X3ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PRC4.gz
31	z/OS 2.4 Part 11 of 19 - z/OS Distribution Libraries volume 1 Multilingual	M03X4ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS1.gz
32	z/OS 2.4 Part 12 of 19 - z/OS Distribution Libraries volume 2 Multilingual	M03X5ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS2.gz
33	z/OS 2.4 Part 13 of 19 - z/OS Distribution Volume Multilingual	M03X6ML	Optional, ADCD May 2021 Edition RSU level 2103	D4DIS3.gz

	Name	Part No.	Image Classification	File Name
34	z/OS 2.4 Part 14 of 19 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M03X7ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGA.gz
35	z/OS 2.4 Part 15 of 19 - z/OS Page Datasets (Local) volume 2 Multilingual	M03X8ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGB.gz
36	z/OS 2.4 Part 16 of 19 - z/OS Page Datasets (Local) volume 3 Multilingual	M03X9ML	Optional, ADCD May 2021 Edition RSU level 2103	D4PAGC.gz
37	z/OS 2.4 Part 17 of 19 - z/OS Page Datasets (Local) volume 4 Multilingual	M03XBML	Optional, ADCD May 2021 Edition RSU level 2103	D4USR1.gz
38	z/OS 2.4 Part 18 of 19 - Stand Alone RES volume Multilingual	M03XCML	Optional, ADCD May 2021 Edition RSU level 2103	SARES1.ZPD
39	z/OS 2.4 Part 19 of 19 – zCX Container volume	M03XDML	Optional, ADCD May 2021 Edition RSU level 2103	D4ZCX1.gz
40	Engineering Workflow Management 7.0.1 Multilingual	M03XFML	Optional, ADCD May 2021 Edition RSU level 2103	D4BLZ1.gz
41	DB2 V12 Part 1 of 2 Multilingual	M03XGML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBC1.gz
42	DB2 V12 Part 2 of 2 Multilingual	M03XHML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBC2.gz
43	DB2 V11 Part 1 of 2 Multilingual	M03XJML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBB1.gz
44	DB2 V11 Part 2 of 2 Multilingual	M03XKML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBB2.gz
45	CICS 5.6 Multilingual	M03XLML	Optional, ADCD May 2021 Edition RSU level 2103	D4C560.gz
46	CICS 5.5 Multilingual	M03XMM L	Optional, ADCD May 2021 Edition RSU level 2103	D4C551.gz
47	IMS 14.1 Multilingual	M03XNML	Optional, ADCD May 2021 Edition RSU level 2103	D4IME1.gz
48	IMS 15.2 Multilingual	M03XPML	Optional, ADCD May 2021 Edition RSU level 2103	D4IMF1.gz
49	IBM Installation Manager 1.9 Multilingual	M03XQML	Optional, ADCD May 2021 Edition RSU level 2103	D4INM1.gz
50	DB2 Archive Logs Multilingual	M03XRML	Optional, ADCD May 2021 Edition RSU level 2103	D4DBAR.gz
51	Tivoli Omegamon XE products 5.5.1 Multilingual	M03XSML	Optional, ADCD May 2021 Edition RSU level 2103	D4KAN1.gz
52	WAS V9.0 Part 1 of 2 Multilingual	M03XTML	Optional, ADCD May 2021 Edition RSU level 2103	D4W901.gz
53	WAS V9.0 Part 2 of 2 Multilingual	M03XVML	Optional, ADCD May 2021 Edition RSU level 2103	D4W902.gz

	Name	Part No.	Image Classification	File Name
54	Zowe Open Source Project Multilingual	M03XWML	Optional, ADCD May 2021 Edition RSU level 2103	D4ZWE1.gz
55	ADCD Metadata for Tools Multilingual	M03XXML	Optional, ADCD May 2021 Edition RSU level 2103	ADCDTOOLS.XML
56	Extended ADCD Checksum Multilingual	M03XYML	Optional, ADCD package MD5 checksum	may2021_adcd_md5.txt

Validating the downloaded installer file

ZD&T installer now supports the signing of the installer. With the signed installer, you can validate the completeness of the downloaded package by running a command with the signing certificate that is packaged in the installer.

ZD&T Parallel Sysplex installer file ZDT_Install_PS_V13.3.0.tgz that you download from IBM is a compressed file. The installer file contains the installer ZDT_Install_PS_V13.3.0.x86_64 and the corresponding signature file ZDT_Install_PS_V13.3.0.x86_64.sig.

To validate the signature file for the installer, run the following command.

```
oPSnssl dgst -sha256 -verify <PUBLICKEYNAME> -signature <signature_file> <FILETOSIGN>
```

For example, run the following command from the directory that contains the extracted installer files.

```
oPSnssl dgst -sha256 -verify signature/public.PSm -signature ZDT_Install_PS_V13.3.0.x86_64.sig ZDT_Install_PS_V13.3.0.x86_64
```

If the installation file is valid, you will receive the message of Verified OK.

Enabling the license servers

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.

Software-based license		Required/Optional	Complete
1	“Installing the Software-based License Server” on page 204 Describes the steps to install the Software-based License Server	Required	<input type="checkbox"/>
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 37. Enablement checklist for hardware-based licensing

Hardware-based license		Required/ Optional	Complete
1	<p>“Installing the Hardware-based License Server” on page 212</p> <p>Describes the steps to install the Hardware-based License Server.</p>	Required	<input type="checkbox"/>
2	<p>Obtaining an update file for Hardware-based License Server</p> <p>Describes the steps to obtain an update file for Hardware-based License Server.</p>	Required	<input type="checkbox"/>
3	<p>“Applying the update file to the Hardware-based License Server” on page 215</p> <p>Describes the steps to apply the update file to ZD&T Parallel Sysplex.</p>	Required	<input type="checkbox"/>

Table 38. Switching from Hardware-based License Server to Software-based License Server

Hardware-based license to Software-based license		Required/ Optional	Complete
1	<p>“Returning a hardware-based license key” on page 216</p> <p>Describes the steps to return a hardware-based license key.</p>	Required	<input type="checkbox"/>
2	<p>“Uninstalling the Hardware-based License Server” on page 218</p> <p>Describes the steps to uninstall Hardware-based License Server.</p>	Required	<input type="checkbox"/>
3	<p>“Installing the Software-based License Server” on page 204</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 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_V13.3.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PS_V13.3.0.tgz
```

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

```
./ZDT_Install_PS_V13.3.0.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 197. 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 `zpd`.

To create the user ID, use the following command:

```
useradd -d /home/ibmsys2 -g zpd -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_V13.3.0.x86_64 --install --zdtswlicense --ackVM
```

- To install the dependencies, run the following command:

```
./ZDT_Install_PS_V13.3.0.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 205. 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:

LoginTime	KeyID	FID	FeatureName	Address	User	Machine
Oct 30, 17:25:40	432975633343422885	333	CPU License	9.26.158.161	ibmsys1	zdt-dev4 Mon
	#	3				

5. 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:

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 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_XXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.
2. Obtain a license key file that invalidates the current license key from the Rational® License Key Center.
 - a. Log in to the [Rational License Key Center](#) , 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_XXXXXXXXX.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_XXXXXXXXX_update.zip` file, run the following command:

```
/opt/IBM/LDK/update_license Hostname_XXXXXXXXX_update.v2c.
```

This command installs the invalidating license key file on the server.

- b. Restart the license server daemon to make the license key file active by entering 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_XXXXXXXXX.zip`. This file verifies that the license key is invalidated and is used to complete the return of the software-based license key file.

- d. If needed, transmit the `Hostname_XXXXXXXXX.zip` file to the system you use to log in to the Rational License Key Center.
4. Use the verification file to complete the return process in the Rational License Key Center.
 - a. Log in to the [Rational License Key Center](#), 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 207.

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

1. Open the directory that contains the installer `ZDT_Install_PS_V13.3.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V13.3.0.tgz
```

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

```
./ZDT_Install_PS_V13.3.0.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_V13.3.0.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 202](#).

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 207](#).
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_V13.3.0.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_V13.3.0.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_V13.3.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V13.3.0.tgz
```

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


```
./ZDT_Install_PS_V13.3.0.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 197. 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.

```
z1091-1-10.55.12.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_V13.3.0.x86_64 --install --zdthwlicense
```

To install the dependencies silently, run the following command:

```
./ZDT_Install_PS_V13.3.0.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 216](#).

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

1. Open the directory that contains the installer `ZDT_Install_PS_V13.3.0.tgz`.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V13.3.0.tgz
```

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

```
./ZDT_Install_PS_V13.3.0.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.

```
z1091-1-10.55.12.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_V13.3.0.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 202](#).

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 216.
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_V13.3.0.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_V13.3.0.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_V13.3.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzvf ZDT_Install_PS_V13.3.0.tgz
```

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

```
./ZDT_Install_PS_V13.3.0.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 197. 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.

```
z1091-1-10.55.05.01.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 steps 1 - 4, you can install ZD&T Parallel Sysplex silently by using the following command.

```
./ZDT_Install_PS_V13.3.0.x86_64 --install --zdtps
```

To install the dependencies, run the following command:

```
./ZDT_Install_PS_V13.3.0.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 194.
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>/C4RES1.ZPD <output path of volume>/C4RES1
```

For example, `./Z1091_ADCD_install /home/ibmsys1/volumes/C4RES1.ZPD /home/ibmsys1/volumes/C4RES1`.

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 [Products contained in this release](#).

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> > 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
```

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](#).


```

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

```

- Open another terminal emulator sessions (#701), and enter Enable All in the master console of z/VM with the device address 0700.

```

z/VM ONLINE

      / VV      VVV MM      MM
     /  VV      VVV MMM     MMM
ZZZZZ /  VV      VVV MMMM    MMMM
      /  VV      VVV  MM MM  MM MM
     /  VV  VVV  MM  MMM  MM
      /  VVVVV  MM  M   MM
     /  VVV      MM      MM
ZZZZZ /  V      MM      MM

      built on IBM Virtualization Technology

Fill in your USERID and PASSWORD and press ENTER
(Your password will not appear when you type it)
USERID  ==>
PASSWORD ==>

COMMAND ==>

RUNNING  ZVM640

```

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.

```

opmsg 'VARY CN(*),ACTIVATE'
opmsg '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 ZVM710 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.Z24D.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 `oprmmsg "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.Z24D.* library. If you want to edit any parm or proc, ADCD has USER.Z24D.* 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 207](#).

- For the steps to return the hardware-based license, see [“Returning a hardware-based license key”](#) on page 216.

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.

<p>Version 9.0 or above</p>	<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 43. – 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="951 642 1471 716">cd /usr/z1090/bin ./uimserverstop</pre> 2. Enter these two commands to stop the product license server: <pre data-bbox="951 804 1471 919"># 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.
<p>Version 8.5</p>	<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 Parallel Sysplex”](#) on page 218.

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 211.

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_V13.3.0.tgz.
2. Change the authority.

```
chmod 755 ZDT_Install_PS_V13.3.0.tgz
```

3. Decompress the installer.

```
tar -xzf ZDT_Install_PS_V13.3.0.tgz
```

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

```
./ZDT_Install_PS_V13.3.0.x86_64
```

6. Select **2** for ZD&T Parallel Sysplex.
7. Select **2** to update.

- Press ENTER, and read the license agreements carefully. At the end of the license, enter yes to accept or no to decline the terms.
- 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.

```
z1091-1-10.55.12.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_V13.3.0.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:

- Stop the z/VM and Z instances. For more information, see [“Stopping Z instances and z/VM” on page 225](#)
- Run the installer from the root user ID.

```
./ZDT_Install_PS_V13.3.0.x86_64
```

- Select **2** for ZD&T Parallel Sysplex.
- Select **3** to uninstall.

Alternatively, you can run uninstall ZD&T Parallel Sysplex silently by using the following command.

```
./ZDT_Install_PS_V13.3.0.x86_64 --uninstall --zdtps
```

z/OS 2.4 ADCD reference

Learn about general release information of ADCD z/OS V2R4 May Edition of 2021, hints and tips, and the customization of new ADCD release for ZD&T.

IBM Extended z/OS ADCD for ZD&T

In addition to the general ADCD packages, IBM Extended z/OS ADCD for Z Development and Test Environment is also included in the ZD&T offering.

The following components are supported in the IBM Extended z/OS ADCD for ZD&T.

- 31-bit SDK for z/OS, Java Technology Edition V7R1, V8R0
- 64-bit SDK for z/OS, Java Technology Edition V7R1, V8R0
- Application Discovery for IBM Z V5.1.07
- Alternate Library for REXX on zSeries V1.4.0
- Automation Control for z/OS V1.1.1

- BATCHPIPES FOR OS/390® V2.1.0
- CICS Deployment Assistant for z/OS V5.3.0
- CICS Interdependency Analyzer for z/OS V5.5.0
- CICS Transaction Gateway for z/OS V9.2.0
- CICS Transaction Server Feature Pack for Dynamic Scripting V5.5.0, V5.6.0
- CICS Transaction Server Feature Pack for Modern Batch V5.5.0, V5.6.0
- CICS Transaction Server for z/OS V5.5.0, V5.6.0
- CICS VSAM Recovery V5.2.0
- CL/SuperSession V2.1
- Compiler for REXX on zSeries V1.4.0
- Connect:Direct® for z/OS V6.0.1
- Data Interfile Transfer Testing and Operations Utility for MVS VSE VM (DITTO) V1.3.0
- Data Set Commander for z/OS V9.1.0
- Data Virtualization Manager for z/OS 2.1
- Db2 Accessories Suite for z/OS V3.3, V4.1
- Db2 Administration Tools for z/OS V11.2.0, V12.1.0
- Db2 Analytics Accelerator for z/OS 7.5
- Db2 for z/OS V11.1.0, V12.1.0
- Db2 Object Comparison Tool for z/OS V11.2.0, V12.1.0
- Db2 Query Management Facility for z/OS V11.1.0, V12.2.0
- Db2 Utilities Suite for z/OS V11.1.0, V12.1.0
- Debug for z/OS 14.2.2, 15.0.3
- Dependency Based Build 1.1.2
- Developer for z/OS V14.2.2
- Encryption Facility for z/OS v2.4.0
- Engineering Workflow Management V7.0.1
- Enterprise COBOL for z/OS V6.3.0
- Enterprise PL/I for z/OS V5.3.0
- Explorer Extensions 15.0.3
- Explorer for z/OS V3.1.1.26, V3.2.0.17
- HourGlass V7.1
- Installation Manager for z/OS V1.9.1.2
- Information Management System V14.1.0, V15.2.0
- Migration Utility V4.1
- MQ V9.2
- OMEGAMON® Performance Management Suite for z/OS V5.5.1
- Open Enterprise Python for z/OS 3.9
- Operational Decision Manager V8.10.4
- Print Services Facility for z/OS V4.7.0
- Rational COBOL Runtime for zSeries V6.0.1
- RSE API 1.0.7
- SDK for Node.js - z/OS V12.0
- TDMF for z/OS V5.9.0

- Tivoli Management Services on z/OS V5.5.1
- Tools Customizer for z/OS V1.1
- UrbanCode® Deploy V7.0.5
- VS FORTRAN LIBRARY MVS V2.6.0
- WebSphere Application Server for z/OS v9.0.5.4
- Workload Scheduler distributed Agent for z/OS V9.4.0
- Workload Scheduler for z/OS V9.5.0
- XML Toolkit for z/OS V1.11.0
- Z Common Data Provider V2.1.0
- Z Monitoring Suite V1.2.0
- Z NetView® V6.3
- Z Open Automation Utilities V1.1.1
- Z System Automation V4.2
- Z Distribution for Zowe 1.0.0
- z/OS V2R4
- z/OS Connect Enterprise Edition V3.0.36

Note: There are supporting programs and more programs in the offering. Note that some bundled products have important usage restrictions. Refer to the product license for terms and conditions on the use of these programs with IBM Z Development and Test Environment.

Modified ADCD z/OS V2R4 May Edition of 2021

To support IBM Cloud Pak® for Applications, some new software packages are added or updated to ADCD z/OS V2R4 May Edition of 2021.

Software packages that are added.

- z/OS Explorer Extensions Version 15.0.3 (HLQ - FELF00)
- z/OS Explorer API Version 1.0.7 (HLQ - HUH100)
- IBM Z Open Automation Utilities Version 1.1.1 (HLQ - BGY110)
- Rocket Git 2.26.2-78 (HLQ - RL2110)
- Rocket Bash 4.3.48 (HLQ - RL2110)
- Rocket Perl 5.24 (HLQ - RL2110)
- IBM Python SDK 3.9.0 (HLQ - CYP390)

Software packages that are updated.

- IBM Dependency Based Build is updated to 1.1.1 (HLQ - BGZ110)
- IBM Debugger is updated to 15.0.3 (HLQ - EQAF00)
- z/OS Explorer is updated to 3.2.0.17 (HLQ - FEK320)

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 May Edition of 2021. 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 May Edition of 2021](#).

Note: The following topics from Application Developers Controlled Distribution (ADCD) Release Guide z/OS V2R4 May Edition of 2021 are changed.

- **ACS routines provided on ADCD**

The SMS configuration that is documented in “Customizing ADCD for ZD&T” on page 231 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 B4DBAR 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 234.

- **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 231.

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 May Edition of 2021.

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 (May 2019)
 - A4USR1 (November 2019)
 - B4USR1 (May 2020)
 - C4USR1 (December 2020)
 - D4USR1 (May 2021)
- 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)
 - B4DBB2 (ADCD May 2020 Edition)
 - B4DBC2 (ADCD May 2020 Edition)
 - C4DBB2 (ADCD December 2020 Edition)
 - C4DBC2 (ADCD December 2020 Edition)
 - D4DBB2 (ADCD May 2021 Edition)
 - D4DBC2 (ADCD May 2021 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/ZTDPY
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.

Automating NetView 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 xxxxxxx.

Valid NetView Commands are displayed as follows.

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.

Automated Task Startup at IPL time

System symbolic parameters are used by Netview to control the startup of subsystems during the IPL process. The symbolic parameters are defined in FEU . * . PARMLIB.

- For ADCD z/OS systems that are deployed to non-OpenShift targets, the symbolic parameters are defined in IEASYMAU.
- For ADCD z/OS systems that are deployed to OpenShift targets, the symbolic parameters are defined in IEASYMNZ.

The symbolic parameter name matches to the name of the z/OS subsystem, for example, SYMDEF(&CICSTS55='NOS'). This symbolic parameter name indicates that CICSTS55 will not start automatically after an IPL. Set the value of the symbolic parameter to IPL to allow the task to start automatically. You must ensure that you selected CICS 5.5 when the image was created for this deployment.

Adding a task to NetView Automation

When you customize ADCD, and want new tasks to start automatically after a system IPL, add the tasks to NetView automation. Otherwise, you need to start the new tasks manually.

To add tasks to NetView automation, complete the following steps:

1. Update NetView.CNM01.CNMCLST(GLBCNM01). Each system task is defined in this member with a block of REXX code. Take a DBB system task as an example.

```
/** DBB START **/  
X = X + 1  
VARNAME = SYSSTR||TASK.X  
INTERPRET SYSSTR||TASK.X ' = "DBB" '  
'GLOBALV PUTC 'VARNAME  
  
VARNAME = 'DBB'  
DBB = X  
'GLOBALV PUTC 'VARNAME  
  
VARNAME = SYSSTR||DESC.X  
INTERPRET SYSSTR||DESC.X ' = "Dependency Based Build" '  
'GLOBALV PUTC 'VARNAME X  
  
VARNAME = SYSSTR||STRT.X  
INTERPRET SYSSTR||STRT.X ' = "MVS S DBB" '  
'GLOBALV PUTC 'VARNAME  
  
VARNAME = SYSSTR||STOP.X  
INTERPRET SYSSTR||STOP.X ' = "MVS P DBB" '  
'GLOBALV PUTC 'VARNAME  
  
VARNAME = SYSSTR||PARN.X  
INTERPRET SYSSTR||PARN.X ' = "RESTCASH" '  
'GLOBALV PUTC 'VARNAME  
  
VARNAME = SYSSTR||STAT.X  
INTERPRET SYSSTR||STAT.X ' = '  
IF STATUS <> '' THEN  
'GLOBALV PUTC 'VARNAME  
  
VARNAME = SYSSTR||DWNL.X  
INTERPRET SYSSTR||DWNL.X ' = "3" '  
'GLOBALV PUTC 'VARNAME  
/** DBB END **/
```

X = X + 1

Specifies the increment of task counters by 1.

DBB

Sets the task variable name to system task name.

Dependency Based Build

Sets the short description of DBB.

MVS S DBB

Sets the z/OS start command for TCP/IP.

MVS P DBB

Sets the z/OS shutdown command for DBB.

RESTCASH

Sets the dependencies that must be up or active before DBB is started.

=

Sets all user tasks with this same setting

3

This value can be set to all tasks.

After you save the new task block in the member GLBCNM01, the task will be active when you start NetView next time.

2. Ensure that the related startup and shutdown messages are configured in the NetView.CNM01.DSIPARM(AZDTTABL).

- If the startup messages are configured, the following code sets the task status to UP. Take TCP/IP as an example.

```
*
IF MSGID = 'EZAIN11I' & TEXT=MESSAGE THEN
EXEC(CMD('POSTUP TCPIP') ROUTE(ONE AUTO1));
```

You must ensure that no other code blocks exist in this member, which might result in conflicting matches. If the conflicting matches occur, the first code block takes the precedence.

- If the shutdown messages are configured, the following code sets the task status to DOWN. Take TCP/IP as an example.

```
*
IF MSGID = 'IEF404I' & JOBNAME = 'TCPIP' THEN
EXEC(CMD('SETSTAT TCPIP DOWN') ROUTE(ONE AUTO1));
```

3. Set the subsystem startup requirements at the IPL time. All tasks that are defined to NetView can be controlled to start or not at IPL time, through the system symbolic parameters are defined in the FEU.Z24D.PARMLIB.

- The member IEASYMAU is used to provision ADCD to non-OpenShift environments.
- The member IEASYMNZ is used to provision ADCD to OpenShift environments.
- The system symbols that are defined need to match the z/OS task name that is defined in the member GLBCNM01. Take DBB as an example.

```
SYMDEF (&DBB= 'NOS')
```

or

```
SYMDEF (&DBB= 'IPL')
```

NOS

NOS signifies NO START.

IPL

IPL signifies to start DBB at IPL Time.

Configuring TCP/IP network

This section documents the TCP/IP configuration changes that are made to ADCD.

TCP/IP addressing was symbolized in the system parmlib configuration. A few system symbolic parameters are defined in FEU.Z24D.PARMLIB. If ADCD is used for different purpose, different members are selected. If ADCD is used for the IPL in the ZD&T, the member IEASYMAU is selected. If ADCD is used for the IPL in the Red Hat OpenShift, the member IEASYMNZ is selected. Both IEASYMAU and IEASYMNZ contain the same symbolic parameters. However, the default values are different for each environment.

The following sample is taken from the member IEASYMAU, which shows the default values.

```
SYMDEF (&DYNXCFIPADDRESS.= '10.1.1.10')
SYMDEF (&HOMEIPADDRESS1.= '10.1.1.2')
SYMDEF (&DEFAULTROUTEADDR.= '10.1.1.1')
SYMDEF (&EPHEMERALPORTS.= '10000 65534')
SYMDEF (&GBLRESL.= 'GBLRESOL')
```

DYNXCFIPADDRESS

DYNXCFIPADDRESS is the address that is used in the dynamic XCF configuration member ADCD.Z24D.TCPPARMS(ZCXDVIPA). This address needs to be in the same subnet as the TCP/IP home IP address. When an instance is provisioned from a created ADCD image, the value of

DYNXCIPADDRESS will be modified by ZD&T Enterprise Edition to match the IP address that is specified by the customer.

HOMEIPADDRESS1

HOMEIPADDRESS1 is the main TCP/IP IP address of the z/OS system, and is referenced in the ADCD.Z24D.TCPPARMS(ZPDTDEV1). When an instance is provisioned from a created ADCD image, the value of HOMEIPADDRESS1 will be modified by ZD&T Enterprise Edition to match the IP address that is specified by the customer.

DEFAULTROUTEADDR

DEFAULTROUTEADDR is the TCP/IP Gateway address, and is referenced in the ADCD.Z24D.TCPPARMS(ZPDTDEV1). DEFAULTROUTEADDR is typically the zPDT tunnel interface address. When an instance is provisioned from a created ADCD image, the value of DEFAULTROUTEADDR will be modified by ZD&T Enterprise Edition to match the IP address that is specified by the customer.

EPHEMERALPORTS

EPHEMERALPORTS defines the ephemeral ports to use for this z/OS system, and is referenced in the ADCD.Z24D.TCPPARMS(PROF3). PROF3 is an %include from the member PROF2.

GBLRESL

GBLRESL defines the Resolver configuration member to select for this IPL, and is referenced in the FEU.Z24D.PROCLIB(RESOLVER).

Improving Java application performance

This section describes some changes that can be made to improve the performance of Java applications on the systems that are based on zPDT. Some changes are implemented as follows. You can refer to the following examples for other applications.

The Java application performance can be improved by using some tuning options and a persistent Java cache that can even speed up to restart the complete system.

Improving the performance with Java cache and tuning options

Java cache can significantly speed up the startup of applications and improve the application performance for repeated workloads. The Java cache structures are populated when an application process is executed. Therefore, the benefits of the cache are not experienced until the subsequent and similar processes are executed. Some z/OS subsystems heavily rely on Java. With ZD&T-based ADCD, the following z/OS subsystems use a persistent Java cache.

- z/OS Explorer
 - RSE API
- Note:** RSE API shares the cache with z/OS Explorer.
- Dependency Based Built (Personal and Shared Daemons)
 - z/OS Connect EE
 - z/OS Management Facility
 - UrbanCode Deploy
 - Debug Profile Service

The following Java tuning options are recommended:

- Xms256m
- Xmx512m
- Xquickstart
- Xshareclasses:nonFatal
- Xshareclasses:groupAccess
- Xshareclasses:cacheDirPerm=0777

- Xscmx50m
- Xshareclasses:cacheDir=/javasc/xxx,name=yyy

Note: For the values of xxx and y, see the details of JACHER REXX that are described in the [JCACHER](#).

- Xlp:objectheap:pagesize=1m,warn,pageable
- Xlp:codecache:pagesize=1m,pageable

The following Java tuning option need to be avoided.

- Xtune:virtualized

This option was removed from the default CICS WebSphere Liberty profile because this option caused negative impact to the Liberty startup.

Persisting Java cache

The Java cache is stored in the memory. To persist the Java cache and obtain the benefits during the restart process, the cache must be saved to a file system, and then be restored back to the memory in the early process of z/OS startup. Some automation was built by using NetView, REXX, and a UNIX System Services file system.

The following NetView tasks are included:

RETCASH

Process to restore the cache in the memory when a z/OS system is started.

JCACHER

General utility to manage the Java cache. JCACHER utility is running during system shutdown to back up the cache in the memory to a UNIX System Services file system. The only prerequisite is to shut down the system in a normal way by using NetView. For more information on how to use this utility, see the details in the FEU . XXXX . PROCLIB (JCACHER).

The JACHER REXX program is stored in the FEU . XXX . SYSEXEC (JCACHER). JCACHER is a utility that can be structured. You can easily add new subsystems that need to be managed by using the required Java cache options. For ADCD, all the Java cache structures are written to subdirectories of /javasc in the UNIX System Services. To ensure that JACHER can work properly, the subdirectories and cache names need to have specific values. The subdirectory needs to be named after the started task name or PROC name. The name of the cache needs to be a concatenation of the started task name with the string cache. These parameters are defined by using the Java option Xshareclasses.

For example, z/OS Explorer is the subsystem of z/OS, and the task of z/OS subsystem on ADCD is RSED. Therefore, to define a cache for z/OS Explorer, you need to use the following Java option.

```
"-Xshareclasses:cacheDir=/javasc/rsed,name=rsedcache
```

Therefore, to add a z/OS subsystem that can be managed by JACHER, you just need to create a new subdirectory under /javasc, and set the owner of the directory and permissions to ensure that the new subsystem can write to the subdirectory. Typically, the owner of the directory is the user ID that is assigned to run started task by the RACF STARTED class. Then, set up the new subsystem to use the -Xshareclasses parameters.

Other considerations

To speed up the startup of a z/OS system, you can select not to activate or start the subsystems that consume significant resources during initial IPL process if you do not use the subsystems. Take the following z/OS subsystems as examples.

- z/OSMF for the task IZUSVR1
- z/OS Explorer for the task RSED and JMON
- UrbanCode for the task BUZAGNT
- z/OS Connect for the task ZOSCSR

- Dependency Based Build Daemons (DBB and DBBS)
- Debug for the tasks EQAPROF and EQARMTD

For more information on how to enable or disable the specific task startup during IPL process, see [“Adding a task to NetView Automation” on page 238](#).

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

<i>Table 40. Troubleshooting tips</i>	
Symptom	Problem Determination

<i>Table 40. Troubleshooting tips (continued)</i>	
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.	<p>Any message beginning with AWS is a zPDT message. Enter the following zPDT command</p> <pre>msgInfo message-number</pre> <p>Where message-number is the 10 character message. This command displays more information about the reason for the message. See 4.1.36, "The msgInfo command" in the zPDT Guide and Reference.</p>

Installing ZD&T

<i>Table 41. Troubleshooting the Installation of the ZD&T</i>	
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 47.
<p>When you are attempting to run <code>launchpad.sh</code> or <code>launchpad-console.sh</code>, 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 <code>launchpad.sh</code> or <code>launchpad-console.sh</code>, you get one of the following errors:</p> <pre>sntl-sud-xxxx is required by z1091-1-xxxx but is already installed</pre> <p>Or</p> <pre>zpdtd-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 47. If these messages are from running <code>launchpad.sh</code> or <code>launchpad-console.sh</code>, perform an uninstall of the previous version before you attempt a new installation. For the step-by-step process to perform an uninstall, see "Uninstalling the earlier versions of ZD&T" on page 47.

Table 41. 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 47.</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>zpdtsdk-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 zpdtsdk-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 42. 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 42. 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> <ul style="list-style-type: none"> • <pre>Error: Unable to locate expected license: xxxx</pre> • <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 42. 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 “reset” 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 <pre style="background-color: #f0f0f0; padding: 5px;">The remote QUERY command executed successfully</pre> – 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 <pre style="background-color: #f0f0f0; padding: 5px;">iptables -L -n</pre> <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 <pre style="background-color: #f0f0f0; padding: 5px;">/opt/safenet_sentinel/common_files/sentinel_keys_server/sntlconfigsrv.xml</pre> <p>For more information, see paragraph 8.4.2, “Security” in chapter 8 of the zPDT Guide and Reference.</p> <p>In some cases, it might be helpful to restart the license daemon where the USB key is plugged in. Restarting the license daemon is done as root with the command:</p> <pre style="background-color: #f0f0f0; padding: 5px;">/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv restart</pre>
<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> <pre style="background-color: #f0f0f0; padding: 5px;">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>

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 43. 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 244".</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 AD CD S1RES1 and SARES1 volumes.</p>

Defining the ZD&T machine characteristics (DEVMAP)

Table 44. Defining the ZD&T machine characteristics (DEVMAP). Troubleshooting defining the ZD&T machine characteristics (DEVMAP)	
Symptom	Problem Determination
Unable to run find_io command at create_devmap.pl	The path to the find_io command is not available. Ensure that you run aws_bashrc from the user ID that runs create_devmap.pl and ZD&T. This updates the path to the needed files.

Starting and stopping ZD&T

Table 45. Starting and stopping ZD&T. Troubleshooting the starting and stopping of ZD&T	
Symptom	Problem Determination
AWSSTA140E - Initialization failure after license obtained	Ensure that you are starting ZD&T from the home directory of the user who is created for running ZD&T. Ensure that you have run the aws_bashrc command while logged on as that user, and from that user's home directory.
The following messages occur during startup: AWSECH004S Unable to define RAS/FEDC memory, RC=-3. AWSECH004S Unable to define RAS/FEDC memory, RC=-3.	This message is seen with many awsckd definitions. The Linux kernel setting SHMMNI, which sets the system-wide maximum number of shared memory segments, is probably too low. The default is 4096. Try increasing this setting by adding kernel.shmni = 8192 to /etc/sysctl.conf and activate it using the command sysctl -p.
The following message occurs when iplimg z/VM 6.2 "CP requires hardware features not available on this processor"	Older z/VM V6R2 maintenance levels may not work with zPDT Version 1 Release 6. Either migrate to z/VM V6R3 or ensure PTF VM65007 is applied to your z/VM V6R2 system.

z/OS networking issues

Table 46. z/OS networking issues . Troubleshooting z/OS networking issues	
Symptom	General z/OS networking configuration errors
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.

Table 46. z/OS networking issues . Troubleshooting z/OS networking issues (continued)

<p>Problem Determination</p>	<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>
<p>Symptom</p>	<p>Cannot connect to ZD&T z/OS using a remote 3270 terminal emulator</p>
<p>Problem Determination</p>	<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.
<p>Symptom</p>	<p>Cannot ping to a z/OS defined IP address on the ZD&T instance</p>

Table 46. 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 47. 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 47. 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="412 380 1179 632"> # 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="412 1058 735 1083"> 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 41](#) 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.

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. However, the software licensing does not support the native zPDT systems.

Note: For definitions of some of the terms used in these topics, see the “Glossary” on page 271.

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 16 on page 255](#) 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.

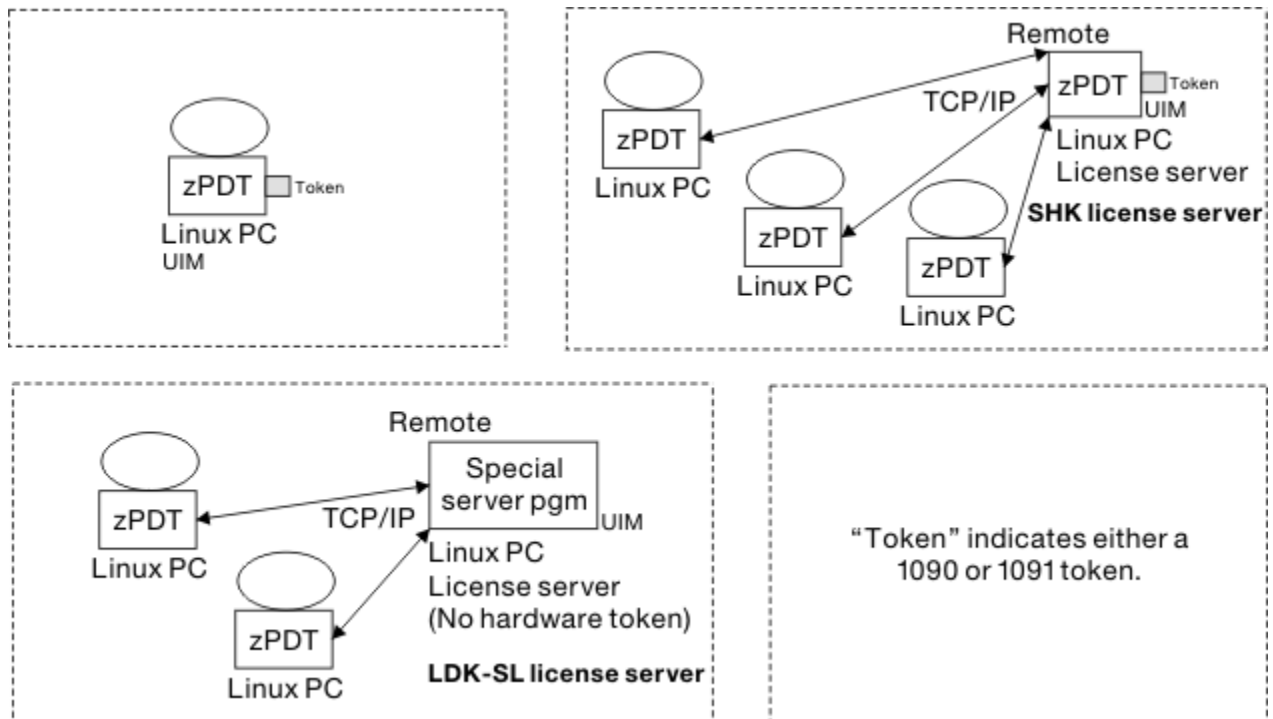


Figure 16. Options for obtaining zPDT licenses

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 16 on page 255 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 17 on page 257 illustrates the conceptual operation.

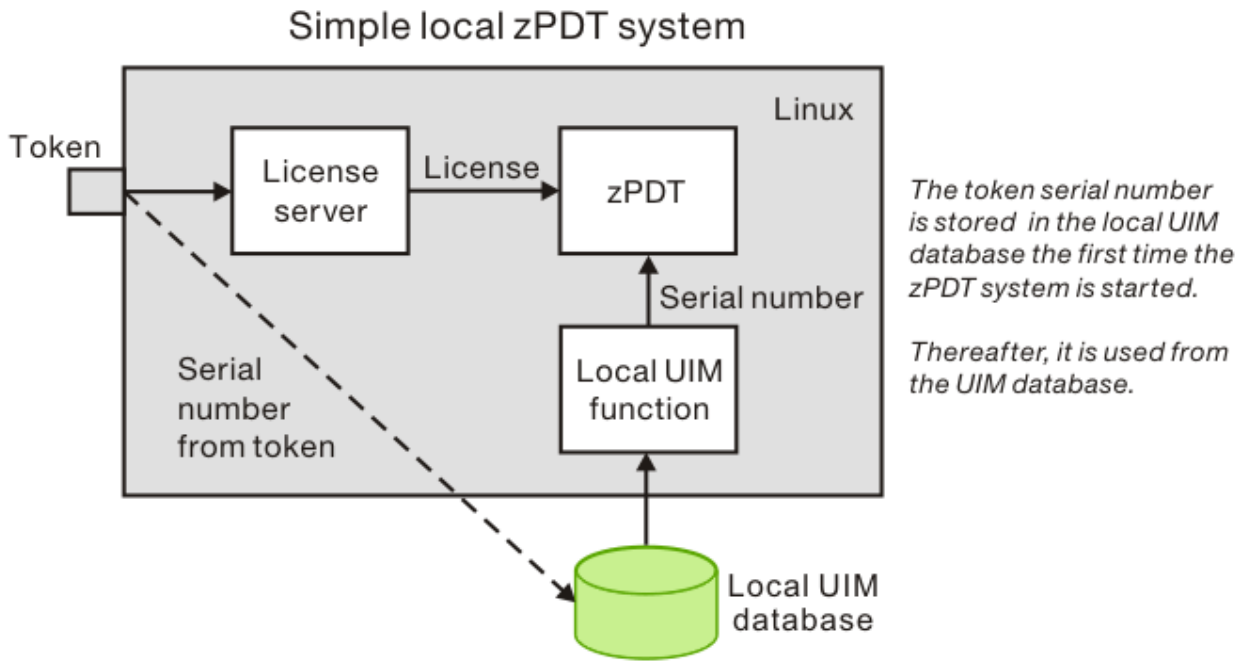


Figure 17. Simple local operation

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 18 on page 257](#) illustrates the general concept.

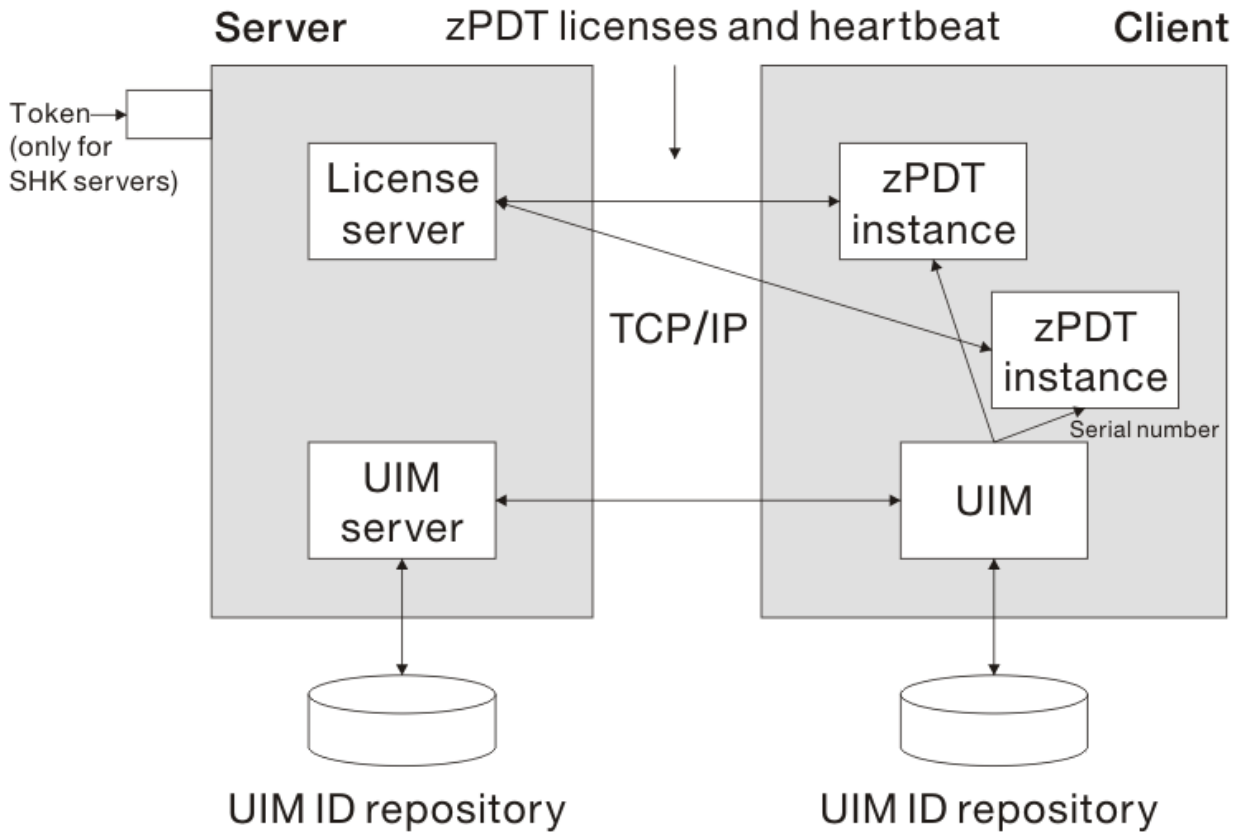


Figure 18. License and UIM servers

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 17 on page 257](#), 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 18 on page 257](#):

- 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 267](#).

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 259](#) and [“LDK client configuration” on page 260](#)

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 260. 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 259](#) or [“LDK client configuration” on page 260](#).
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 268.

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 274.](#)

ADCD

See [“Application Developers Controlled Distribution” on page 271.](#)

Application Developers Controlled Distribution

A pre-built package that contains z/OS, z/OS products, and z/OS subsystems. ZD&T users can create images from Application Developers Controlled Distribution.

Application Developers Controlled Distribution is abbreviated as ADCD.

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.

component

A collection of artifacts, such as volumes, data sets, CICS resources, or DB2 resources, from an IBM Z platform or Application Developers Controlled Distribution (ADCD). A component is reusable with different components to create more than one [image](#).

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.

image

A collection of components that are packaged together as a single deliverable. An image contains the z/OS core system and user-selected z/OS applications and subsystems. An image will be used to provision emulated IBM Z environments on target environments.

instance

A z/OS instance that is running on an emulated IBM Z hardware from a created [“image” on page 271.](#)

instance number

A number in the range of 1 and 255 assigned to each zPDT instance on a base Linux machine. Each zPDT instance must operate under a different Linux user ID and the instance number is assigned to the user ID. The instance number is used in the same manner as the LPAR number on a larger System Z[®].

license

The product entitlement that you purchased and the corresponding terms and conditions under which use is allowed.

A technical indicator on the USB hardware device that is used to activate the hardware device and allow an emulated z Systems instance to operate. The [zPDT Guide and Reference](#) and the z Systems emulator messages use the words license and licenses to refer to these indicators stored on the hardware device. You can use a remote license server with the offering that derives its name from this particular usage of the word.

In the Rational® License Key Center, a quantity of product entitlement parts purchased.

In zPDT, a logical function that enables one z Systems CP for a zPDT system. Multiple CPs require multiple licenses. The token functions provide licenses.

license entitlement

The Rational License Key Center interface and the quantity of entitlement parts purchased.

license key file

A file that represents the user's license entitlement. License key files must be obtained, installed, and available to operate the product.

license 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 272.

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

The remote ZD&T that distributes authentication by using software-based license keys.

See also [“license server”](#) on page 272.

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.

source environment

The environment where ZD&T users extract [components](#).

Several types of source environments are supported, including existing physical IBM Z platforms, Linux systems that run a [ZD&T emulator](#), and [ADCD](#).

target environment

An Intel x86-compatible environment that is capable of running one or more [ZD&T emulators](#).

Several types of target environments are supported, including Linux, Docker, and OpenStack.

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 274.

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 enable ZD&T operation.

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.

ZD&T emulator

A component of ZD&T that emulates IBM Z architecture with virtual I/O and devices on Intel x86-compatible platforms. It enables IBM z/OS operating system, middleware and software for z/OS to run on Intel x86-compatible platforms without physical IBM Z platforms. It is intended for mainframe application demonstration, development, testing and education.

The ZD&T emulator is based on the IBM System z Personal Development Tool (zPDT) program.

ZD&T emulator instance

A ZD&T emulator instance provides a working environment that contains the z/OS core system as well as user-selected applications and z/OS subsystems in the target x86-compatible environment. Having multiple ZD&T emulator instances means that multiple independent ZD&T emulator environments are running in parallel.

A ZD&T emulator instance is also known as an emulated IBM Z environment (created by ZD&T).

ZD&T Enterprise Edition web server

A server that enables [ZD&T provisioning tools](#). It is a component of ZD&T Enterprise Edition.

The ZD&T Enterprise Edition web server can be installed on the same system as the ZD&T hardware-based license server or as the ZD&T software-based license server.

ZD&T provisioning tools

Refers to a web UI with role-based access control and REST APIs. The ZD&T provisioning tools ease the process of provisioning and managing your emulated IBM Z environments. With either the web UI or REST APIs, you can perform tasks including but not limited to:

- Extract artifacts, such as volumes and data sets, from existing IBM Z platforms to create components.
- Transfer components to the storage server.
- Build images from components.
- Provision emulated IBM Z environments (also known as ZD&T emulator instances) of selected images on target x86-compatible environments.
- Manage and monitor components, images, and instances (also known as ZD&T emulator instances).

ZD&T z/OS Extraction Utilities

A component of ZD&T Enterprise Edition to be installed on one or more IBM Z platforms to allow ZD&T to use such IBM Z platforms as source environments to extract z/OS, CICS, and Db2 resources.

Installation of ZD&T z/OS Extraction Utilities is required only if you need to create components from either the physical or emulated IBM Z platforms.

zPDT

The IBM System z Personal Development Tool (zPDT) provides an environment with one or more IBM System z processors (with several emulated I/O device types), based on a personal computer Linux environment. As the name implies, it is intended for development and related purposes, such as education and demonstrations.

For more information, see [IBM ZPDT Guide and Reference](#).

Index

A

access, product license server [221](#)
activate a USB hardware device for v9.1 [17](#), [39](#), [215](#)
activation [213](#)
activation, license [213](#)
activation, Rational Token [93](#), [206](#)
activation, replacement update file [40](#), [216](#)
activation, troubleshooting USB hardware device [253](#)
activation, USB hardware device [213](#)
applying the update file [94](#), [207](#)

C

connections, troubleshooting Rational License Key Servers [252](#), [254](#)

D

device activation [17](#), [39](#), [215](#)
device activation, troubleshooting [253](#)
device map, defining [18](#), [41](#), [221](#)

E

enabling product operation [90](#), [202](#)

F

file, update [214](#)

H

hardware device activation [17](#), [39](#), [215](#)
hardware device activation, troubleshooting [253](#)

I

initial Rational Token license [214](#)
initial Rational Token license, Steps to get [214](#)
initial update file, Steps to get [214](#)
installing Software License Server [91](#)
installing Software-based License Server [204](#)

K

key [17](#), [38](#)

L

license file replacement [40](#), [216](#)
license key [17](#), [38](#)

O

obtain an update file [39](#), [92](#), [93](#), [205](#), [213](#)
Obtain an update file [213](#)
Obtaining update files for Rational Tokens [214](#)

P

pointing to a Rational License Key Server [254](#)
prerequisites [11](#), [32](#), [58](#), [197](#)
product license server access, set up [221](#)
product operation
 enabling [90](#), [202](#)

R

Rational License Key Center [39](#), [92](#), [93](#), [205](#), [213](#)
Rational License Key Server, pointing to [254](#)
Rational License Key Servers, troubleshooting connections [252](#), [254](#)
Rational Token activation [93](#), [206](#)
Rational Token license, initial [214](#)
Rational Tokens [253](#)
Rational Tokens, update file [214](#)
replacement license file [40](#), [216](#)
replacement update file [40](#), [216](#)

S

set up product license server access [221](#)
software license server [94](#)
Software License Server
 installing [91](#)
software-based license server [207](#)
Software-based License Server
 installing [204](#)
Steps to get the initial update file [214](#)
Steps to get the Rational Token license [214](#)

T

Tokens [253](#)
troubleshooting and support [242](#)
troubleshooting connections, Rational License Key Servers [252](#), [254](#)
troubleshooting USB hardware device activation [253](#)

U

update file [39](#), [92](#), [93](#), [205](#), [213](#)
update file, initial [214](#)
update file, obtain [213](#)
update file, replacement [40](#), [216](#)
update file. Rational Tokens [214](#)
USB hardware device [17](#), [38](#), [213](#)
USB hardware device activation [17](#), [39](#), [215](#)

USB hardware device activation, troubleshooting [253](#)
USB hardware device and license activation [213](#)
USB hardware device licensing [17](#), [38](#)

V

V9.1 USB hardware device activation [17](#), [39](#), [215](#)

