IBM Z Development and Test Environment 14.2.x

IBM

Tables of Contents

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
- Learners Edition
- 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

© Copyright IBM Corporation 2014, 2022

Overview

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

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

ZD&T Personal Edition
> ZD&T Personal Edition enables a single user to run an IBM® Z distribution on a personal computer. For more information about ZD&T Personal Edition, see Personal Edition.

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 three ZD&T editions.

Table 1. Differences in four ZD&T editions

Feature	Personal Edition	Enterprise Edition	Parallel Sysplex
License type	USB	- USB1 - Software-based license - Authorized User Single Session (AUSS) - Rational® Tokens	- USB1 - Software-based license - Rational Tokens

Feature	Personal Edition	Enterprise Edition	Parallel Sysplex
Single user	Yes	Yes	Yes
Multiple users	No	Yes	Yes
Scalable2	No	Yes	Yes
Cloud/VM support	No	No	No
REST API	No	Yes	No
Web UI	No	Yes	No
Native Linux interface	Yes	Yes for advanced Linux users	Yes
Dataset transfer support	No	Yes	No
Volume transfer support	No	Yes	No
Integration with IBM AD	No	Yes	No
Docker container support	No	Yes	Yes
Red Hat OpenShift support	No	Yes	No

License consumption

- Each general purpose processor consumes one license.
- z/OS Integrated Information Processor (zIIP) does not consume any license.
- Each z/OS instance running on the ZD&T emulator must have equal or more general purpose processors than zIIPs.

Notes:

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.

- **What's new in ZD&T**
  Learn the new features and enhancements in ZD&T version 14.2.0. and all subsequent 14.2.x fix packs.
- **Learning resources for ZD&T**
  Access web-based resources for learning more about ZD&T.
- **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.

What's new in ZD&T

Learn the new features and enhancements in ZD&T version 14.2.0. and all subsequent 14.2.x fix packs.

Version	New features and enhancements
V14.2.2, Enterprise Edition	- The zPDT emulator is updated to version GA-12 Fix pack 1 for all editions (personal, enterprise, and parallel sysplex). - New JIT code: JIT incorporates the open-source Eclipse OMR technology as the basis for the JIT compiler. - Capstone Disassembler is added to improve the debug capabilities. manop command - `disasm <addr>` - zPDT core Dump Enhancement: The key element is the inclusion of many (or all) of the Linux library files that are relevant to the failing program. - The timeouts associated with the awsstop command have been slightly lengthened to avoid unexpected messages while ending zPDT operation. - Crypto Express 8s level and CCA version 8.1.71i provides some minor internal fixes. - Missing script is added to check and select type of core collected. See IBM® Redbooks for more information. - DFLTCC rare issue is resolved while ipl'ing Linux on Z under 2nd level z/VM (z/VM on z/VM). - Fixed delay freeing Gen2 (LDK) licenses on zPDT termination. - Fixed core dump when awsrdr has been passed an invalid file / card deck name. - Fixed problem defining 2048 devices in the devmap. - This version is a fix pack that contains security fixes and documentation improvements. To download the fix pack, go to Fix Central. **Fixed APARs** <table><tr><th>APAR</th><th>Description</th></tr><tr><td>PH59689</td><td>ZD&T 14.x: Provision fails with error code 11102 when extracted image contains more than 256 communication devices</td></tr><tr><td>PH59976</td><td>ZD&T 14.X: LICENSE SERVER TAKES TOO LONG TO RELEASE LICENSES AFTER AN INSTANCE IS STOPPED VIA AWSSTOP</td></tr><tr><td>PH57773</td><td>ZD&T V14 LIMITED TO 2047 DEVICES</td></tr></table>
V14.2.1, Enterprise Edition	This version is a fix pack that contains security fixes and documentation improvements. To download the fix pack, go to Fix Central. **Fixed APARs** <table><tr><th>APAR</th><th>Description</th></tr><tr><td>PH58138</td><td>ZD&T 13.3+: Documentation needs update to include newly included ADCD IPL script (zdt_config_adcd_ipl.sh).</td></tr></table>

Version	New features and enhancements
V14.2.0, Enterprise Edition	- New signed installers are available for Enterprise Edition. For more information, see Enterprise Edition installation packages. - The Extended ADCD package is updated to z/OS® V2R5 May Edition of 2023. Using an earlier edition might work, although the performance is not guaranteed. For more information about the added or updated components, see Extended ADCD for ZD&T. - Installing the Sandbox Operator no longer automatically installs the IBM Cloud Pak foundational services and starts the IBM® License Service. Now you must perform these actions manually before installing the Sandbox Operator. For more information, see Install and start IBM License Service. - You can now specify what cipher suite the Instance Controller will use when communicating with the web UI. For more information, see the `spec.controller.ciphers` property in the configuration reference WaziSandboxSystem/v1alpha1. - The documentation of the Sandbox installation process is updated and enhanced for better user experience. For more information, see Installing Sandbox. **Fixed APARs**

APAR	Description
PH55992	LPARNAME (XX) filter used in IEASYMXX members will not resolve correctly to the specified value of SYSTEM_NAME.

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
- IBM® Community: Automated Testing
- zPDT® Guide and Reference
- zPDT Sysplex Extensions - 2020

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.

Trademarks

Terms and conditions for product documentation

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

Applicability

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

Personal use

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

Commercial use

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

Rights

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

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

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

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

Personal Edition

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

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

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

The following topology describes the various components of Personal Edition.

Figure 1. The topology and various components of Personal Edition

ZD&T Personal Edition provides the following features:

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

License Key (USB)

ZD&T Personal Edition comes with a USB hardware device that contains the fingerprint that is used to identify the entitlement of user. After you install Personal Edition, you need to insert the USB hardware device to the USB port of the machine to enable the license. For more information, see Enabling a license key.

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.

- **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.
- **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.
- **Installation packages**
  Learn about how to download installation packages.
- **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.
- **Installing Personal Edition**
  Learn about the steps to install ZD&T Personal Edition.
- **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.
- **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.
- **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 `./runzpdt` script that is provided in /opt/ConfigGuideSample.
- **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.
- **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.
- **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.
- **Uninstalling Personal Edition**
  Learn about the steps to uninstall Personal Edition interactively or silently.

Checklist

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

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

Table 1. New installation checklist

Installing Personal Edition		Required/Optional	Complete
1	**Prerequisites** Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	☐
2	**Obtaining installation packages** Describes the steps to download all required packages to install Personal Edition.	Required	☐
3	**Installing Personal Edition** Describes the steps to run the installation package.	Required	☐
4	**Enabling a license key** Describes the steps to obtain a license key, and apply the key to the machine.	Required	☐
5	**Configuring Z instances** Describes the steps to set up the ADCD package and create the device map to start Z instances.	Required	☐
6	**Starting Z instances** Describes the steps to start Personal Edition and Z instances.	Required	☐
7	**Stopping Z instances** Describes the steps to stop Personal Edition and z/OS® images.	Optional	☐
8	**Configuring the network** Describes various ways to configure the network.	Optional	☐

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

Table 2. Update checklist

Updating Personal Edition		Required/Optional	Complete
1	**Returning an existing license key** Describes the steps to return an existing hardware-based license key.	Required	☐
2	**Prerequisites** Describes the software, hardware requirements, and the steps to set up the basic Linux system for newer version of ZD&T.	Required	☐
3	**Obtaining installation packages** Describes the steps to download all required packages to install Personal Edition.	Required	☐
4	• If you installed ZD&T V11.0 or later versions, compete the following step: ◦ **Updating Personal Edition to the latest version** • If you installed zD&T V10.0.1 or earlier versions, complete the following steps: ◦ **Uninstalling the earlier versions of ZD&T** ◦ **Installing Personal Edition** Describes the steps to update Personal Edition.	Required	☐
5	**Enabling a license key** Describes the steps to obtain a license key, and apply the key to the machine.	Required	☐
6	**Configuring Z instances** Describes the steps to set up the ADCD package and create the device map to start Z instances.	Required	☐

To update the license, complete the following steps:

Table 3. License update checklist

Updating the license		Required/Optional	Complete
1	**Returning an existing license key** Describes the steps to return an existing hardware-based license key.	Required	☐
2	**Enabling a license key** Describes the steps to obtain a license key, and apply the key to the machine.	Required	☐

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

Table 4. Uninstallation checklist

Uninstalling Personal Edition		Required/Optional	Complete
1	**Returning an existing license key** Describes the steps to return an existing hardware-based license key.	Required	☐
2	**Uninstalling Personal Edition** Describes the steps to uninstall Personal Edition.	Required	☐

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

  - Additional `YUM` commands on RHEL 8 operating system

    ```
    yum -y install ncurses-libs
    yum -y install libnsl
    ```

  - `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 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 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 and Configuring the network.

Installation packages

Learn about how to download installation packages.

Two types of packages are provided for installation: one for Extended ADCD, which can be downloaded optionally; one for the other components of ZD&T Personal Edition. To download the installation packages, complete the following steps:

1. Log on to <u>Passport Advantage®</u>.
2. Select Software Downloads and Media Access.
3. Select Program offering and agreement number, and then click Continue.
4. Enter the part description or part number, and then 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 lists in the following tables.

After a package is downloaded, you can verify the integrity of the downloaded package by using the .md5 checksum file. To do a checksum on a downloaded package, use command `md5sum -c <checksum_file_name>`.

Table 1. ZD&T Personal Edition V14.0.0

	Name	Part No.	Image Classification	File Name
1	IBM Z Development and Test Environment Personal Edition 14.0 Installation Multilingual eAssembly	G06Z5ML	Required	N/A
2	IBM Z Development and Test Environment Personal Edition 14.0 Installation Multilingual	M0835ML	Required	ZDT_Install_PE_V14.0.tgz
3	IBM Z Development and Test Environment Personal Edition 14.0 Installer checksum Multilingual	M0836ML	Required	pe.md5

Table 2. Extended ADCD z/OS V3.1 December Edition of 2023, uploaded in March 2024

	Name	Part No.	Image Classification	File Name
1	ADCD 1Q24 z/OS V3.1 for IBM Z Development and Test 14.2.2 RSU 2309 Multilingual eAssembly	G0BGBML	Optional	N/A
2	ADCD 1Q24 z/OS 3.1 Part 1 of 20 - RES volume 1 Multilingual	M0HDDML	Optional	A3RES1.ZPD
3	ADCD 1Q24 z/OS 3.1 Part 2 of 20 - RES volume 2 Multilingual	M0HDFML	Optional	A3RES2.gz
4	ADCD 1Q24 z/OS 3.1 Part 3 of 20 - System volume Multilingual	M0HDGML	Optional	A3SYS1.gz
5	ADCD 1Q24 z/OS 3.1 Part 4 of 20 - Configuration volume Multilingual	M0HDHML	Optional	A3CFG1.gz
6	ADCD 1Q24 z/OS 3.1 Part 5 of 20 - UNIX System Services volume 1 Multilingual	M0HDJML	Optional	A3USS1.gz
7	ADCD 1Q24 z/OS 3.1 Part 6 of 20 - UNIX System Services volume 2 Multilingual	M0HDKML	Optional	A3USS2.gz
8	ADCD 1Q24 z/OS 3.1 Part 7 of 20 - UNIX System Services volume 3 Multilingual	M0HDLML	Optional	A3USS3.gz
9	ADCD 1Q24 z/OS 3.1 Part 8 of 20 - z/OS Products volume 1 Multilingual	M0HDMML	Optional	A3PRD1.gz
10	ADCD 1Q24 z/OS 3.1 Part 9 of 20 - z/OS Products volume 2 Multilingual	M0HDNML	Optional	A3PRD2.gz
11	ADCD 1Q24 z/OS 3.1 Part 10 of 20 - z/OS Products volume 3 Multilingual	M0HDPML	Optional	A3PRD3.gz
12	ADCD 1Q24 z/OS 3.1 Part 11 of 20 - z/OS Products volume 4 Multilingual	M0HDQML	Optional	A3PRD4.gz
13	ADCD 1Q24 z/OS 3.1 Part 12 of 20 - z/OS Products volume 5 Multilingual	M0HDRML	Optional	A3PRD5.gz
14	ADCD 1Q24 z/OS 3.1 Part 13 of 20 - z/OS Distribution Libraries volume 1 Multilingual	M0HDSML	Optional	A3DIS1.gz
15	ADCD 1Q24 z/OS 3.1 Part 14 of 20 - z/OS Distribution Libraries volume 2 Multilingual	M0HDTML	Optional	A3DIS2.gz
16	ADCD 1Q24 z/OS 3.1 Part 15 of 20 - z/OS Distribution Volume Multilingual	M0HDVML	Optional	A3DIS3.gz
17	ADCD 1Q24 z/OS 3.1 Part 16 of 20 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M0HDWML	Optional	A3PAGA.gz
18	ADCD 1Q24 z/OS 3.1 Part 17 of 20 - z/OS Page Datasets (Local) volume 2 Multilingual	M0HDXML	Optional	A3PAGB.gz
19	ADCD 1Q24 z/OS 3.1 Part 18 of 20 - z/OS Page Datasets (Local) volume 3 Multilingual	M0HDYML	Optional	A3PAGC.gz
20	ADCD 1Q24 z/OS 3.1 Part 19 of 20 - z/OS User volume Multilingual	M0HDZML	Optional	A3USR1.gz
21	ADCD 1Q24 z/OS 3.1 Part 20 of 20 - Stand Alone RES volume Multilingual	M0HF0ML	Optional	SARES1.ZPD
22	ADCD 1Q24 Engineering Workflow Management 7.0.2 Multilingual	M0HF1ML	Optional	A3BLZ1.gz
23	ADCD 1Q24 DB2 V12 Part 1 of 2 Multilingual	M0HF2ML	Optional	A3DBC1.gz
24	ADCD 1Q24 DB2 V12 Part 2 of 2 Multilingual	M0HF3ML	Optional	A3DBC2.gz
25	ADCD 1Q24 DB2 V13 Part 1 of 2 Multilingual	M0HF4ML	Optional	A3DBD1.gz
26	ADCD 1Q24 DB2 V13 Part 2 of 2 Multilingual	M0HF5ML	Optional	A3DBD2.gz
27	ADCD 1Q24 CICS 5.6 Multilingual	M0HF6ML	Optional	A3C560.gz
28	ADCD 1Q24 CICS 6.1 Multilingual	M0HF7ML	Optional	A3C610.gz
29	ADCD 1Q24 IMS 15.2 Multilingual	M0HF8ML	Optional	A3IMF1.gz
30	ADCD 1Q24 IBM Installation Manager 1.9 Multilingual	M0HF9ML	Optional	A3INM1.gz
31	ADCD 1Q24 DB2 Archive Logs Multilingual	M0HFBML	Optional	A3DBAR.gz
32	ADCD 1Q24 IBM Z Monitoring Suite v1.2	M0HFCML	Optional	A3KAN1.gz
33	ADCD 1Q24 WAS V9.0 Part 1 of 2 Multilingual	M0HFDML	Optional	A3W901.gz
34	ADCD 1Q24 WAS V9.0 Part 2 of 2 Multilingual	M0HFFML	Optional	A3W902.gz
35	ADCD 1Q24 Zowe Open Source Project Multilingual	M0HFGML	Optional	A3ZWE1.gz
36	ADCD Metadata for Tools Multilingual	M0HFHML	Optional	ADCDTOOLS.XML
37	ADCD 1Q24 Inventory for Sandbox Multilingual	M0HFJML	Optional	inventory.txt
38	ADCD 1Q24 Devmap for Sandbox Multilingual	M0HFKML	Optional	devmap.txt
39	ADCD 1Q24 Extended ADCD Notices Multilingual	M0HFLML	Optional	notices_ADCD_RSU 2309
40	ADCD 1Q24 Inventory with SHA256 Checksums for Sandbox Multilingual	M0HFMML	Optional	inventory_sha256.txt
41	ADCD 1Q24 IPL Configuration Script Multilingual	M0HFNML	Optional	zdt_config_adcd_ipl.sh
42	ADCD z/OS V2.5 for IBM Z Development and Test 14.2 RSU 2303 Multilingual eAssembly	G08N2ML	Optional	N/A
43	ADCD 3Q'23: z/OS 2.5 Part 1 of 20 - RES volume 1 Multilingual	M0FDQML	Optional	D5RES1.ZPD

	Name	Part No.	Image Classification	File Name
44	ADCD 3Q'23: z/OS 2.5 Part 2 of 20 - RES volume 2 Multilingual	M0FDRML	Optional	D5RES2.gz
45	ADCD 3Q'23: z/OS 2.5 Part 3 of 20 - System volume Multilingual	M0FDSML	Optional	D5SYS1.gz
46	ADCD 3Q'23: z/OS 2.5 Part 4 of 20 - Configuration volume Multilingual	M0FDTML	Optional	D5CFG1.gz
47	ADCD 3Q'23: z/OS 2.5 Part 5 of 20 - UNIX System Services volume 1 Multilingual	M0FDVML	Optional	D5USS1.gz
48	ADCD 3Q'23: z/OS 2.5 Part 6 of 20 - UNIX System Services volume 2 Multilingual	M0FDWML	Optional	D5USS2.gz
49	ADCD 3Q'23: z/OS 2.5 Part 7 of 20 - UNIX System Services volume 3 Multilingual	M0FDXML	Optional	D5USS3.gz
50	ADCD 3Q'23: z/OS 2.5 Part 8 of 20 - z/OS Products volume 1 Multilingual	M0FDYML	Optional	D5PRD1.gz
51	ADCD 3Q'23: z/OS 2.5 Part 9 of 20 - z/OS Products volume 2 Multilingual	M0FDZML	Optional	D5PRD2.gz
52	ADCD 3Q'23: z/OS 2.5 Part 10 of 20 - z/OS Products volume 3 Multilingual	M0FF0ML	Optional	D5PRD3.gz
53	ADCD 3Q'23: z/OS 2.5 Part 11 of 20 - z/OS Products volume 4 Multilingual	M0FF1ML	Optional	D5PRD4.gz
54	ADCD 3Q'23: z/OS 2.5 Part 12 of 20 - z/OS Products volume 5 Multilingual	M0FF2ML	Optional	D5PRD5.gz
55	ADCD 3Q'23: z/OS 2.5 Part 13 of 20 - z/OS Distribution Libraries volume 1 Multilingual	M0FF3ML	Optional	D5DIS1.gz
56	ADCD 3Q'23: z/OS 2.5 Part 14 of 20 - z/OS Distribution Libraries volume 2 Multilingual	M0FF4ML	Optional	D5DIS2.gz
57	ADCD 3Q'23: z/OS 2.5 Part 15 of 20 - z/OS Distribution Volume Multilingual	M0FF5ML	Optional	D5DIS3.gz
58	ADCD 3Q'23: z/OS 2.5 Part 16 of 20 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M0FF6ML	Optional	D5PAGA.gz
59	ADCD 3Q'23: z/OS 2.5 Part 17 of 20 - z/OS Page Datasets (Local) volume 2 Multilingual	M0FF7ML	Optional	D5PAGB.gz
60	ADCD 3Q'23: z/OS 2.5 Part 18 of 20 - z/OS Page Datasets (Local) volume 3 Multilingual	M0FF8ML	Optional	D5PAGC.gz
61	ADCD 3Q'23: z/OS 2.5 Part 19 of 20 - z/OS User volume Multilingual	M0FF9ML	Optional	D5USR1.gz
62	ADCD 3Q'23: z/OS 2.5 Part 20 of 20 - Stand Alone RES volume Multilingual	M0FFBML	Optional	SARES1.ZPD
63	ADCD 3Q'23: Engineering Workflow Management 7.0.2 Multilingual	M0FFCML	Optional	D5BLZ1.gz
64	ADCD 3Q'23: DB2 V12 Part 1 of 2 Multilingual	M0FFDML	Optional	D5DBC1.gz
65	ADCD 3Q'23: DB2 V12 Part 2 of 2 Multilingual	M0FFFML	Optional	D5DBC2.gz
66	ADCD 3Q'23: DB2 V13 Part 1 of 2 Multilingual	M0FFGML	Optional	D5DBD1.gz
67	ADCD 3Q'23: DB2 V13 Part 2 of 2 Multilingual	M0FFHML	Optional	D5DBD2.gz
68	ADCD 3Q'23: CICS 5.6 Multilingual	M0FFJML	Optional	D5C560.gz
69	ADCD 3Q'23: CICS 6.1 Multilingual	M0FFKML	Optional	D5C610.gz
70	ADCD 3Q'23: IMS 15.2 Multilingual	M0FFLML	Optional	D5IMF1.gz
71	ADCD 3Q'23: IBM Installation Manager 1.9 Multilingual	M0FFMML	Optional	D5INM1.gz
72	ADCD 3Q'23: DB2 Archive Logs Multilingual	M0FFNML	Optional	D5DBAR.gz
73	ADCD 3Q'23: IBM Z Monitoring Suite v1.2	M0FFPML	Optional	D5KAN1.gz
74	ADCD 3Q'23: WAS V9.0 Part 1 of 2 Multilingual	M0FFQML	Optional	D5W901.gz
75	ADCD 3Q'23: WAS V9.0 Part 2 of 2 Multilingual	M0FFRML	Optional	D5W902.gz
76	ADCD 3Q'23: Zowe Open Source Project Multilingual	M0FFSML	Optional	D5ZWE1.gz
77	ADCD Metadata for Tools Multilingual	M0FFTML	Optional	ADCDTOOLS.XML
78	ADCD 3Q'23 Inventory for Sandbox Multilingual	M0FFVML	Optional	inventory.txt
79	ADCD 3Q'23 Devmap for Sandbox 1.4 Linux Multilingual	M0G01ML	Optional	devmap.txt
80	ADCD 3Q'23 Extended ADCD Notices Multilingual	M0FFXML	Optional	notices_ADCD_RSU 2303
81	ADCD 3Q'23 Inventory with SHA256 Checksums for Sandbox Multilingual	M0G7CML	Optional	inventory_sha256.txt
82	ADCD 3Q'23 IPL Configuration Script Multilingual	M0FFZML	Optional	zdt_config_adcd_ipl.sh

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_V14.0.0.tgz that you download from IBM® is a compressed file. The installer file contains the installer ZDT_Install_PE_V14.0.0.x86_64 and the corresponding signature file ZDT_Install_PE_V14.0.0.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_V14.0.0.x86_64.sig ZDT_Install_PE_V14.0.0.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_V14.0.0.tgz.
2. Change the authority.

   `chmod 755 ZDT_Install_PE_V14.0.0.tgz`

3. Decompress the installer.

   `tar -xzvf ZDT_Install_PE_V14.0.0.tgz`

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

   `./ZDT_Install_PE_V14.0.0.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](#). 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.
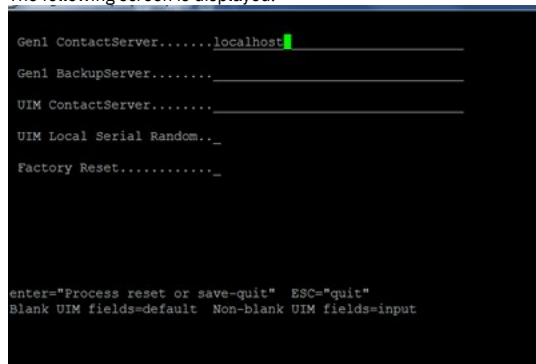
Note: The installer changes the iptables and Firewall. If you choose to install Personal Edition without network configuration, ports 1947, 3270, 9451 and 9450 are opened. You need to review the requirement with your security administration.
If you choose to install Personal Edition with network configuration, 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_V14.0.0.x86_64 --install`

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

  `./ZDT_Install_PE_V14.0.0.x86_64 --install --net`

- To install the dependencies, run the following command.

  `./ZDT_Install_PE_V14.0.0.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.
- 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 IBM 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 1. 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.
- **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.
- **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.

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 (LKC), and select your account. If you do not have access to LKC, 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 IBM 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 IBM License Key Center refers to the USB hardware device that is uniquely identified by its serial number.

1. Log in to IBM License Key Center (LKC), 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.
- **Creating the device map**
  You can create the device map from a non-root user ID.

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 <u>Checklist</u>.
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 <u>Products contained in this release</u>.

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 <pathtocommand>/create_devmap.pl pathtodisks > 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 <u>zPDT® Guide and Reference</u>.

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 `./runzpdt` script that is provided in /opt/ConfigGuideSample.

To start Z instances, copy the `./runzpdt` 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 `./runzpdt` 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
./runzpdt -d mygenerateddevmap -l cs
```

The script takes the following options:

`./runzpdt -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 ./runzpdt 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.

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

Stopping Z instances

If possible, always shut down z/OS® cleanly. Typically, shutting down cleanly begins by starting a procedure that shuts down all active subsystems.

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

You need to remember the IPL loadparm that is used to start ZD&T Personal Edition. To shut down ZD&T Personal Edition, use the respective shutdown command. All shutdown scripts are in the ADCD.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 `oprmsg "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 S0W1:   HOSTNAME   XXXX
000075 ;
Replace XXXX with Hostname used in Step2

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

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

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

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

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

11. Rename PROF2 in USER.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.

Path	Interface Name	Current State	MAC Address	IPv4 Address	IPv6 Address
F0	eth0	UP, RUNNING	fa:16:3e:2f:c9:9c	9.28.160.193	fe80::f816:3eff:fe2f:c99c%eth0
*	virbr0	UP, NOT-RUNNING	52:54:00:77:c7:21	192.168.122.1	
*	virbr0-nic	DOWN	52:54:00:77:c7:21	*	
A0	tap0	UP, RUNNING	12:3e:b7:80:ad:d1	172.26.1.1	fe80::103e:b7ff:fe80:add1%tap0
A1	tap1	DOWN	02:a1:a1:a1:a1:a1	*	
A2	tap2	DOWN	02:a2:a2:a2:a2:a2	*	
A3	tap3	DOWN	02:a3:a3:a3:a3:a3	*	
A4	tap4	DOWN	02:a4:a4:a4:a4:a4	*	
A5	tap5	DOWN	02:a5:a5:a5:a5:a5	*	
A6	tap6	DOWN	02:a6:a6:a6:a6:a6	*	
A7	tap7	DOWN	02:a7:a7:a7:a7:a7	*	

```
    End of FIND_IO
```

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

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

13. Verify whether the static IP works.
    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=osatrl2,act
    ```

Updating Personal Edition

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

For the detailed steps to update Personal Edition, see Checklist.

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

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	• Stop any currently active instance of RD&T.
	   ◦ To stop a base instance of RD&T, see Stopping Z instances.
	   ◦ If you are running a product license server:
	      1. Stop the UIM server by using the command from a non-root user ID.
	
	      `cd /usr/z1090/bin`
	      `./uimserverstop`
	
	      2. Enter these two commands to stop the product license server:
	
	      `cd /opt/safenet_sentinel/common_files/sentinel_keys_server`
	      `./loadserv stop`
	
	• 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	• 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	• 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.

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_V14.0.0.tgz.
2. Change the authority.

   chmod 755 ZDT_Install_PE_V14.0.0.tgz

3. Decompress the installer.

   tar -xzvf ZDT_Install_PE_V14.0.0.tgz

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

   ./ZDT_Install_PE_V14.0.0.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_V14.0.0.x86_64 --update

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

   ./ZDT_Install_PE_V14.0.0.x86_64 --update --net

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

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_V14.0.0.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_V14.0.0.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.

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 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).

Source environments

ZD&T Enterprise Edition can work with your source environments to extract and provision the necessary volumes or data sets that are required to target environments. The following source environment types are supported:

Genuine z/OS on IBM Z
    ZD&T z/OS Extraction Utilities must be installed and configured on such a source environment to extract z/OS artifacts.
Existing z/OS instances
    ZD&T z/OS Extraction Utilities must be installed and configured on such a source environment to extract z/OS artifacts.
Extended Application Developers Controlled Distribution (ADCD)
    ZD&T Enterprise Edition also comes with the Extended ADCD package. The Extended 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, and JES2. You can select the software packages that you need, and provision the packages out of Extended ADCD to the target environment. For more information about Extended ADCD, see Extended ADCD reference.

You can specify and configure source environments on the web server. For more information, see Adding source environments.

ZD&T z/OS Extraction Utilities

ZD&T z/OS Extraction Utilities is required if you want to create components from IBM Z platforms, either the physical or emulated IBM Z platforms. It needs 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. For more information, see Optional: Installing ZD&T z/OS Extraction Utilities.

Target environments

A target environment is an Intel x86-compatible environment that is capable of running the ZD&T emulator, where you can run z/OS software by provisioning z/OS instances with ZD&T. The following target environments are available in this tool to which you can provision instances.

Emulator

The emulator provided with ZD&T, also known as IBM System z® Personal Development Tool (zPDT®), provides the ability to run genuine z/OS software on Intel x86-compatible systems by providing emulation of the IBM Z instructions and devices. For more information about setting up the emulator manually, see Optional (Legacy): Installing the emulator.

Web server

The web server provides role-based UI and API solutions for enabling component and image creation from source environments, and provisioning z/OS instances based on the created images to target environments.

The 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.

For more information, see Setting up the web server.

Storage server

ZD&T Enterprise Edition stores extracted information on the intermediary storage machine, for example, 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 prepare such a storage server and ensure enough storage on the storage server to hold the required artifacts.

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.

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 1. The AUSS and various components of Enterprise Edition
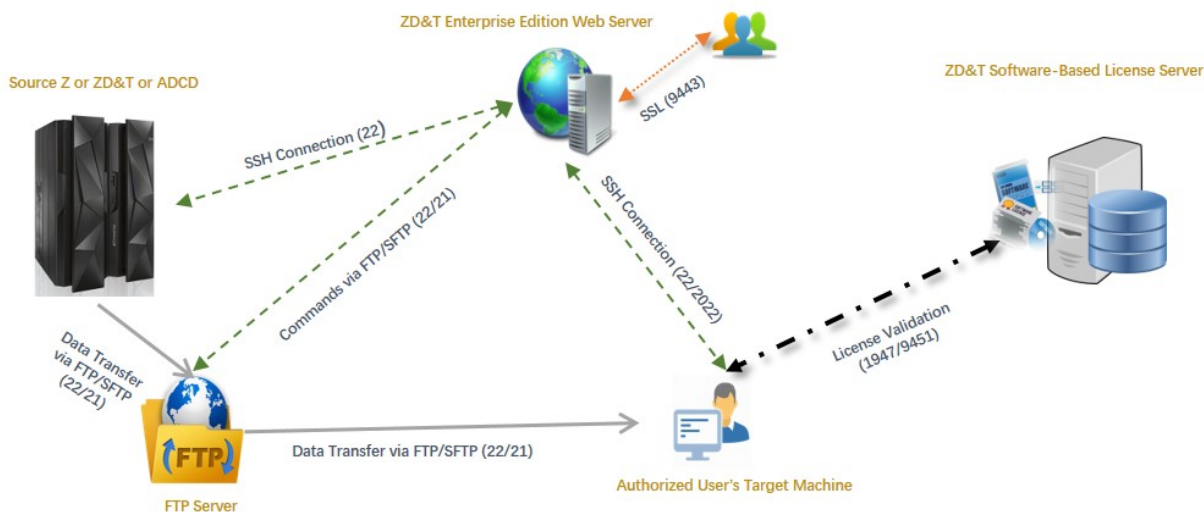
Figure 2. The RVU and various components of Enterprise Edition
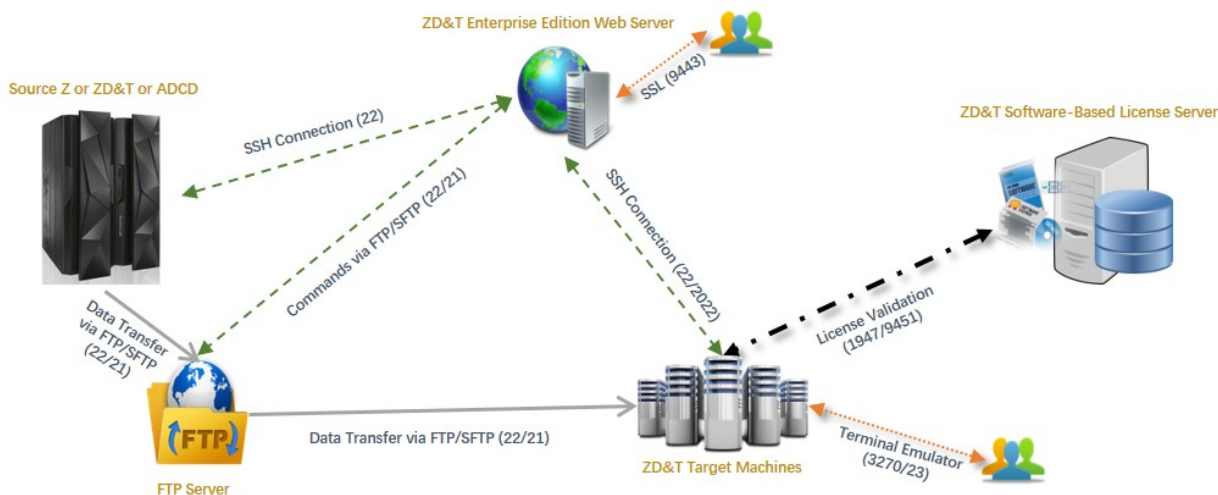
For more information, see Setting up the software-based license server.

Note: The web server, storage server, and 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.

- **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.
- **Prerequisites**
  Learn about the ZD&T prerequisites.
- **Downloading installation packages**
  Learn about how to download installation packages.
- **Installing and configuring**
  Learn how to install and configure ZD&T components.
- **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.
- **Scenarios**
  Learn about the scenarios of ZD&T Enterprise Edition for development and test activities.
- **Troubleshooting and workarounds**
  Learn how to troubleshoot ZD&T Enterprise Edition for development and test activities.
- **Sharing feedback**
  You can share anonymous data on the web server to help shape the product.
- **Uninstalling**
  Learn how to uninstall ZD&T.

Checklist

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

Table 1. New installation checklist for Enterprise Edition

Installing Enterprise Edition	Required/Optional	Complete
1 Hardware and software requirements Describes the software and hardware requirements, and the steps to set up the basic Linux® system.	Required	☐
2 Obtaining installation packages Describes the steps to download all required packages to install Enterprise Edition.	Required	☐
3 Setting up the software-based license server Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐
4 Installing the web server Describes the steps to install the Enterprise Edition.	Required	☐
5 Optional: Installing ZD&T z/OS Extraction Utilities If you want to create components from IBM Z platforms, install the ZD&T z/OS Extraction Utilities.	Conditional [1]	☐
6 Configuring the web server Describes the steps to configure the Enterprise Edition server environment before you start ZD&T Enterprise Edition.	Required	☐
7 Starting and stopping the web server Describes the steps to start the Enterprise Edition web server.	Required	☐
8 User's Guide • Enabling z/OS software to run on target environments • Using REST APIs. Describes the steps to create images, and provision instances from created images by using the Enterprise Edition web user interface or REST APIs.	Required	☐
Note: 1. Required only if you want to create components from IBM Z platforms.		

Table 2. Update checklist for Enterprise Edition

Updating Enterprise Edition	Required/Optional	Complete
1 Hardware and software requirements Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	☐
2 Obtaining installation packages Describes the steps to download all required packages to install Enterprise Edition.	Required	☐
3 Upgrading the web server Describes the steps to update Enterprise Edition.	Required	☐
4 Configuring the web server Describes the steps to configure the Enterprise Edition server environment before you start ZD&T Enterprise Edition.	Required	☐
5 Starting and stopping the web server Describes the steps to start the Enterprise Edition web server.	Required	☐

Updating Enterprise Edition	Required/Optional	Complete

6	User's Guide
	
	    • Enabling z/OS software to run on target environments
	    • Using REST APIs.
	
	Describes the steps to create images, and provision instances from created images by using the Enterprise Edition web user interface or REST APIs.
		Required	☐

Table 3. Update checklist for license server

Updating license server	Required/Optional	Complete
1	Returning a software-based license key Describes the steps to return a software-based or hardware-based license or Rational® Tokens.	Required	☐
2	Upgrading the software-based license server Describes the steps to update the software-based license server or hardware-based license server.	Required	☐
3	Setting up the software-based license server Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐

Table 4. Update checklist for license

Updating license	Required/Optional	Complete
1	Returning a software-based license key Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	☐
2	Setting up the software-based license server Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐

Table 5. Uninstallation checklist for Enterprise Edition

Uninstalling Enterprise Edition	Required/Optional	Complete
1	Returning a software-based license key Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	☐
2	Uninstalling the software-based license server Describes the steps to update the software-based license server or hardware-based license server	Required	☐
3	Uninstalling the web server Describes the steps to uninstall Enterprise Edition.	Required	☐

Prerequisites

Learn about the ZD&T prerequisites.

- **Hardware and software requirements**
  Learn about hardware and software requirements for ZD&T Enterprise Edition.
- **Required firewall ports for network flow**
  Before you install and use ZD&T Enterprise Edition, make sure that you are aware of the required firewall ports.
- **Limitations**
  Before you use ZD&T Enterprise Edition, make sure that you are aware of the following known limitations.

Hardware and software requirements

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

For a complete list of 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
- Source environments
  - z/OS system requirements
  - Db2 extraction requirements
- Target environments:
  - Linux target environment requirements
  - Docker target environment requirements
  - Red Hat OpenShift requirements

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, and Enterprise Edition metadata, must be set up. To transfer volumes images files from the storage server or to the storage server, you must choose SFTP as the transferring method.

- **Disk space**
  - Sufficient space is needed to hold numerous and potentially large files for extracted IBM® Z volumes.

- Sufficient disk space to potentially hold multiple Extended ADCD z/OS® distributions.
- **Software requirements**
  - A running SFTP server
- **SFTP server**
  - Open the firewall port for SFTP command.

z/OS system requirements

If you want to set up source environments from either genuine z/OS on IBM Z® or existing z/OS instances to extract and provision the necessary volumes or data sets, the following z/OS system prerequisites must be met. Note that extracting from a z/OS instance initially provisioned from Extended ADCD is also considered as extracting from an existing z/OS instance; therefore, the following prerequisites apply.

- Supported z/OS versions: V2.4 and V2.5.
- You must install all the PTFs that are identified with the SMP/E FIXCAT of `IBM.TargetSystem-RequiredService.AlternateHypervisors`. Installation of all these PTFs must be completed on the z/OS system before extraction. For more information, see [IBM Fix Category Values and Descriptions](#).
- You need to install ZD&T z/OS Extraction Utilities to create components from IBM Z platforms. For more information, see [Installing ZD&T z/OS Extraction Utilities](#).
- To run the extraction code, the following minimum requirements for the z/OS user ID must be met:
  - For 32-bit Java™, the user ID needs a minimum region of approximately 150 MB.
  - For 64-bit Java, the user ID needs a minimum region of approximately 250 MB.

  The region size can be specified in the OMVS segment (**ASSIZEMAX**) for remote login, for example, SSH, and for TSO login through the TSO segment (**SIZE** or **MAXSIZE**). If OMVS or TSO segments do not limit sizes, global settings might need to be adjusted in the BPXPRM*xx* member of PARMLIB. These values are only estimates, and your environment might need more, or you might be able to specify less.
  Example

  ```
  alu ibmuser OMVS(ASSIZEMAX(262144000))
  ```

- The TCP/IP profile must contain the following values:

  z/OS IP address
  : This value must be defined in either the HOME or INTERFACE statement. Using system symbolic parameters for the IP address is not supported.
  z/OS default route address
  : This value must be defined using the BEGINRoutes statement. Using system symbolic parameters for the default route is not supported. Routing configurations via OMPROUTE are also not supported.

- 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 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 a volume component](#) and [Creating a data set component](#).
  - Make sure to grant READ access to DFDSS program ADRDSSU.
  - At least one offline DASD device must be available on the source z/OS system before the system is extracted.
  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.
  - To extract on-premise z/OS volumes, the user ID that runs the extraction must have the capability to use substantial CPU time, as this operation might take a considerable amount of time. To prevent potential errors, it is recommended to grant the user ID unlimited CPU time, at least during the extraction process. You can use the TSO OMVS segment to modify a user's maximum CPU time.
    Example

    ```
    ALTUSER userid OMVS(CPUTIMEMAX(86400))
    ```

- If you want to extract volumes from an existing z/OS instance, Java must be installed, and the user who creates volume components must have access to Java after login.

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 DSNTEP2.

Restriction:
The latest edition of Extended ADCD is distributed with Db2 versions exceeding V12 Function Level 504, and therefore segmented (non-UTS) and partitioned (non-UTS) tablespaces are not supported. 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 provisioning of the Db2 component might fail when Db2 objects are created.

Linux target environment 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.
- 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.
- Starting with V14.0.1, sha256sum must be installed. It can be installed with package coreutils.
- The emulator must be installed on a different machine than the web server, storage server, and license server.

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 Required firewall ports for network flow.

- 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 libstdc++.i686
    yum -y install perl
    yum -y install zip
    yum -y install unzip
    yum -y install gzip
    yum -y install bc
    yum -y install ncurses-libs # Required for RHEL 8 only
    yum -y install libnsl # Required for RHEL 8 only
    ```

  - `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 perl
    apt-get -y install zip
    apt-get -y install unzip
    apt-get -y install gzip
    apt-get -y install bc
    apt-get -y install libasound2
    apt-get -f install
    ```

  Note: The 'nc' command is not available by default on the RHEL 7.4 and 7.5. As the 'nc' command is required to pass the validation before you start a 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 'zpdt' 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 'zpdt'.
  - If you use a non-root user ID to provision instances, assign the user ID to the group 'zpdt'.
- **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 Linux port 2022 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.

     ```
     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 172.26.1.2:22
     iptables -A FORWARD -p tcp -d 172.26.1.2 --dport 2022 -j ACCEPT
     ```

  3. Run 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 commands `ifconfig`, `ip -o address show`, and so on.
  - 2022 is the port number that will be routed to port 22.
  - 172.26.1.2 is the IP address of network interface tap0, which can be found by running command `find_io`.

Docker target environment 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 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
  - 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 requirements

Red Hat OpenShift®
Sandbox is supported in customer-managed clusters on x86_64 architecture for OpenShift EUS releases 4.10 and later. For more information, see Install Red Hat OpenShift Container Platform.

On IBM Cloud® managed Classic or VPC OpenShift, Sandbox is supported only for x86_64 architecture clusters running OpenShift 4.11 or later on RHEL8 nodes. For OpenShift 4.8 to 4.10, use Sandbox 2.2.

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

Figure 1. Custom SecurityContextConstraints definition

```
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:
- '*'
```

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

Figure 2. Custom SecurityContextConstraints 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:
- '*'
```

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

Note: When you plan for persistent volume storage, if you intend to expand z/OS storage later, for example by adding extra volumes, you should account for that as well.

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 1. Cluster base resources

	Count	Memory (GiB)
Control nodes	3	16/node

Table 2. Scheduling capacity for worker nodes

Software	Memory (GiB)	CPU (cores)	Ephemeral storage (GiB)	Persistent storage (GiB)
1 single sandbox instance	$M + 2$	$P + 1$	2	$V * 1.125$

A minimal `Starter` profile with a single instance that uses the included Extended ADCD image needs M = 8 GiB, P = 3, and V = 270 GiB. So the cluster requires at least one worker node with a capacity of at least 10 GiB of memory, 4 cores, 2 GiB of ephemeral storage and approximately 304 GiB of persistent storage.

To scale up from the `Starter` profile, scale up worker nodes accordingly.

For example, to run 5 instances with Extended ADCD, 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.

Note: Due to the requirements of setting up and running the zPDT emulator, Sandbox 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 Sandbox service account and Sandbox system containers that are executed by the access. For more information about securing your cluster, see Kubernetes documentation.

Sandbox requires the IBM License Service. To enable this, Sandbox installs the IBM Cloud Pak® foundational services when it is installed, and creates the IBM License Service automatically if it is not already in the cluster.

- **Storage server requirements**
- **z/OS system requirements**
- **Db2 extraction requirements**
- **Linux target environments requirements**
- **Docker target environments requirements**
- **Red Hat OpenShift target environments requirements**

Installing Sandbox

Before you install Sandbox, there might be additional preparation required.

Make sure that you understand all of the prerequisites and requirements before you start your installation to ensure that your installation goes as smoothly as possible. For more information about setting up Red Hat® OpenShift® Container Platform, see Setting up your Red Hat OpenShift Container Platform cluster. For more information about planning your deployment and implementation, see Planning.

The following tasks represent the typical task flow for installing your Sandbox:

1. Install Red Hat OpenShift Container Platform
2. Configure storage
3. Add the IBM® operator catalog and prepare your cluster
4. Install and start IBM License Service
5. Install IBM Wazi Sandbox operator
6. Create the entitlement key secret
7. Enable the Sandbox License Server
8. Install the Sandbox provisioning tools
9. Create your Sandbox instance

1. Install Red Hat OpenShift Container Platform

Review the Planning section.

Sandbox requires OpenShift to be installed and running. For detailed steps about how to install OpenShift, see Installing Red Hat OpenShift Container Platform.

Note: Ensure that you have a supported version of OpenShift installed. For more information about OpenShift compatibility, see Supported versions of Red Hat OpenShift Container Platform.
For any steps that use the Red Hat OpenShift Container Platform command line interface (CLI), you must be logged in to your OpenShift cluster with `oc login`.

After you install OpenShift, you can verify and gain access to your container software by using your IBM entitlement key that you create in Create the entitlement keys.

2. Configure storage

Configure cluster persistent storage

Sandbox supports OpenShift Container Storage and was tested with the Rook-Ceph® operator, NFS, and additional storage providers and storage classes that are listed below.

- **Requirements:**
  - `volumeBindingMode: Immediate` mode
    Sandbox does not support storage classes that have the volume binding mode `WaitForFirstConsumer`. If you want to use a storage class that has this mode, create a new storage class with a different name, change the mode to `volumeBindingMode: Immediate`, and use the newly created storage class. For more information, see Volume Binding Mode.

  - `ReadWriteOnce` access mode
    Volume files must not be used by more than one instance at a time or the z/OS® volumes might corrupt.

  Sandbox stores z/OS volumes for each instance on a `PersistentVolumeClaim`. When you plan for persistent volume storage, account for the situation where you will expand z/OS storage such as by adding extra volumes, and make the `PersistentVolumeClaim` large enough to accommodate this growth.

  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. Consult your z/OS system administrator.
  - 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.
- **Recommended storage providers**
  For Linux®® on x86 hardware, the following storage providers are supported for Sandbox and recommended:

  - OpenShift Container Storage 4.2 or later versions
  - IBM Cloud® Block storage and IBM Cloud File storage
  - IBM Storage Suite for IBM Cloud Paks. This suite of offerings includes:
    - File storage from IBM Spectrum® Scale
    - Block storage from IBM Spectrum Virtualize, FlashSystem or DS8K
- **Supported storage classes in IBM Cloud**
  The following storage classes in IBM Cloud have been tested with Sandbox. High performance classes are recommended, because disk performance directly affects the emulator.

  - **OpenShift (VPC)**
    - `ibmc-vpc-block-10iops-tier` (Recommended)
    - `ibmc-vpc-block-5iops-tier`
    - `ibmc-vpc-block-custom`
    - `ibmc-vpc-block-general-purpose`
  - **OpenShift (Classic)**
    - `ibmc-block-gold` (Recommended)
    - `ibmc-block-silver`
    - `ibmc-block-custom`
    - `ibmc-file-gold-gid`
  Note:
  - Storage classes with the reclaim policy `Retain` can work, but are unsupported because they can lead to data security issues.
  - For Classic OpenShift, the file-based storage classes must be the `-gid` versions. And in the custom resource (CR) for the sandbox instance, you must specify the corresponding `fsGroup` (`spec.fsGroup`). For example, by default, this is 65531 for the `ibmc-file-gold-gid`.
  - Slower storage classes can be deployed and work, but are unreliable due to the dependency of the sandbox instance to have high I/O to their disk volumes.
- **Using NFS storage**

NFS is not recommended due to potential performance issues and possible accidental corruption of z/OS volume files on shared storage. NFS storage is not supported in the Sandbox web server UI.

If NFS storage is used, follow these steps strictly:

1. Enable the group write mode for the exported share on the NFS server, for example, by using the command: `chmod g+rwx /path/to/share`.
2. On the NFS server, note the numeric group ID of the share by using `ls -lnd` on the directory. In the following example, the numeric group ID is 4567.

```
$ ls -lnd /path/to/share
drwxrwxr-x 16 1234 4567 4096 Nov 10 20:04 /path/to/share
```

3. When using the exported share, manually create **PersistentVolume** and **PersistentVolumeClaim**.
4. For the **WaziSandboxVolumeCopy** and **WaziSandboxSystem** custom resources, set the **spec.fsGroup** parameter to the numeric group ID used by the NFS server for the exported share and discovered in step 2.

- **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 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.
  - To simplify copying of a **PersistentVolumeClaim** that has cloud-ready z/OS volume images, consider installing OpenShift Virtualization. The included Containerized Data Importer can be used to create an efficient clone of a **PersistentVolumeClaim** or copy it based on the capabilities of the storage class.

  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.

Configure image storage

In addition to cluster storage, to use Sandbox instances, an SFTP server is required. The SFTP server is used to transfer z/OS image volumes to the cluster storage for sandbox instances.

You can install an SFTP server on a Linux server or OpenShift.

Note:

- No SFTP server is provided with or managed by Sandbox; however, you can find images for SSH/SFTP in Docker Hub.
- To transfer faster, set up the server inside the cluster with fast local storage, but this is not required.
- Sandbox must have a user credential that has read access to the SFTP server. SSH certificates are not supported currently.

Option 1: Installing an SSH or SFTP server on a Red Hat Linux server

To install an SFTP server on a Red Hat Linux 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 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
```

Option 2: Installing an SSH or SFTP server on OpenShift

As you might follow different guidelines for the security of your SFTP servers and container with privilege, you can merge the following instructions with the guidelines you followed for your business process.

To deploy an SFTP server within your cluster, using an SFTP container image that is approved with your organization might be the easiest way. Also, you can build a custom container to start an SFTP server. The following example shows a Docker file that can be used to build a functional SFTP container.

```
FROM registry.access.redhat.com/ubi8/ubi:latest
ARG home=/home/sshuser
RUN yum -y update && \
    yum -y install openssh-server \
    openssh-clients && \
    /usr/bin/ssh-keygen -A && \
    groupadd sshgroup && \
    useradd -ms /bin/bash -g sshgroup sshuser && \
    echo 'sshuser:PASSWORD' | chpasswd && \
    mkdir $home/.ssh && \
    touch $home/.ssh/authorized_keys && \
    chown sshuser:sshgroup $home/.ssh/authorized_keys && \
    chmod 600 $home/.ssh/authorized_keys
EXPOSE 22
CMD /usr/sbin/sshd && sleep infinity
```

In this example, the installation replaces `PASSWORD` with the password you specified. Also, to improve the security of the access, you need to change the `sshd` configuration. For example, you can change the configuration to prevent login sessions to allow only the access as an SFTP server.

To deploy the above container into your OpenShift or Kubernetes environment, you can use the following deployment YAML file.
Note: Due to the requirements of `sshd`, the container needs to be started with privileged access, which might have specific security concerns for your organization. Therefore, you can follow all the guidelines of your organizations, and also follow the best practices where they are applicable.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: <instance name>
  namespace: <namespace>
spec:
  selector:
    matchLabels:
      app: <instance name>
  template:
    metadata:
      labels:
        app: <instance name>
    spec:
      containers:
        - name: <instance name>
          image: <image>
          ports:
            - containerPort: 22
          securityContext:
            privileged: true
            allowPrivilegeEscalation: true
          volumeMounts:
            - name: <volume name>
              mountPath: /data-stor
      volumes:
        - name: <volume name>
          persistentVolumeClaim:
            claimName: <PVC name>
```

To use the YAML file, you need to change any values as needed. Also, you need to manually create the PVC and PV that has the required space to hold the z/OS volumes that are needed before you change the values. To allow access your user ID and other user IDs that are dependent to your environment, you might need to change the permissions to your storage.

3. Add the IBM operator catalog and prepare your cluster

Add the IBM operator catalog

The IBM operator catalog is an index of operators available to automate deployment and maintenance of IBM Software products into Red Hat OpenShift Container Platform clusters. Operators within this catalog are built following Kubernetes best practices and IBM standards to provide a consistent, integrated set of capabilities. The catalog can be added to any Red Hat OpenShift Container Platform 4.10 and newer cluster by application of the *CatalogSource* resource.

On Red Hat OpenShift Container Platform clusters that run Red Hat OpenShift Container Platform 4.10 or later, the catalog tag in the CatalogSource is set to latest to obtain the most recent version of the catalog. Another Option on the Red Hat OpenShift Container Platform clusters that run version 4.10 or later, the *olm.catalogImageTemplate* annotation can be used to switch the catalog tag from latest to a different tag based on the Kubernetes version used by the cluster. Setting the annotation causes the catalog to automatically switch to a catalog that is compatible with the cluster whenever the cluster is updated to a new Kubernetes version.

Creating CatalogSource with OpenShift

1. Log in to your Red Hat OpenShift Container Platform cluster.
2. Click Home > Search > Resources, then select CatalogSource.
3. Click Create CatalogSource.
4. In the CatalogSource name field, enter `ibm-operator-catalog`.
5. In the Publisher name field, enter `IBM`.
6. In the Image (URL of container image) field, enter `icr.io/cpopen/ibm-operator-catalog:latest`.
7. Under Availability, select the Cluster-wide CatalogSource option.

You can verify successful CatalogSource creation by going into the OperatorHub where the IBM Operator Catalog now exists as a source value. You are also provided with a confirmation after the CatalogSource value is completed in Red Hat OpenShift Container Platform. It can sometimes take approximately 5 minutes for the CatalogSource to appear in OperatorHub after successful completion.

CLI enablement

The catalog can be added by applying the following YAML file to the Red Hat OpenShift Container Platform cluster. Create this file and name it *catalog_source.yaml*.

```
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
```

Apply the YAML file by using the command:

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

Verify the CLI installation:

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

You receive this output on error:

```
Error from server (NotFound): catalogsources.operators.coreos.com "ibm-operator-catalog" not found
```

You receive this output on success:

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

4. Install and start IBM License Service

Sandbox requires the IBM License Service, a part of the IBM Cloud Pak® foundational services, version 3.23 or a later 3.x version, to track Virtual Processor Core (VPC) usage.

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.

After you set up your cluster and installed the IBM operator catalog, follow the instructions at IBM Cloud Pak foundational services version 3.23, and then create an OperandRequest to start the License Service by using the provided YAML.

1. Create an OperandRequest in OpenShift:
   a. Log in to your Red Hat OpenShift Container Platform cluster.
   b. Click Home > Search > Resources, and then select OperandRequest.
   c. Click Create OperandRequest.
   d. Paste the following YAML into the editor:

```
apiVersion: operator.ibm.com/v1alpha1
kind: OperandRequest
metadata:
  name: create-ibm-license-service
spec:
  requests:
    - operands:
        - name: ibm-licensing-operator
      registry: common-service
      registryNamespace: ibm-common-services
```

e. Click Create.
2. Start the License Service by using CLI:
    a. Add the IBM operator catalog by saving the preceding YAML file to the Red Hat OpenShift Container Platform cluster.
    b. Create a file with this YAML and name it license_service.yaml.
    c. Apply the YAML file by using the following command:

```
oc apply -f license_service.yaml
```

For more information about the License Service, see IBM Cloud Pak foundational services License Service.

5. Install IBM Wazi Sandbox operator

After Sandbox was installed in the cluster with the IBM Operator Catalog or CASE bundle, you will be able to see IBM Wazi Sandbox Operator inside OperatorHub. Install the Sandbox Operator and it will be managing the lifecycle of sandbox instances.

A project is a Kubernetes namespace. 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.

You can install your operator by using the OpenShift console or the OpenShift CLI.

Option 1: Install the operator with the OpenShift console

1. Select the Administrator perspective.

2. Choose Operators > OperatorHub.

3. Filter by keyword for IBM Wazi.

4. Click the IBM Wazi 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 single specific namespace is recommended because Sandbox is privileged and can consume significant cluster resources. Installing only to one namespace for authorized users gives the cluster administrator better control over who can use those privileges and resources.
        Important: Although the Sandbox Operator can be installed in multiple namespaces on a cluster, this is highly discouraged. If you choose to install to multiple namespaces, do not install different versions because this can cause compatibility problems with cluster-wide resources such as the `CustomResourceDefinition`s used by Sandbox.

7. Click Install.

OpenShift informs you that the installation is complete. You can verify that installation by navigating to Operators > Installed Operators, and selecting your project from the Projects drop-down list. The operator that you have just installed and all of its dependent operators in the project are listed with a status of Succeeded.

Option 2: Install the operator with the OpenShift CLI

1. Determine whether you want to create a new, or reuse an existing namespace to install your operator into. To create a new namespace in the CLI, run the following command:

```
oc create namespace <namespace>
```

Where `<namespace>` is the name of the namespace that you want to create.

2. Create Operator Group.
    You must create an operator group in your custom project (namespace), or your operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an Operator group for the namespace, do not create a second one.

Create the Operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: <your-operator-group>
  namespace: <namespace>
spec:
  targetNamespaces:
    - <namespace>
EOF
```

3. Install your operator with the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: <your-operator-group>
  namespace: <namespace>
spec:
  channel: <latest-version-of-your-cloud-pak>
  installPlanApproval: Automatic
  name: <your-operator-group>
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

4. After a few minutes, the operator is installed. Run the following command to verify that all of the components are in the `Succeeded` state. 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
```

6. Create the entitlement key secret

Complete the following steps to create a docker-registry secret to enable your deployment to pull operand images from the IBM Entitled Registry.

Option 1: Create the entitlement key secret with the OpenShift console

1. Obtain the entitlement key that is assigned to your IBMid. Log in to with the IBMid and password details that are associated with the entitled software.
2. In the OpenShift console, ensure that you are in the Project (for example, `paks`) that you installed the operator into.
3. Click Workloads > Secrets > Create, then select Image pull secret.
4. In the Secret name field, enter `ibm-entitlement-key`.
5. In the Registry server address field, enter `cp.icr.io`.
6. In the Username field, enter `cp`.
7. In the Password field, enter the credential that you received from entitlement.
8. (Optional) In the Email field, enter a contact email address.

Option 2: Create the entitlement key secret with the CLI

1. Obtain an entitlement key for the image pull secret from IBM Container Library. Click Get an entitlement key if you are not on the entitlement page.
2. Create the secret on the namespace that you installed the **wazi-sandbox-operator** with the following command:

```
oc create secret -n <NAMESPACE> docker-registry ibm-entitlement-key --docker-server=cp.icr.io --docker-username=cp --docker-password=<your IBM Entitled registry key> --dry-run -o yaml | oc apply -f -
```

7. Enable the Sandbox License Server

The Sandbox License Server provides a server for centralized management of license keys for one or more instances of Sandbox. To enable the Sandbox License Server, you need to install the License Server, obtain an update file, and apply the update file to the Sandbox License Server machine. Each license key file is uniquely identified based on a hardware fingerprint that is determined at the time of generating the request file.

Warning:

- The Sandbox 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 Sandbox 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 Sandbox License Server again after the movement.
- The license for Sandbox is based on a hardware signature. If the underlying hardware changes, the licenses on the Sandbox License Server will become invalid, and prevent any Sandbox instance from working. This scenario might occur easily if the Sandbox License Server is installed on a virtual machine, and it will require the assistance of IBM to help resolve the problem. This time-consuming effort might need a week to resolve the problem. Therefore, it is strongly recommended that you install the Sandbox License Server only on a physical hardware.

Step 1: Install the Sandbox License Server

To install the Sandbox License Server, run the installer from root user ID.

1. Open the directory that contains the installer wazi-sandbox-install.tgz.
2. Change the authority.

```
chmod 755 wazi-sandbox-install.tgz
```

3. Decompress the installer.

```
tar -xvf wazi-sandbox-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 then 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 then enter `y` to accept or `n` to decline the terms.
9. Enter `y` to install the Sandbox License Server on the virtual hardware, or enter `n` to decline the installation.
10. After the installation completes, run the following commands to verify whether the installation is successful.
    - RHEL operating system

    ```
    rpm -qa | grep zpdt
    ```

    - Ubuntu operating system

    ```
    dpkg -l | grep zpdt
    ```

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

    ```
    zpdtldk
    zpdtuim
    ```

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

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

    - To delete the user ID, use the following command.

    ```
    userdel -r ibmsys2
    ```

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

- To install the Sandbox License Server without installing the required Linux operating system dependencies, run the following command.

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

- To install the Sandbox License Server with installing the required Linux operating system dependencies, run the following command.

```
./zdt-license-server --install --dep
```

Step 2: Obtain an update file

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

1. Log in to the machine where you installed the Sandbox 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 Wazi for Red Hat CodeReady.
6. Select the license key name, for example, IBM WAZI FOR RED HAT CODEREADY WORKSPACES VIRTUAL PROCESSOR CORE. 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. Then, 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 Server Instances and Number of Licenses fields. 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 Sandbox 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.

Step 3: Apply the update file to the Sandbox License Server machine

After you obtain the update file, apply it to the Sandbox License Server machine to enable the license server.

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 previous step of obtaining an update file for the Sandbox 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 follows.

```
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


Thesse are the currently active sessions:

        KeyID         FID      FeatureName       Address      User      Machine         LoginTime            #
  432975633343422885 333     CPU License    9.26.158.161   ibmsys1    zdt-dev4    Mon Oct 30, 17:25:40    3
```

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

```
./uimserverstart
```

8. Install the Sandbox provisioning tools

Sandbox provisioning tools (shortened as provisioning tools) include a web UI with role-based access control and REST APIs. The 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 an image from components.
- Provision a sandbox instance from the selected image to Red Hat OpenShift.
- Manage and monitor components, images, and instances.

For more information, see the User's Guide of web UI and REST APIs.

9. Create your Sandbox instance

For more information, see:

- Configuring a Red Hat OpenShift target environment
- Provisioning instances to a Red Hat OpenShift environment

Installing Sandbox in an air-gapped environment

An air-gapped environment is a restricted environment where the company's network policy does not allow access to unsecured networks like the public internet or is behind a firewall hence requiring additional steps to pull the required images from a public registry like Red Hat® registry.

To install Sandbox in an air-gapped environment, you must first install the IBM Cloud Pak® foundational services by following the instructions at IBM® Catalog Management Plug-in for IBM Cloud® Paks (ibm-pak plugin).

Use the following CASE bundle commands to install Wazi Sandbox in an air-gapped environment.

Installation methods

For more information, see Installing your IBM Cloud Pak by mirroring Cloud Pak images to a private container registry (with ibm-pak plug-in).

The instructions in the preceding link explains both **connected mirroring** and **disconnected mirroring**. The difference between connected and disconnected mirroring procedures is outlined in the Mirror images section.

You can install Sandbox into one or more namespaces. For example, to install Sandbox into a namespace called `ibm-wazi-sandbox`, following the instructions in the preceding link, you must export the following:

```
export CASE_NAME=ibm-wazi-developer-sandbox
export CASE_VERSION=2.6.0
export NAMESPACE=ibm-wazi-sandbox
export CASE_INVENTORY_SETUP=waziSandboxOperatorSetup
```

Important: When you follow the instructions in the preceding links to install the catalog, you must choose the alternative way, which uses the `oc ibm-pak launch` command.

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
- Configuring network policies
- Configuring cluster persistent storage
- Configuring cluster network forwarding

- [Securing Sandbox in IBM Cloud](#): If you run Sandbox in IBM Cloud® or another public cloud, it is important to secure access to Sandbox to avoid exposing it on the public internet unintentionally.

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 for Dev Spaces 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 for VS Code that is running on developers' workstations
- IBM Wazi for Dev Spaces 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 subelements: `profile` and `scope`, which are described as follows:

Use `profile` to control what ports to expose.

Table 1. `profile` options

Profile	Description
`wazi`	Default. Includes encrypted ports of Wazi's host components.
`wazi-all`	Includes encrypted and unencrypted ports of Wazi's host components.
`custom`	Empty profile. No ports will be exposed by default unless they are given in `zPorts`.

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

Table 2. `scope` options

Scope	
`route`	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`	Default. Exposes ports only on the OpenShift cluster using a ClusterIP Service.

Note: Starting from Sandbox 2.1.0, the default scope option is changed from `route` to `cluster`. If you have been using `route` and want to continue using it, set it manually.
`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 subelements: `cluster`, `route`, and `nodeport`, which work similarly to the `scope`.

Table 3. Subelements of `zPorts`

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 subelements have `name` and `port` to identify the port. The `route` subelement also has a `tls` subelement to control how TLS is handled.

For more details, see [Configuration reference](#).

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 with the `WaziSandboxSystem` custom resource, which are shown as follows:

Table 4. NetworkPolicy for sandbox instances

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

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 allowlist. Some considerations when adding ingress rules:

- Traffic to `Route`s or `NodePort` services might be coming from OpenShift ingress controller pods. To allow this, configure a `namespaceSelector` that matches the label `network.openshift.io/policy-group: ingress`.
- The Sandbox Operator needs access to the pod for health status checking. To allow this, configure a `podSelector` that matches the label `app.kubernetes.io/name: ibm-wazi-developer-sandbox-operator`.
- Blocking incoming traffic to ephemeral ports, especially for UDP can cause prevent some services on the pod and on z/OS from functioning correctly. For example, if the Sandbox configuration references the SFTP server or License Server by name, or if you need to use DNS on z/OS, blocking DNS responses might cause failures.

Note: The policies above only manage ingress, not egress. If you want to control egress from the sandbox instance, you can manually create a `NetworkPolicy` to do this, but you must still allow traffic to at least the configured license server or servers on ports 1947 (TCP and UDP), and port 9451 (UDP), and the configured SFTP server at the configured port. Basic functioning of Sandbox and the z/OS instance might also require allowing DNS traffic (usually port 53) and its responses.

Due to limitations of `NetworkPolicy`, it might be easier to control ingress and egress at the cluster level, for example, with `EgressNetworkPolicy` or with a firewall on the host machines or between the cluster and other networks.

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, NFS, and additional storage providers and storage classes that are listed below.

Requirements:

- `volumeBindingMode: Immediate` mode
  Sandbox does not support storage classes that have the volume binding mode `WaitForFirstConsumer`. If you want to use a storage class that has this mode, create a new storage class with a different name, change the mode to `volumeBindingMode: Immediate`, and use the newly created storage class. For more information, see Volume Binding Mode.

- `ReadWriteOnce` access mode
  Volume files must not be used by more than one instance at a time or the z/OS volumes might corrupt.

Sandbox stores z/OS volumes for each instance on a `PersistentVolumeClaim`. When you plan for persistent volume storage, account for the situation where you will expand z/OS storage such as by adding extra volumes, and make the `PersistentVolumeClaim` large enough to accommodate this growth.

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. Consult your z/OS system administrator.
- 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.

Recommended storage providers

For Linux® on x86 hardware, the following storage providers are supported for Sandbox and recommended:

- OpenShift Container Storage 4.2 or later versions
- IBM Cloud Block storage and IBM Cloud File storage
- IBM Storage Suite for IBM Cloud Paks. This suite of offerings includes:
  - File storage from IBM Spectrum® Scale
  - Block storage from IBM Spectrum Virtualize, FlashSystem or DS8K

Supported storage classes in IBM Cloud

The following storage classes in IBM Cloud have been tested with Sandbox. High performance classes are recommended, because disk performance directly affects the emulator.

OpenShift (VPC)

- `ibmc-vpc-block-10iops-tier` (Recommended)
- `ibmc-vpc-block-5iops-tier`
- `ibmc-vpc-block-custom`
- `ibmc-vpc-block-general-purpose`

OpenShift (Classic)

- `ibmc-block-gold` (Recommended)
- `ibmc-block-silver`
- `ibmc-block-custom`
- `ibmc-file-gold-gid`

Notes:

- Storage classes with the reclaim policy `Retain` can work, but are unsupported because they can lead to data security issues.
- For Classic OpenShift, the file-based storage classes must be the `-gid` versions. And in the custom resource (CR) for the sandbox instance, you must specify the corresponding `fsGroup` (`spec.fsGroup`). For example, by default, this is 65531 for the `ibmc-file-gold-gid`.
- Slower storage classes can be deployed and work, but are unreliable due to the dependency of the sandbox instance to have high I/O to their disk volumes.

Using NFS storage

NFS is not recommended due to potential performance issues and possible accidental corruption of z/OS volume files on shared storage. NFS storage is not supported in the Sandbox web server UI.

If NFS storage is used, follow these steps strictly:

1. Enable the group write mode for the exported share on the NFS server, for example, by using the command: `chmod g+rwx /path/to/share`.
2. On the NFS server, note the numeric group ID of the share by using `ls -lnd` on the directory. In the following example, the numeric group ID is 4567.

   ```
   $ ls -lnd /path/to/share
   drwxrwxr-x 16 1234 4567 4096 Nov 10 20:04 /path/to/share
   ```

3. When using the exported share, manually create **PersistentVolume** and **PersistentVolumeClaim**.
4. For the **WaziSandboxVolumeCopy** and **WaziSandboxSystem** custom resources, set the **spec.fsGroup** parameter to the numeric group ID used by the NFS server for the exported share and discovered in step 2.

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 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.
- To simplify copying of a **PersistentVolumeClaim** that has cloud-ready z/OS volume images, consider installing OpenShift Virtualization. The included Containerized Data Importer can be used to create an efficient clone of a **PersistentVolumeClaim** or copy it based on the capabilities of the storage class.

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.

- 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 in both the **WaziSandboxVolumeCopy** and **WaziSandboxSystem** custom resources.

Configuring cluster network forwarding

Sandbox needs to forward network traffic to the z/OS instance, so the pods must allow IPv4 forwarding with the **net.ipv4.ip_forward**. If this is not the current default on the cluster, add **spec.zpdt.forceNetworkForwarding: true** to the **WaziSandboxSystem** custom resource, but this means that the sysctl must be added to the allowlist or the pod will fail and restart continuously.

For clusters that use **MachineConfig** and **MachineConfigPool**, do allow-listing by following the OpenShift documentation on enabling sysctls and allow the **net.ipv4.ip_forward**.

For clusters that do not use this mechanism, including managed Red Hat® OpenShift on IBM Cloud, you can add the sysctl to the allowlist by using a **DaemonSet** which will run an init container that configures the kubelet on the node to allow **ip_forward** to be enabled during a deployment. It will leave a container running which should be headless and only there to cause a redeploy if the node is rebooted.

To forward network traffic to the z/OS instance, take these steps:

1. Collect a list of nodes to target.

   ```
   [root@IBMC-Term1 ibmcloud]# oc get nodes
   NAME            STATUS    ROLES           AGE     VERSION
   10.48.27.207    Ready     master,worker   160d    v1.21.6+bb8d50a
   ```

```
    10.48.27.209    Ready    master,worker    160d    v1.21.6+bb8d50a
    10.48.27.227    Ready    master,worker    160d    v1.21.6+bb8d50a
```

2. With the name of nodes that need to have the whitelist enable, create a `node-0x.yaml` file for that node. Use the following YAML file and replace the <WORKER_NODE_HOSTNAME> with the name of the node listed in the `oc get nodes` command output.

```yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: modify-allowed-unsafe-sysctls-node1
  namespace: kube-system
  labels:
    app: modify-allowed-unsafe-sysctls-node1
spec:
  selector:
    matchLabels:
      name: modify-allowed-unsafe-sysctls-node1
  template:
    metadata:
      labels:
        name: modify-allowed-unsafe-sysctls-node1
    spec:
      nodeSelector:
        kubernetes.io/hostname: <WORKER_NODE_HOSTNAME>
      hostPID: true
      hostIPC: true
      initContainers:
        - image: alpine:3.6
          imagePullPolicy: IfNotPresent
          name: init
          command: ["/bin/sh", "-c"]
          args:
            - |
              grep allowedUnsafeSysctls /host/etc/kubernetes/kubelet.conf
              if [ $? -ne 0 ]; then
                echo -e '\nallowedUnsafeSysctls:\n- "net.ipv4.ip_forward"' >> /host/etc/kubernetes/kubelet.conf
                chroot /host systemctl restart kubelet
              fi
          securityContext:
            privileged: true
          volumeMounts:
            - name: host
              mountPath: /host
      containers:
        - image: registry.ng.bluemix.net/armada-master/pause:3.5
          imagePullPolicy: IfNotPresent
          name: sleepforever
          resources:
            requests:
              cpu: 0.01
      volumes:
        - name: host
          hostPath:
            path: /
```

3. Run the `oc apply -f node-01.yaml` command to apply the `node-0x.yaml` file into the cluster for each node that needs to be configured.

You can restrict the `DaemonSet` to selected nodes if you want, use the same selector or affinity with the `WaziSandboxSystem` custom resource to schedules sandbox instances to only those nodes.
On IBM Cloud with Classic infrastructure, you can also isolate the selected nodes so that they are not edge nodes.

Securing Sandbox in IBM Cloud

If you run Sandbox in IBM Cloud or another public cloud, you can configure Sandbox to use `NodePort` services to connect to z/OS for protocols that cannot be exposed using `Route`. You can also manually create a `LoadBalancer` service to connect.
Important: In cloud environments, it is important to secure access to these ports to avoid exposing it on the public internet unintentionally. Although these instructions focus on `NodePort`, the same advice applies to `Route` and `LoadBalancer`: these ports might be exposed by default, and you need to block access as described for `NodePort`.
The following instructions are for IBM Cloud, but similar methods and protections apply to other public clouds.

Red Hat OpenShift in IBM Cloud is available in several infrastructure options: Classic, VPC, and Satellite. Sandbox supports VPC (recommended) and Classic. Sandbox was not tested with Satellite.

To configure Sandbox in the supported infrastructures, refer to the following key points. For comprehensive information about the configuration and security of Classic and VPC, see IBM Cloud documentation.

Classic and VPC
    In either Classic or VPC infrastructure, there are protocols that do not work with `Route` blocking the public access to `NodePort`. To use these protocols securely outside of IBM Cloud, for example with an IDE on your laptop, you must set up and use a VPN.
    Note: You need to follow the instructions in the Configuring cluster network forwarding section for Sandbox to function correctly in IBM Cloud.
VPC
    VPC clusters that are created in the console have public and private endpoints, and cannot be later converted to private clusters as noted in Creating a standard VPC cluster in the console. To create a private cluster, create the cluster with the CLI and use the `--disable-public-service-endpoint` flag by following the instructions at Creating standard VPC clusters from the CLI.

    On VPC clusters, worker nodes are only connected to VPC subnets. So your `NodePort` services are private by default as described in Network segmentation and privacy for VPC clusters.

    For more information about creating a VPN to access the services, see VPNs for VPC overview and Setting up VPC VPN connectivity.

Classic
> Classic clusters have public endpoints by default and do not have private endpoints according to [Creating a standard classic cluster in the console](#) and the `--private-service-endpoint` option in [Creating a standard classic cluster in the CLI](#). This means that `NodePort` services are opened to the public internet by default. For more information about securing traffic, see [Network segmentation and privacy for classic clusters](#) and [Controlling inbound traffic to NLB or NodePort services](#).
>
> To use a VPN to connect securely, see [Setting up classic VPN connectivity](#).

Authorizing users to Wazi Sandbox by using role-based access control (RBAC)

You can authorize access to IBM® Wazi Sandbox resources by using the Kubernetes and Red Hat® OpenShift® Container Platform role-based access control (RBAC) facilities.

RBAC objects, including rules, roles, and role bindings, allow you to define permissions and assign those permissions to users. Rules define the actions that are applied to specified resources in an OpenShift Container Platform cluster. A `Role` or `ClusterRole` resource contains a set of rules that represent permissions and can be bound to users or groups. In turn, a `RoleBinding` or `ClusterRoleBinding` resource respectively associates users or groups with a created `Role` or `ClusterRole` resource at the namespace or cluster scope. For more information about RBAC for OpenShift Container Platform, see [Using RBAC to define and apply permissions](#) in the OpenShift Container Platform documentation.

Important: To install the Wazi Sandbox Operator, you must be granted the `cluster-admin ClusterRole` by using a `ClusterRoleBinding`. Only a `cluster-admin` role user can grant other users the `cluster-admin` access. For more information about how to add role-based control to a user, see [Creating a cluster admin](#) in the OpenShift Container Platform documentation.

Note: Users who have `cluster-admin` privileges do not need to apply the examples below because they already have authority to create all resources within OpenShift Container Platform.

1. Define rules and roles in YAML files.

The following three RBAC policies give varying levels of access:

`wazi-sandbox-admin`
> The `wazi-sandbox-admin` role is the full authorization needed to create, use, delete, and troubleshoot sandbox instances. It is not as privileged as `cluster-admin`, but it can create and view secrets, use significant CPU and storage resources, and exec into a privileged pod so it should only be granted to trusted users. It allows users to:
>
> - View a specified project in OpenShift Container Platform.
> - View workloads (`StatefulSet`, `Job`, and `Pod`) that are running in a project, including pod status and logs.
> - View installed operators in a project.
> - View created custom resource definitions (`CustomResourceDefinition`).
> - Create, read, update, and delete Wazi Sandbox instances or volume copies (`WaziSandboxSystem`, `WaziSandboxVolumeCopy`) and all related prerequisites and configuration resources (`ConfigMap`, `NetworkPolicy`, `PersistentVolumeClaim`, `Route`, `Service`, and `Secret`).
> - Exec into pods.
>
> To define the role and rules, customize the following template:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: wazi-sandbox-admin
rules:
  - apiGroups:
      - apiextensions.k8s.io
    resources:
      - customresourcedefinitions
    verbs:
      - get
  # Needed to view OCP projects UI
  - apiGroups:
      - project.openshift.io
    verbs:
      - get
    resources:
      - projects
  # Needed to view pods, logs, services, pvcs and general status - no create/update/exec
  # Used to see resources related to Wazi Sandbox custom resources
  - apiGroups:
      - ""
    verbs:
      - get
      - list
      - watch
    resources:
      - configmaps
      - pods
      - pods/log
      - pods/status
      - events
      - namespaces
      - namespaces/status
      - persistentvolumeclaims
      - secrets
      - services
  # Needed to view statefulsets
  # Used to see resources related to Wazi Sandbox System custom resource
  - apiGroups:
      - apps
```

```yaml
      verbs:
        - get
        - list
        - watch
      resources:
        - statefulsets
  # Needed to view jobs
  # Used to see resources related to Wazi Sandbox Volume Copy custom resource
  - apiGroups:
      - batch
    verbs:
      - get
      - list
      - watch
    resources:
      - jobs
  # Needed to edit configmaps, pvcs, secrets, services
  # Used to create/edit resources related to Wazi Sandbox custom resources
  - apiGroups:
      - ""
    verbs:
      - create
      - delete
      - patch
      - update
    resources:
      - configmaps
      - persistentvolumeclaims
      - services
      - secrets
  # Needed to exec into pods (for troubleshooting)
  - apiGroups:
      - ""
    verbs:
      - create
    resources:
      - pods/exec
  # Needed to edit network policies
  # Used to create/edit resources related to Wazi Sandbox custom resources
  - apiGroups:
      - networking.k8s.io
    verbs:
      - get
      - list
      - watch
      - create
      - delete
      - patch
      - update
    resources:
      - networkpolicies
  # Needed to edit routes
  # Used to create/edit resources related to Wazi Sandbox System custom resource
  - apiGroups:
      - route.openshift.io
    verbs:
      - get
      - list
      - watch
      - create
      - delete
      - patch
      - update
    resources:
      - routes
  # Needed to view Installed Operators
  - apiGroups:
      - operators.coreos.com
    resources:
      - clusterserviceversions
      - catalogsources
      - installplans
      - subscriptions
      - operatorgroups
    verbs:
      - get
      - list
      - watch
  # Needed for working with Wazi Sandbox custom resources
  - apiGroups:
      - sandbox.wazi.ibm.com
    resources:
      - wazisandboxsystems
      - wazisandboxvolumecopies
    verbs:
      - get
      - list
      - watch
      - create
      - delete
      - patch
      - update
```

wazi-sandbox-edit

The **wazi-sandbox-edit** role is more restricted. Users with this role can still create and use sandbox instances, but cannot create or view secrets or exec into the sandbox instance pod. This means that to transfer z/OS® volumes into the cluster from an SFTP server, a more privileged user such as one with **wazi-sandbox-admin** or **cluster-admin** role would need to create the **Secret** with the SFTP server password. Also, some troubleshooting actions will not be possible since access to the pod is limited. Users with this authorization can still use significant CPU and storage resources. It allows users to:

- View a specified project in OpenShift Container Platform.
- View workloads (**StatefulSet**, **Job**, and **Pod**) that are running in a project, including pod status and logs.
- View installed operators in a project.
- View created custom resource definitions (**CustomResourceDefinition**).
- Create, read, update, and delete Wazi Sandbox instances or volume copies (**WaziSandboxSystem**, **WaziSandboxVolumeCopy**) and some less sensitive related prerequisites and configuration resources (**ConfigMap**, **PersistentVolumeClaim**, **Route**, **Service**).

To define the role and rules, customize the following template:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: wazi-sandbox-edit
rules:
 - apiGroups:
    - apiextensions.k8s.io
   resources:
    - customresourcedefinitions
   verbs:
    - get
 # Needed to view OCP projects UI
 - apiGroups:
    - project.openshift.io
   verbs:
    - get
   resources:
    - projects
 # Needed to list network policies associated with Wazi Sandbox custom resources
 - apiGroups:
    - networking.k8s.io
   verbs:
    - list
   resources:
    - networkpolicies
 # Needed to view configmaps, pods, logs, services, pvcs and general status - no create/update/exec
 # Used to view resources related to Wazi Sandbox custom resources
 - apiGroups:
    - ""
   verbs:
    - get
    - list
    - watch
   resources:
    - configmaps
    - pods
    - pods/log
    - pods/status
    - events
    - namespaces
    - namespaces/status
    - persistentvolumeclaims
    - services
 # Needed to view statefulsets
 # Used to view resources related to Wazi Sandbox System custom resource
 - apiGroups:
    - apps
   verbs:
    - get
    - list
    - watch
   resources:
    - statefulsets
 # Needed to view routes
 # Used to view resources related to Wazi Sandbox System custom resource
 - apiGroups:
    - route.openshift.io
   verbs:
    - get
    - list
    - watch
   resources:
    - routes
 # Needed to view jobs
 # Used to view resources related to Wazi Sandbox Volume Copy custom resource
 - apiGroups:
    - batch
   verbs:
    - get
    - list
    - watch
   resources:
    - jobs
 # Needed to edit pvcs
 - apiGroups:
    - ""
   verbs:
    - create
    - delete
    - patch
```

```
            - update
        resources:
            - persistentvolumeclaims
    # Needed to view Installed Operators
    - apiGroups:
        - operators.coreos.com
      resources:
        - clusterserviceversions
        - catalogsources
        - installplans
        - subscriptions
        - operatorgroups
      verbs:
        - get
        - list
        - watch
    # Needed for working with Wazi Sandbox custom resources
    - apiGroups:
        - sandbox.wazi.ibm.com
      resources:
        - wazisandboxsystems
        - wazisandboxvolumecopies
      verbs:
        - create
        - update
        - get
        - list
        - watch
        - delete
```

wazi-sandbox-view

The `wazi-sandbox-view` role is the minimum authorization to use a sandbox instance, but users with this role cannot create or delete sandbox instances. The instance would need to be created by a user with higher authority, for example, `wazi-sandbox-admin`, `wazi-sandbox-edit`, or `cluster-admin`. It allows users to:

- View a specified project in OpenShift Container Platform.
- View workloads (`StatefulSet`, `Job`, and `Pod`) that are running in a project, including pod status and logs.
- View installed operators in a project.
- View created custom resource definitions (`CustomResourceDefinition`).
- View Wazi Sandbox instances (`WaziSandboxSystem`) and resources required to use them (`Route`, `Service`).

To define the role and rules, customize the following template:

```
# This role allows use of Wazi Sandbox instances, but not creation or
# deletion.
#
# It is suited for a scenario where another user with the `wazi-sandbox-edit`
# or `wazi-sandbox-admin` role creates the Wazi Sandbox instances for use by
# general users.
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: wazi-sandbox-view
rules:
  - apiGroups:
      - apiextensions.k8s.io
    resources:
      - customresourcedefinitions
    verbs:
      - get
  # Needed to view OCP projects UI
  - apiGroups:
      - project.openshift.io
    verbs:
      - get
    resources:
      - projects
  # Needed to list resources associated with Wazi Sandbox custom resources for
  # general status
  - apiGroups:
      - ""
    verbs:
      - list
    resources:
      - persistentvolumeclaims
  # Needed to list statefulsets
  # Used to view resources related to Wazi Sandbox System custom resource
  - apiGroups:
      - apps
    verbs:
      - list
    resources:
      - statefulsets
  # Needed to list jobs
  # Used to view resources related to Wazi Sandbox Volume Copy custom resource
  - apiGroups:
      - batch
    verbs:
      - list
    resources:
      - jobs
  # Needed to view pods logs and status, and use services - no create/update/exec
  # Used to view resources related to Wazi Sandbox custom resources
  - apiGroups:
```

```
        - ""
    verbs:
      - get
      - list
      - watch
    resources:
      - pods
      - pods/log
      - pods/status
      - events
      - namespaces
      - namespaces/status
      - services
# Needed to view and use routes
# Used to view resources related to Wazi Sandbox System custom resource
- apiGroups:
    - route.openshift.io
  verbs:
    - get
    - list
    - watch
  resources:
    - routes
# Needed to view Installed Operators
- apiGroups:
    - operators.coreos.com
  resources:
    - clusterserviceversions
    - catalogsources
    - installplans
    - subscriptions
    - operatorgroups
  verbs:
    - get
    - list
    - watch
# Needed to view Wazi Sandbox custom resources
- apiGroups:
    - sandbox.wazi.ibm.com
  resources:
    - wazisandboxsystems
  verbs:
    - get
    - list
    - watch
```

2. Apply the roles that you created in the preceding step to your OpenShift Container Platform cluster, and bind the roles to users.

Important: You must either be a `cluster-admin` to perform the following steps or have RBAC privileges to create role bindings for users or groups, or both.

1. Log in to OpenShift Container Platform.
   a. Log in to the OpenShift Container Platform web console for your cluster. Then, click your username in the upper-right corner, and then click Copy Login Command.
   b. In the new browser tab that opens, click Display Token. Then, copy the command that is displayed under Log in with this token section.
   c. In a command prompt, paste the command that you copied in the preceding step to log in to OpenShift Container Platform.
2. Apply the role to OpenShift Container Platform. This procedure assumes that the role is `wazi-sandbox-view` and the YAML file is named `wazi-sandbox-view.yaml`, but you might be applying a different role or specified a different file name.

   `oc create -f wazi-sandbox-view.yaml`

3. Bind the role to a user, and limit the scope of the role to a namespace:

   ```
   oc project <project-name>
   oc adm policy add-role-to-user wazi-sandbox-view <user-name>
   ```

Related information

- Using RBAC to define and apply permissions in the OpenShift Container Platform documentation.

Bring Your Own Certificate (BYOC)

Using custom certificates

The Instance Controller in the sandbox instance pod uses an automatically generated service serving certificate by default, but you can bring your own certificate.

To use a custom certificate,

1. Create a key/value `Secret` of `type: kubernetes.io/tls` in the namespace where you will deploy your sandbox instance. The `Secret` must have two keys: `tls.crt` and `tls.key`. The value for each is base 64-encoded PEM certificate data. The value of `tls.key` is the private key for the certificate. The value of `tls.crt` is a certificate or certificate chain. For example:

   ```
   kind: Secret
   apiVersion: v1
   metadata:
   ```

```
    name: custom-certificate
    namespace: my-namespace
data:
  tls.crt: >-
    _base64-encoded PEM certificate chain data_
  tls.key: >-
    _base64-encoded PEM key data_
type: kubernetes.io/tls
```

2. In your `WaziSandboxSystem` custom resource, set `controller.secretName` to the name of the `Secret`, for example:

```
apiVersion: sandbox.wazi.ibm.com/v1alpha1
kind: WaziSandboxSystem
metadata:
  name: sandbox-example
  namespace: my-namespace
spec:
  license:
    accept: true
  controller:
    secretName: custom-certificate
  zpdt:
    configMap:
      name: sandbox-configmap
    secretName: sandbox-secret
  zVolumes:
    existingClaimName: sandbox-pvc
```

Note: Sandbox does not manage any certificates on the z/OS running in the sandbox instance. Manage z/OS certificates as you normally do.

Rotating keys

- To rotate the key for the default generated service serving certificate, delete the generated `Secret`. It will have a name similar to the `WaziSandboxSystem` name, but with the suffix `service-cert`, for example, `my-sandbox-service-cert` for a sandbox instance named `my-service`.
- To rotate a custom key, delete or update the `Secret` that you created.

In order for the sandbox instance to pick up the changed `Secret` in either case, delete the pod, and it will restart automatically and use the new key. As this can interrupt sessions on z/OS, it is strongly recommended that you manually shut down z/OS before deleting the pod.

Configuration reference

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

- Custom resource definitions
- Network flow

Custom resource definitions

Use the following information to learn about the custom resource definitions (CRDs) used by IBM Wazi Sandbox.

- IBM Wazi Sandbox custom resource definitions

IBM Wazi Sandbox custom resource definitions

Use this information to learn about the custom resource definitions (CRDs) used by IBM Wazi Sandbox.

In the following table, CRDs that do not have a hyperlink must not be manually edited unless otherwise instructed, and are manipulated only by the operator.

Table 1. Sandbox CRDs

Kind	API Version	Operator Name	Description
WaziSandboxSystem	v1alpha1	ibm-wazi-sandbox	Personal z/OS® sandbox system for development and testing.
WaziSandboxVolumeCopy	v1alpha1	ibm-wazi-sandbox	Copy z/OS® volume files for use with Sandbox.

- **WaziSandboxSystem/v1alpha1**
- **WaziSandboxVolumeCopy/v1alpha1**

WaziSandboxSystem/v1alpha1

Personal z/OS® sandbox system for development and testing.

spec

Description: WaziSandboxSystem is the schema for a sandbox instance.

Type: object

Table 1. Description of WaziSandboxSystem/v1alpha1 spec

Property	Type	Description
spec	object	WaziSandboxSystem is the schema for a sandbox instance.
spec.affinity	object	An optional Pod spec.affinity with scheduling constraints for the sandbox instance.
spec.controller	object	Configure the provisioning tools Instance Controller
spec.controller.ciphers	string	A space-separated list of TLSv1.3 cipher suites that the Instance Controller will use. An empty or null string will use the default list of all the TLSv1.3 cipher suites that the Instance Controller supports: "TLS_AES_256_GCM_SHA384 TLS_AES_128_GCM_SHA256"
spec.controller.expose	string	Choose how to expose the Instance Controller. "cluster": Adds it to a ClusterIP Service. "route": Default. Adds it to a ClusterIP Service and a Route. "nodeport": Adds it to a NodePort Service with no Route.
spec.controller.tls	object	If exposing a Route, values here will be used for the "tls" configuration of the Route.
spec.controller.secretName	string	Use a custom certficate for the Instance Controller. If set, an existing Secret that must have two keys "tls.crt" and "tls.key" that contain the certificate and key to be used, respectively.
spec.fsGroup	integer	Supplemental group ID to mount z/OS volume file claim.
spec.license	object	License acceptance.
spec.license.accept	boolean	To install this product, the license agreement http://ibm.biz/wazi-developer-license must be accepted. This field is required. 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.
spec.licensedProduct	string	Licensed product being used with Sandbox.
spec.networkPolicy	string	Create a NetworkPolicy for the sandbox instance. "allow-all": Default. 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.
spec.nodeSelector	object	An optional Pod spec.nodeSelector which must match a node's labels for the sandbox instance to be scheduled on that node.
spec.persistence	object	Use dynamic provisioning for storage.
spec.persistence.useDynamicProvisioning	boolean	If true (default), dynamic provisioning will be used if creating a PersistentVolumeClaim.
spec.portProfile	object	Configure what ports to expose and how to expose them.
spec.portProfile.profile	string	Select a port profile to use. "wazi": Default profile. 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.
spec.portProfile.scope	string	Choose how to expose the ports specified in the profile. If you chose the "custom" profile that has no ports, ignore this setting. "cluster": Default. Exposes ports only on the OpenShift cluster using a ClusterIP Service. "route": 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.
spec.resources	object	Compute resources required by the sandbox instance. The defaults are appropriate for the Extended ADCD default configuration, but if the default devmap is changed or if using a custom image, the resources must be sized appropriately. If the devmap uses N CPs, N+1 CPUs should be configured here, and the memory resource should be 25% higher than the memory configured in the devmap.
spec.resources.limits	object	Limits describes the maximum amount of compute resources allowed.
spec.resources.limits.cpu	string	The maximum number of CPU cores to use. Default: 4
spec.resources.limits.memory	string	The maximum size of memory to use. Default: 10Gi
spec.resources.limits.ephemeral-storage	string	The maximum amount of ephemeral storage to use. Default: 2Gi
spec.resources.requests	object	Requests describes the minimum amount of compute resources required.
spec.resources.requests.cpu	string	The number of CPU cores to request. Default: 3.5
spec.resources.requests.memory	string	The size of memory to request. Default: 8Gi
spec.resources.requests.ephemeral-storage	string	The amount of ephemeral storage to request. Default: 1Gi
spec.service	object	DEPRECATED: use zPorts
spec.service.customPorts	array	Expose arbitrary custom ports on z/OS.
spec.service.exposeEclipseClientPorts	boolean	Expose the z/OS ports required for IBM Eclipse clients other than Z Open Development 2.0.1 (rsed, rse-server, debug-manager, debug-profile-service). Default: true
spec.service.exposeSSHPort	boolean	Expose the z/OS Secure Shell port as name "ssh". Default: true
spec.service.exposeTN3270SecurePort	boolean	Expose the z/OS TN3270 with TLS port as name "tn3270-secure". Default: true

Property	Type	Description
spec.service.exposeZOSConnectPort	boolean	Expose the z/OS Connect port as name "zosconnect". Default: true
spec.service.exposeLocalVTAMPort	boolean	Expose the unsecured Local VTAM (main console) port as name "local-vtam". Default: false
spec.service.exposeZOSMFPort	boolean	Expose the z/OS Management Facility port as name "zosmf". Default: true
spec.service.exposeVSCodeClientPorts	boolean	Expose the z/OS ports required for VS Code clients (remote-debug-service, rse-api, debug-profile-service). Default: true
spec.service.exposeFTPPort	boolean	Expose the unsecured z/OS FTP port as name "ftp". Default: false
spec.service.exposeZOD201ClientPorts	boolean	Expose the z/OS ports (including unsecured ports) required for IBM Z Open Development 2.0.1 (rsed, rse-server, debug-manager, dtcn-cicsts54, dtcn-cicsts55). Default: false
spec.service.exposeTN3270UnsecuredPort	boolean	Expose the unsecured z/OS TN3270 as name "tn3270-unsecured". Default: false
spec.tolerations	array	An optional Pod spec.tolerations that will affect scheduling of the sandbox instance.
spec.zPorts	object	Control exposed ports on the z/OS instance.
spec.zPorts.cluster	array	Ports to be exposed with a ClusterIP Service
spec.zPorts.nodeport	array	Ports to be exposed with a NodePort Service
spec.zPorts.route	array	Ports to be exposed with a ClusterIP Service and create Routes for
spec.zVolumes	object	Configuration for the PersistentVolumeClaim (PVC) used to store z/OS volume files. Sandbox can use an existing claim or allocate a new one. If Sandbox allocates the PVC, that PVC will be deleted when the WaziSandboxSystem is deleted.
spec.zVolumes.existingClaimName	string	Select an existing PersistentVolumeClaim to store z/OS volume files. Otherwise, create a new PersistentVolumeClaim by expanding "Advance configuration" and completing the fields. If you select an existing PersistentVolumeClaim, the input in "Advance configuration" will be ignored.
spec.zVolumes.name	string	Optional: The name for the PersistentVolumeClaim to be created for storing z/OS volume files
spec.zVolumes.selector	object	A PersistentVolumeClaim spec.selector that is a label query over volumes to consider for binding.
spec.zVolumes.selector.label	string	Optional: The selector label for creating the PersistentVolumeClaim
spec.zVolumes.selector.value	string	Optional: The selector value for creating the PersistentVolumeClaim
spec.zVolumes.size	string	The size of the PersistentVolumeClaim to be created. Default: 300Gi
spec.zVolumes.storageClassName	string	The storage class name for the PersistentVolumeClaim to be created. This might be provided by the cluster administrator.
spec.zpdt	object	Configuration of the Z hardware emulator.
spec.zpdt.configMap	object	A ConfigMap for the Z hardware emulator. If spec.zpdt.configMap.create is true, the properties here will be used to create a ConfigMap, Otherwise the name property must name an existing ConfigMap which can contain the same properties.
spec.zpdt.configMap.licenseServer	string	The address or resolvable hostname of the primary License Server
spec.zpdt.configMap.devmap	string	The path to the emulator machine characteristics and device map file, relative to the mounted z/OS volumes
spec.zpdt.configMap.iplIODFAddress	string	The volume that contains the IODF
spec.zpdt.configMap.shutdownTimeout	integer	How long to wait for z/OS shutdown to complete (seconds)
spec.zpdt.configMap.name	string	The existing ConfigMap to be used. If you do not create a new ConfigMap, the value is required.
spec.zpdt.configMap.iplAddress	string	The IPL volume for z/OS, also known as SYSRES
spec.zpdt.configMap.sftpPort	string	The port of the SFTP server
spec.zpdt.configMap.secondaryLicenseServer	string	Optional: The address or resolvable hostname of the secondary License Server
spec.zpdt.configMap.sftpPath	string	The path to the directory that contains the z/OS volume files on the SFTP server
spec.zpdt.configMap.shutdownCommand	string	The z/OS command to initiate shutdown

Property	Type	Description
spec.zpdt.config Map.iplLoadPar m	string	The 2-character suffix of the LOADxx member of SYS1.IPLPARM
spec.zpdt.config Map.sftpHost	string	The address or resolvable hostname of the SFTP server
spec.zpdt.config Map.iplZOS	string	If set to true, z/OS is IPLed when the sandbox instance is created.
spec.zpdt.config Map.sftpUser	string	The user to log in to the SFTP server
spec.zpdt.config Map.create	boolean	Create a ConfigMap that contains configuration details about the License Server and SFTP server. If set to false, select an existing ConfigMap from the drop-down menu. If set to true, complete the fields. Default: false
spec.zpdt.copyB lockSize	integer	Optional: The size in bytes of blocks used when copying z/OS volume files
spec.zpdt.copyT hreads	integer	Optional: The number of concurrent copies to perform when copying z/OS volume files
spec.zpdt.copyV olumes	boolean	Copy z/OS volume files before starting the sandbox instance. Default: true. For faster provisioning, use WaziSandboxVolumeCopy to copy volumes instead.
spec.zpdt.force NetworkForwar ding	boolean	Use sysctl to enable IPv4 forwarding. Enable this only if the cluster admin has added the net.ipv4.ip_forward sysctl to the allowlist and network connections to z/OS are not working. If the sysctl is not in the allowlist, Pods will be respawning rapidly, consuming cluster resources, and the sandbox will not start.
spec.zpdt.secret Name	string	The existing Secret that contains keys named "sftpPassword" and "zdtAuthPassword"

status

Description: Status defines the observed state of the WaziSandboxSystem.

Type: object

Table 1. Description of WaziSandboxSystem/v1alpha1 status

Property	Type	Description
status	object	Status defines the observed state of the WaziSandboxSystem.
status.versions	array	Versions defines the operator version.
status.conditions	array	Condition represents the state of the operator's reconciliation.

WaziSandboxVolumeCopy/v1alpha1

Copy z/OS® volume files for use with Sandbox.

spec

Description: WaziSandboxVolumeCopy is the schema for a job to copy z/OS volume files.

Type: object

Table 1. Description of WaziSandboxVolumeCopy/v1alpha1 spec

Property	Type	Description
spec	object	WaziSandboxVolumeCopy is the schema for a job to copy z/OS volume files.
spec.nodeSelect or	object	An optional Pod spec.nodeSelector which must match a node's labels for the sandbox instance to be scheduled on that node.
spec.resources	object	Compute resources for the z/OS volume copy job. The cpu should be sized for the spec.copyThreads parameter, and the memory should be spec.copyBlockSize times spec.copyThreads plus 8GB.
spec.resources.l imits	object	Limits describes the maximum amount of compute resources allowed.
spec.resources.l imits.cpu	string	The maximum number of CPU cores to use. Default: 4
spec.resources.l imits.memory	string	The maximum size of memory to use. Default: 10Gi
spec.resources.l imits.ephemeral -storage	string	The maximum amount of ephemeral storage to use. Default: 10Mi
spec.resources.r equests	object	Requests describes the minimum amount of compute resources required.
spec.resources.r equests.cpu	string	The number of CPU cores to request. Default: 1
spec.resources.r equests.memor y	string	The size of memory to request. Default: 2Gi

Property	Type	Description
spec.resources.requests.ephemeral-storage	string	The amount of ephemeral storage to request. Default: 2Mi
spec.affinity	object	An optional Pod spec.affinity with scheduling constraints for the copy job.
spec.copyTimeout	integer	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. Note: The copy might still be proceeding as expected. Check the log of the Pod.
spec.licensedProduct	string	Licensed product being used with Sandbox.
spec.destination	object	Provide information about the destination of z/OS volume files to be copied.
spec.destination.config	object	Destination configuration information. Ignored if an existing destination ConfigMap is used.
spec.destination.config.pvcPath	string	Currently unused and ignored. The path to the directory on the PersistentVoumeClaim that will hold the z/OS volume files. If "destination.existingConfigName" is set, skip this field.
spec.destination.existingClaimName	string	Select an existing PersistentVolumeClaim that will hold the z/OS volume files. Otherwise, create a new PersistentVolumeClaim by expanding "Advance configuration" and completing the fields. If you select an existing PersistentVolumeClaim, the details in "Advance configuration" will be ignored.
spec.destination.existingConfigName	string	Name of the ConfigMap that has configuration details in the form of "destination.config". Optional. If specified, "destination.config" will be ignored.
spec.destination.selector	object	A PersistentVolumeClaim spec.selector that is a label query over volumes to consider for binding.
spec.destination.selector.label	string	Optional: The selector label for creating the PersistentVolumeClaim
spec.destination.selector.value	string	Optional: The selector value for creating the PersistentVolumeClaim
spec.destination.size	string	The size of the PersistentVolumeClaim to be created. Default: 300Gi
spec.destination.storageClassName	string	The storage class name for the PersistentVolumeClaim to be created. The name might be provided by the cluster administrator.
spec.destination.type	string	The type of the destination of volume files. Only "pvc" is valid currently.
spec.manifest	object	Provide volume manifest information for the copy.
spec.manifest.contents	array	For type "inline", the volumes to be copied.
spec.manifest.contentsFile	string	For the type "text" or "checksum", specify the path from source to the file. For the type "checksum", the file is in the format produced by the "md5sum" or "sha256sum" commands. For the type "json", it is in the format of a JSON array the same as in the contents object.
spec.manifest.decompress	boolean	Optional: Decompress files that end with ".gz" in the source when writing to the destination. Default: true
spec.manifest.digestAlgorithm	string	Select the default algorithm for volume digests if not specified with the volume. If you selected "none" for the Digests section, ignored this setting. Required for the manifest type "checksum". Default: sha256.
spec.manifest.digestValidation	string	Optional: Select how to validate digests. If you selected "none" for the Digests section, ignored this setting. "ignore": No validation, and ignores any digests in the manifest. "warn": Warns missing digests, or causes 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. Default: verify.
spec.manifest.digests	string	Optional: Select where to calculate volume digests. Default: both
spec.manifest.type	string	Optional: Select the type of manifest. The "text", "checksum", and "json" types refer to a file found at the source, whose named is specified in the "Contents file" field. The "inline" type uses the content object to provide the information inline. Default: text
spec.fsGroup	integer	Supplemental group ID to mount z/OS volume file claim.
spec.copyThreads	integer	Optional: The number of concurrent copies to perform when copying z/OS volume files
spec.copyBlockSize	integer	Optional: The size in bytes of blocks used when copying z/OS volume files
spec.source	object	Provide information about the source of z/OS volume files to be copied.
spec.source.config	object	If you specified an existing ConfigMap, skip this section.
spec.source.config.pvcPath	string	Currently unused and ignored. For source type "pvc" the path to the directory on the PersistentVolumeClaim with the z/OS volume files.
spec.source.config.sftpHost	string	The address or resolvable hostname of the SFTP server
spec.source.config.sftpPath	string	The path to the directory on the SFTP server with the z/OS volume files
spec.source.config.sftpPort	string	Optional: The port of the SFTP server. Default: 22
spec.source.config.sftpUser	string	The user to log in to the SFTP server
spec.source.existingClaimName	string	Required for source type "pvc". Select an existing PersistentVolumeClaime that has the source z/OS volumes.

Property	Type	Description
spec.source.exis tingConfigName	string	Optional: Select an existing ConfigMap that has configuration details in the form of "source.config". Otherwise, create a new ConfigMap by expanding "Advanced configuration" and completing the "Source configuration details" form. If you select an existing ConfigMap, the "Source configuration details" form will be ignored.
spec.source.sec retName	string	Required for the source type "sftp". Select an existing Secret that has the SFTP password in a key named "sftpPassword".
spec.source.typ e	string	Optional: Select the type of the source of z/OS volume files. "sftp": Copy from an SFTP server. "pvc": Copy from an existing PersistentVolumeClaim. Default: sftp
spec.tolerations	array	An optional Pod spec.tolerations that will affect scheduling of the copy job.
spec.persistenc e	object	Use dynamic provisioning for storage.
spec.persistenc e.useDynamicPr ovisioning	boolean	If true (default), dynamic provisioning will be used if creating a PersistentVolumeClaim.

status

Description: Status defines the observed state of the WaziSandboxVolumeCopy.

Type: object

Table 1. Description of WaziSandboxVolumeCopy/v1alpha1 status

Property	Type	Description
status	object	Status defines the observed state of the WaziSandboxVolumeCopy.
status.versions	array	Versions defines the operator version.
status.conditions	array	Condition represents the state of the operator's reconciliation.

Required firewall ports for network flow

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

In the following tables, communication is initiated from the client using an ephemeral port to connect to the server through the corresponding server port. The ephemeral port on the client is randomly assigned, depending on what's available on the client machine at the time. After a connection is established, the open server port is used for both inbound and outbound traffic.

Server: source z/OS

Server port	Client	Connection type	Protocol	Port usage	Configurable
22	Web server	SSH	TCP	For the connection from the web server to the source z/OS.	Yes. You can use another port for the connection from the web server to the source z/OS. Make sure the right port number is specified on the web server user interface (UI).

Server: web server

Server port	Client	Connection type	Protocol	Port usage	Configurable
22	Web server machine	SSH	TCP	For the installation and maintenance of the web server.	Yes. This port is configurable by the machine administrator.
9443	Browser or REST API client	SSL	TCP	For accessing the web server UI or REST APIs through a secure connection by using an address like https://*host-name*:9443.	Yes. This port can be configured during the installation of the web server.

Server: license server

Server port	Client	Connection type	Protocol	Port usage	Configurable
22	License server machine administrator	SSH	TCP	For the installation and maintenance of the license server.	Yes. This port is configurable by the machine administrator.
1947	Web server or target environment	Encrypted proprietary communication	TCP/UDP	For software licensing.	No
9451	Web server or target environment	Encrypted proprietary communication	TCP	For Unique Identity Manager (UIM) licensing.	No

Server: storage server

Server port	Client	Connection type	Protocol	Port usage	Configurable
22	Web server or target environment	SFTP	TCP	For transferring per-built Extended ADCD files and storing extracted files.	Yes. You can use another port for the connection. Make sure the right port number is specified on the web server UI.

Server: target environment

Server port	Client	Connection type	Protocol	Port usage	Configurable
22	Web server	SSH	TCP	For connecting from the web server to perform the provisioning of a z/OS instance on the target environment.	Yes. You can use another port for the connection from the web server to the target emulator. Make sure the right port number is specified on the web server UI.

Server port	Client	Connection type	Protocol	Port usage	Configurable
2022	z/OS UNIX System Services user	SSH	TCP	For communication from a z/OS user to the z/OS system on the emulator from outside of the target environment.	No
3270	z/OS TN3270 user	TN3270E	TCP	For communication from a z/OS user to the z/OS system on the emulator.	No

Notes:

- This table is not applicable to Red Hat OpenShift target environments. See section Additional ports exposed by Sandbox instead.
- During provisioning to a Linux target environment, the process automatically configures the ports and firewall rules on the target environment, with the exception of port 22. Port 22 needs to be enabled prior to the provisioning process to prevent potential errors when provisioning an instance from a created image.
- For other z/OS ports used for custom purposes, manual firewall rule configuration is required.

Additional ports exposed by Sandbox

Additionally, if you want to install Sandbox for provisioning to Red Hat OpenShfit target environments, it will create services that expose the following ports. You need to add these ports to the WaziSandboxSystem custom resource (CR) and modify firewall configurations accordingly.

Service	From	To	Port	Protocol	Function
zdt-instance-controller	Web server UI	Instance Controller REST Server	8443	HTTPS	Standard communication from the web server UI to the managed Sandbox instance

The Sandbox instance can initiate outbound connections from the pod to set up and manage the instance:

From	To	Port	Protocol	Function
Pod SFTP client	Storage server	Configurable (default: 22, see `sftpPort`)	SFTP	Transfer z/OS® volume files for the Sandbox instance.
Pod License client	License server	1947	TCP,UDP	Validate licensing for the Z hardware emulator.
Pod UIM client	UIM server	9451	TCP	Generate unique Z serial numbers for licensing.

The created service or services might also expose the following named ports on z/OS based on the settings above:

Service	From	To	Port	Protocol	wazi	wazi-all	Routeable	Function
debug-manager	User client	z/OS	7397	TLS	Y	Y		z/OS Debugger
debug-profile-service	User client	z/OS	7392	HTTPS	Y	Y	Y	z/OS Debugger
dtcn-cicsts56	User client	z/OS	7350	TCP		Y		z/OS Debugger (CICS®)
dtcn-cicsts55	User client	z/OS	7360	TCP		Y		z/OS Debugger (CICS)
ftp	User client	z/OS	21	FTP		Y		FTP to / from z/OS
local-vtam	3270 terminal emulator	z/OS	3270	TN3270E		Y		z/OS Local VTAM®
remote-debug-service	User client	z/OS	7394	TLS	Y	Y		z/OS Debugger
rse-api	User client	z/OS	7395	HTTPS	Y	Y	Y	IBM® Explorer for z/OS
rsed	User client	z/OS	4035	TLS	Y	Y		IBM Explorer for z/OS
rse-server	User client	z/OS	4037	TLS	Y	Y		IBM Explorer for z/OS
ssh	User client	z/OS	22	SSH	Y	Y		SSH to z/OS
tn3270-secure	3270 terminal emulator	z/OS	2023	TN3270E	Y	Y		TN3270 terminal sessions to z/OS (TLS)
tn3270-unsecured	3270 terminal emulator	z/OS	23	TN3270E		Y		TN3270 terminal sessions to z/OS (no TLS)
zosconnect	User client	z/OS	9443	TCP		Y		IBM z/OS Connect
zosmf	User client	z/OS	10443	HTTPS	Y	Y	Y	z/OS Management Facility (z/OSMF)

Sandbox instances might have additional network traffic to and from z/OS. Consult your z/OS administrator for details.

Limitations

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

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.

Extraction of uncataloged data sets

When using ZD&T z/OS Extraction Utilities, specifically feucvoli, to find data sets from source environments for extraction, uncataloged data sets are not automatically detected and included. In cases where uncataloged data sets are needed, such as JES2 Checkpoint or JES2 spool, the disk volumes containing these data sets must be included manually, unless there are other cataloged data sets on these volumes that have already been identified. If additional volumes that contain uncataloged data sets are needed, the system programmer of the z/OS source environments must make necessary adjustments to the list of volumes.

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.

On-going extracted volumes

When the creation of images is in progress, if the WebSphere® Liberty server that hosts the web server 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.

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.
- 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.

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 uncataloged 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 a CICS subsystem from a source Z.
- All entities of VSAM file will be transferred. For example, if KSDS has an alternative index, it will be transferred with the primary cluster to the target machine, even though the alternative index is not identified in the Application Discovery project.
- All members of the PDS identified by the Application Discovery project will be transferred. For example, if ibmuser.sysin (xyz) is referred in the JCL, the complete PDS (ibmuser.sysin) will be transferred to the target machine.
- Tapes are not supported.

Downloading installation packages

Learn about how to download installation packages.

Two types of packages are provided for installation: one for Extended ADCD, which can be downloaded optionally; one for the other components of ZD&T Enterprise Edition. To download the ZD&T Enterprise Edition V14.2.0 package or the Extended ADCD package, complete the following steps:

1. Log on to Passport Advantage®.
2. Select Software download and Media access.
3. Select Program offering and agreement number, and then click Continue.
4. Enter the part description or part number, and then 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 lists in the following tables.

After a package is downloaded, you can verify the integrity of the downloaded package by using the .md5 checksum file. To do a checksum on a downloaded package, use command `md5sum -c <checksum_file_name>`.

Table 1. ZD&T Enterprise Edition V14.2.0

	Name	Part No.	Image Classification	File Name
1	IBM Z Development and Test Environment Enterprise Edition 14.2 Installation Multilingual eAssembly	G098DML	Required	N/A
2	IBM Z Development and Test Environment Enterprise Edition 14.2 Installation Multilingual	M0FG0ML	Required	zdt-install.tgz
3	IBM Z Development and Test Environment Enterprise Edition 14.2 Installer checksum Multilingual	M0FG1ML	Required	ee.md5
4	IBM Z Development and Test Environment Enterprise Edition 14.x Quick Start Guide	M0839ML	Required	ZDT_Quick_Start_Guide_14.x.pdf
5	IBM Common Licensing V9.0 Setup Multiplatform Multilingual	CC996ML	Optional	N/A
6	IBM License Key Server V9.0 for Windows Multiplatform Multilingual	CC997ML	Optional	N/A
7	IBM License Key Server V9.0 for Linux x86_64 Multiplatform Multilingual	CC998ML	Optional	N/A
8	IBM License Key Server V9.0 for AIX ppc64 Multiplatform Multilingual	CC999ML	Optional	N/A
9	IBM License Key Administration and Reporting Tool V9.0 Multiplatform Multilingual	CC99AML	Optional	N/A
10	IBM License Key Administration Agent V9.0 Multiplatform Multilingual	CC99BML	Optional	N/A
11	IBM Db2 Standard Edition - VPC Option - Activation 11.5 for Linux	CC36WML	Optional	N/A
12	IBM Db2 Server 11.5 for Linux on AMD64 and Intel EM64T system	CC1U0ML	Optional	N/A
13	IBM Db2 Server 11.5 for Linux on System z Multilingual	CC1U1ML	Optional	N/A
14	IBM Db2 Server 11.5 for Linux on POWER systems (little endian)	CC1U2ML	Optional	N/A
15	IBM Db2 Server 11.5 for AIX Multilingual	CC1U3ML	Optional	N/A
16	IBM Db2 Server 11.5 for Windows on AMD64 and Intel EM64T system	CC1U4ML	Optional	N/A

Table 2. Extended ADCD z/OS V3.1 December Edition of 2023, uploaded in March 2024

	Name	Part No.	Image Classification	File Name
1	ADCD 1Q24 z/OS V3.1 for IBM Z Development and Test 14.2.2 RSU 2309 Multilingual eAssembly	G0BGBML	Optional	N/A
2	ADCD 1Q24 z/OS 3.1 Part 1 of 20 - RES volume 1 Multilingual	M0HDDML	Optional	A3RES1.ZPD
3	ADCD 1Q24 z/OS 3.1 Part 2 of 20 - RES volume 2 Multilingual	M0HDFML	Optional	A3RES2.gz
4	ADCD 1Q24 z/OS 3.1 Part 3 of 20 - System volume Multilingual	M0HDGML	Optional	A3SYS1.gz
5	ADCD 1Q24 z/OS 3.1 Part 4 of 20 - Configuration volume Multilingual	M0HDHML	Optional	A3CFG1.gz
6	ADCD 1Q24 z/OS 3.1 Part 5 of 20 - UNIX System Services volume 1 Multilingual	M0HDJML	Optional	A3USS1.gz
7	ADCD 1Q24 z/OS 3.1 Part 6 of 20 - UNIX System Services volume 2 Multilingual	M0HDKML	Optional	A3USS2.gz
8	ADCD 1Q24 z/OS 3.1 Part 7 of 20 - UNIX System Services volume 3 Multilingual	M0HDLML	Optional	A3USS3.gz
9	ADCD 1Q24 z/OS 3.1 Part 8 of 20 - z/OS Products volume 1 Multilingual	M0HDMML	Optional	A3PRD1.gz
10	ADCD 1Q24 z/OS 3.1 Part 9 of 20 - z/OS Products volume 2 Multilingual	M0HDNML	Optional	A3PRD2.gz
11	ADCD 1Q24 z/OS 3.1 Part 10 of 20 - z/OS Products volume 3 Multilingual	M0HDPML	Optional	A3PRD3.gz
12	ADCD 1Q24 z/OS 3.1 Part 11 of 20 - z/OS Products volume 4 Multilingual	M0HDQML	Optional	A3PRD4.gz
13	ADCD 1Q24 z/OS 3.1 Part 12 of 20 - z/OS Products volume 5 Multilingual	M0HDRML	Optional	A3PRD5.gz
14	ADCD 1Q24 z/OS 3.1 Part 13 of 20 - z/OS Distribution Libraries volume 1 Multilingual	M0HDSML	Optional	A3DIS1.gz
15	ADCD 1Q24 z/OS 3.1 Part 14 of 20 - z/OS Distribution Libraries volume 2 Multilingual	M0HDTML	Optional	A3DIS2.gz

	Name	Part No.	Image Classification	File Name
16	ADCD 1Q24 z/OS 3.1 Part 15 of 20 - z/OS Distribution Volume Multilingual	M0HDVML	Optional	A3DIS3.gz
17	ADCD 1Q24 z/OS 3.1 Part 16 of 20 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M0HDWML	Optional	A3PAGA.gz
18	ADCD 1Q24 z/OS 3.1 Part 17 of 20 - z/OS Page Datasets (Local) volume 2 Multilingual	M0HDXML	Optional	A3PAGB.gz
19	ADCD 1Q24 z/OS 3.1 Part 18 of 20 - z/OS Page Datasets (Local) volume 3 Multilingual	M0HDYML	Optional	A3PAGC.gz
20	ADCD 1Q24 z/OS 3.1 Part 19 of 20 - z/OS User volume Multilingual	M0HDZML	Optional	A3USR1.gz
21	ADCD 1Q24 z/OS 3.1 Part 20 of 20 - Stand Alone RES volume Multilingual	M0HF0ML	Optional	SARES1.ZPD
22	ADCD 1Q24 Engineering Workflow Management 7.0.2 Multilingual	M0HF1ML	Optional	A3BLZ1.gz
23	ADCD 1Q24 DB2 V12 Part 1 of 2 Multilingual	M0HF2ML	Optional	A3DBC1.gz
24	ADCD 1Q24 DB2 V12 Part 2 of 2 Multilingual	M0HF3ML	Optional	A3DBC2.gz
25	ADCD 1Q24 DB2 V13 Part 1 of 2 Multilingual	M0HF4ML	Optional	A3DBD1.gz
26	ADCD 1Q24 DB2 V13 Part 2 of 2 Multilingual	M0HF5ML	Optional	A3DBD2.gz
27	ADCD 1Q24 CICS 5.6 Multilingual	M0HF6ML	Optional	A3C560.gz
28	ADCD 1Q24 CICS 6.1 Multilingual	M0HF7ML	Optional	A3C610.gz
29	ADCD 1Q24 IMS 15.2 Multilingual	M0HF8ML	Optional	A3IMF1.gz
30	ADCD 1Q24 IBM Installation Manager 1.9 Multilingual	M0HF9ML	Optional	A3INM1.gz
31	ADCD 1Q24 DB2 Archive Logs Multilingual	M0HFBML	Optional	A3DBAR.gz
32	ADCD 1Q24 IBM Z Monitoring Suite v1.2	M0HFCML	Optional	A3KAN1.gz
33	ADCD 1Q24 WAS V9.0 Part 1 of 2 Multilingual	M0HFDML	Optional	A3W901.gz
34	ADCD 1Q24 WAS V9.0 Part 2 of 2 Multilingual	M0HFFML	Optional	A3W902.gz
35	ADCD 1Q24 Zowe Open Source Project Multilingual	M0HFGML	Optional	A3ZWE1.gz
36	ADCD Metadata for Tools Multilingual	M0HFHML	Optional	ADCDTOOLS.XML
37	ADCD 1Q24 Inventory for Sandbox Multilingual	M0HFJML	Optional	inventory.txt
38	ADCD 1Q24 Devmap for Sandbox Multilingual	M0HFKML	Optional	devmap.txt
39	ADCD 1Q24 Extended ADCD Notices Multilingual	M0HFLML	Optional	notices_ADCD_RSU 2309
40	ADCD 1Q24 Inventory with SHA256 Checksums for Sandbox Multilingual	M0HFMML	Optional	inventory_sha256.txt
41	ADCD 1Q24 IPL Configuration Script Multilingual	M0HFNML	Optional	zdt_config_adcd_ipl.sh
42	ADCD z/OS V2.5 for IBM Z Development and Test 14.2 RSU 2303 Multilingual eAssembly	G08N2ML	Optional	N/A
43	ADCD 3Q'23: z/OS 2.5 Part 1 of 20 - RES volume 1 Multilingual	M0FDQML	Optional	D5RES1.ZPD
44	ADCD 3Q'23: z/OS 2.5 Part 2 of 20 - RES volume 2 Multilingual	M0FDRML	Optional	D5RES2.gz
45	ADCD 3Q'23: z/OS 2.5 Part 3 of 20 - System volume Multilingual	M0FDSML	Optional	D5SYS1.gz
46	ADCD 3Q'23: z/OS 2.5 Part 4 of 20 - Configuration volume Multilingual	M0FDTML	Optional	D5CFG1.gz
47	ADCD 3Q'23: z/OS 2.5 Part 5 of 20 - UNIX System Services volume 1 Multilingual	M0FDVML	Optional	D5USS1.gz
48	ADCD 3Q'23: z/OS 2.5 Part 6 of 20 - UNIX System Services volume 2 Multilingual	M0FDWML	Optional	D5USS2.gz
49	ADCD 3Q'23: z/OS 2.5 Part 7 of 20 - UNIX System Services volume 3 Multilingual	M0FDXML	Optional	D5USS3.gz
50	ADCD 3Q'23: z/OS 2.5 Part 8 of 20 - z/OS Products volume 1 Multilingual	M0FDYML	Optional	D5PRD1.gz
51	ADCD 3Q'23: z/OS 2.5 Part 9 of 20 - z/OS Products volume 2 Multilingual	M0FDZML	Optional	D5PRD2.gz
52	ADCD 3Q'23: z/OS 2.5 Part 10 of 20 - z/OS Products volume 3 Multilingual	M0FF0ML	Optional	D5PRD3.gz
53	ADCD 3Q'23: z/OS 2.5 Part 11 of 20 - z/OS Products volume 4 Multilingual	M0FF1ML	Optional	D5PRD4.gz
54	ADCD 3Q'23: z/OS 2.5 Part 12 of 20 - z/OS Products volume 5 Multilingual	M0FF2ML	Optional	D5PRD5.gz
55	ADCD 3Q'23: z/OS 2.5 Part 13 of 20 - z/OS Distribution Libraries volume 1 Multilingual	M0FF3ML	Optional	D5DIS1.gz
56	ADCD 3Q'23: z/OS 2.5 Part 14 of 20 - z/OS Distribution Libraries volume 2 Multilingual	M0FF4ML	Optional	D5DIS2.gz
57	ADCD 3Q'23: z/OS 2.5 Part 15 of 20 - z/OS Distribution Volume Multilingual	M0FF5ML	Optional	D5DIS3.gz
58	ADCD 3Q'23: z/OS 2.5 Part 16 of 20 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M0FF6ML	Optional	D5PAGA.gz
59	ADCD 3Q'23: z/OS 2.5 Part 17 of 20 - z/OS Page Datasets (Local) volume 2 Multilingual	M0FF7ML	Optional	D5PAGB.gz
60	ADCD 3Q'23: z/OS 2.5 Part 18 of 20 - z/OS Page Datasets (Local) volume 3 Multilingual	M0FF8ML	Optional	D5PAGC.gz
61	ADCD 3Q'23: z/OS 2.5 Part 19 of 20 - z/OS User volume Multilingual	M0FF9ML	Optional	D5USR1.gz
62	ADCD 3Q'23: z/OS 2.5 Part 20 of 20 - Stand Alone RES volume Multilingual	M0FFBML	Optional	SARES1.ZPD
63	ADCD 3Q'23: Engineering Workflow Management 7.0.2 Multilingual	M0FFCML	Optional	D5BLZ1.gz
64	ADCD 3Q'23: DB2 V12 Part 1 of 2 Multilingual	M0FFDML	Optional	D5DBC1.gz
65	ADCD 3Q'23: DB2 V12 Part 2 of 2 Multilingual	M0FFFML	Optional	D5DBC2.gz
66	ADCD 3Q'23: DB2 V13 Part 1 of 2 Multilingual	M0FFGML	Optional	D5DBD1.gz
67	ADCD 3Q'23: DB2 V13 Part 2 of 2 Multilingual	M0FFHML	Optional	D5DBD2.gz
68	ADCD 3Q'23: CICS 5.6 Multilingual	M0FFJML	Optional	D5C560.gz
69	ADCD 3Q'23: CICS 6.1 Multilingual	M0FFKML	Optional	D5C610.gz
70	ADCD 3Q'23: IMS 15.2 Multilingual	M0FFLML	Optional	D5IMF1.gz
71	ADCD 3Q'23: IBM Installation Manager 1.9 Multilingual	M0FFMML	Optional	D5INM1.gz
72	ADCD 3Q'23: DB2 Archive Logs Multilingual	M0FFNML	Optional	D5DBAR.gz
73	ADCD 3Q'23: IBM Z Monitoring Suite v1.2	M0FFPML	Optional	D5KAN1.gz
74	ADCD 3Q'23: WAS V9.0 Part 1 of 2 Multilingual	M0FFQML	Optional	D5W901.gz
75	ADCD 3Q'23: WAS V9.0 Part 2 of 2 Multilingual	M0FFRML	Optional	D5W902.gz
76	ADCD 3Q'23: Zowe Open Source Project Multilingual	M0FFSML	Optional	D5ZWE1.gz
77	ADCD Metadata for Tools Multilingual	M0FFTML	Optional	ADCDTOOLS.XML

	Name	Part No.	Image Classification	File Name
78	ADCD 3Q'23 Inventory for Sandbox Multilingual	M0FFVML	Optional	inventory.txt
79	ADCD 3Q'23 Devmap for Sandbox 1.4 Linux Multilingual	M0G01ML	Optional	devmap.txt
80	ADCD 3Q'23 Extended ADCD Notices Multilingual	M0FFXML	Optional	notices_ADCD_RSU 2303
81	ADCD 3Q'23 Inventory with SHA256 Checksums for Sandbox Multilingual	M0G7CML	Optional	inventory_sha256.txt
82	ADCD 3Q'23 IPL Configuration Script Multilingual	M0FFZML	Optional	zdt_config_adcd_ipl .sh

- **Validating the signed installers**

Validating the signed installers

The .tgz files that you download from IBM are compressed installation packages. Each ZD&T Enterprise Edition installation package contains the following signed installers and the corresponding signature files.

Installer	Signature file
Web server installer zdt-install	zdt-install.cosign.sig
License server installer zdt-license-server	zdt-license-server.cosign.sig
Emulator installer zdt-emulator	zdt-emulator.cosign.sig

You can validate any signed installer with its signature file by running the following command.

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

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

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

If the installer is valid, you can receive the message of `Verified OK`.

Installing and configuring

Learn how to install and configure ZD&T components.

- **Setting up the web server**
  ZD&T Enterprise Edition provides a web server that enables 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.
- **Optional (Legacy): Installing the emulator**
  To install ZD&T emulator, run the installer with the root user ID.
- **Setting up 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.
- **Optional: Installing ZD&T z/OS Extraction Utilities**
  To create components from z/OS, either genuine z/OS on IBM Z or an existing z/OS instance that runs on Linux®, you need to install ZD&T z/OS Extraction Utilities.
- **Configuring on the web server**
  Learn how to configure on the web server.

Setting up the web server

ZD&T Enterprise Edition provides a web server that enables 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.

- **Upgrading the web server**
  To upgrade the web server, you must run the installer with the root user ID.
- **Installing the web server**
  To install the ZD&T Enterprise Edition web server, run the installer with the root user ID.
- **Installing a signed certificate on the 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 web server.
- **Configuring the web server**
  Before you start your Enterprise Edition web server, ensure that you configure the authentication for Enterprise Edition.
- **Starting and stopping the 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 web server and ensure that the server process runs under the user ID that is specified during the installation.

Upgrading the web server

To upgrade the web server, you must run the installer with the root user ID.

To upgrade from V12.x or earlier, upgrade to V13 first. Make sure that it runs and no manual migration is needed for any component or image, and then upgrade from V13.

1. Open the directory where the installation package zdt-install.tgz is stored.

   `cd <directory>`

2. Change the authority of the installation package.

   `chmod 755 <package_name>`

3. Decompress the installation package.

   `tar -xvf <package_name>`

   When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

   Web server installer
   > zdt-install

4. Run the installer, and then follow instructions to upgrade.

   `./<installer_name>`

   You can upgrade the web server with the following options.
   - Select **1** to reinstall the web server with existing settings except the password of the web server.
   - Select **2** to refresh the installation of the web server by removing all existing settings and configuring new settings.

   More information about the settings can be found in topic [Installing the web server](#).
   Alternatively, you can upgrade the web server silently by using the following command.

   `./<installer_name> --update --zdtee`

5. 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 web server is installed successfully, information, including the right version number, of the installed application is returned.

What's next

If you set the web server to start automatically, you can find the web server URL in the output. If you selected to start the web server manually, you can follow the instructions in topic [Starting and stopping the web server](#).

Installing the web server

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

1. Open the directory where the installation package zdt-install.tgz is stored.

   `cd <directory>`

2. Decompress the installation package.

   `tar -xvf <package_name>`

   When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

   Web server installer
   > zdt-install

3. Run the installer, and then follow instructions to complete the installation.

   `./<installer_name>`

   Standard installation
   > This option will install ZD&T Enterprise Edition with the following default settings:

   > Default user ID for running the web server
   >> zdt
   > Default web server installation directory
   >> /opt/ibm/zdt
   > Default port for accessing the web server
   >> 9443

Default user ID for logging into the web server
    zdtadmin
Default passphrase for the SSH private key to log into the Linux target
    zdttool$4UFromVM

Also, if you select standard installation, the software-based license server will be automatically installed on the same machine.
Custom installation
    You can follow instructions to modify all default settings and complete the installation. You can also choose to automatically install the license server on the same machine, or install it separately by following the instructions in topic Installing the software-based license server.

Note: Terraform is required only when you deploy custom images to your target environment in IBM Cloud®. By running the web server installer, you can choose to automatically install the Terraform binary from HashiCorp. You can also install Terraform manually.
Alternatively, you can run the following command to install the web server silently. 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 Hardware and software requirements.

```
./<installer_name> --install --zdtee --zdtee_port=<port_number> --zdtee_user=<username> --zdtee_path=<installation_path> --zdtee_password=<password>
```

4. 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 web server is installed successfully, information, including the right version number, of the installed application is returned.

Installing a signed certificate on the 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 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 web server. 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>/zdt-server/resources/security.

   ```
   openssl pkcs12 -export -out zdtkey.p12 -inkey cert.key -in cert.crt -password pass:<password_value>
   ```

3. Modify the encrypted key store password.
   a. Get the encryption key that is specified by wlp.password.encryption.key in the <installation_directory>/zdt-server/bootstrap.properties; for example, /opt/ibm/zdt/zdt-server/bootstrap.properties
   b. Run the following command where you installed the web server.

      ```
      <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/zdt-server/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
   ```

Configuring the 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 use the ZD&T Enterprise Edition web server, you can modify how authentication is accomplished by the Enterprise Edition.
- **Auditing the web server**
  Learn about the steps to enable audit log for ZD&T Enterprise Edition.

Authentication for the Enterprise Edition application

Before you use the ZD&T Enterprise Edition web 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.

- **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.
- **Lightweight Directory Access Protocol (LDAP)**
  Learn about the steps to enable LDAP authentication for ZD&T 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' is defined in file *<installation_directory>*/Liberty/usr/servers/zdt-server/basicauth-conf.xml, and the default password of this user ID is set during installation. This password is authorized as an administrator account on the web server. To change the default password, complete the following steps.

1. Find the encryption key that is specified by `wlp.password.encryption.key` in the file *<installation_directory>*/Liberty/usr/servers/zdt-server/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 the web server.

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

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

```
<!-- user authentication -->
<basicRegistry
id="_home_markdown_jenkins_workspace_Transform_in_SSTQBD_14.2.0_com.ibm.zsys.rdt.tools.user.guide.doc_topics_default_ps_zd
t" realm="zdt">
<user name="zdtadmin"
password="*{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB*"/>
</basicRegistry>
```

For more information about authorization on the web server, see Managing users and roles.

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.

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 file *<installation_directory>*/Liberty/usr/servers/zdt-server/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 the web server.

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

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

```
<basicRegistry
id="_home_markdown_jenkins_workspace_Transform_in_SSTQBD_14.2.0_com.ibm.zsys.rdt.tools.user.guide.doc_topics_new_user_zdt"
realm="zdt">
        <user name="zdtadmin" password="{aes}AM1LZsnwLRNsVtYAiwqhVDO9/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 web server. For more information about authorization, see Managing users and roles.

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 *<installation_directory>*/Liberty/usr/servers/zdt-server/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 *<installation_directory>*/Liberty/usr/servers/zdt-server/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-enanbled 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 web server. For more information about authorization, see Managing users and roles.

Auditing the web server

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

The audit log for the web server is defined in the *<installation_directory>*/Liberty/usr/servers/zdt-server/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](#).

Starting and stopping the 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 web server and ensure that the server process runs under the user ID that is specified during the installation.

Changing the user ID

You can change the user ID by modifying *ZDTEE_USER* in the *<installation_directory>*/bin/zdtee_env.sh script file. You also need to change the file ownership of all the files in the directory *<installation_directory>*/Liberty/usr/servers/zdt-server/ to the user ID, for example, `chmod -R newuser:root` `<installation_directory>/Liberty/usr/servers/zdt-server/.`

If the web server is already started, you can run the following command before you restart the server by using the newly specified user ID.

`<installation_directory>/bin/stopServer.sh`

Starting the web server manually

To start the 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 *<installation_directory>*/bin/zdtee_env.sh script file.

To start the web server, run the following command.

`<installation_directory>/bin/startServer.sh`

Note: If you selected to set the web server to start and stop automatically, you can go to the URL in the installation output directly.
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.

Stopping the web server manually

If you need to stop the web server, run the following command.

`<installation_directory>/bin/stopServer.sh`

Optional (Legacy): Installing the 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 [Hardware and software requirements](#). 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.

Setting up 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 IBM 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 IBM 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 LKC, 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 LKC, 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.

Table 1. Enablement checklist for software-based licensing

	Software-based license	Required/Optional	Complete
1	Installing the software-based license server Describe the steps to install software-based license server	Required	☐
2	Obtaining an update file Describes the steps to obtain an update file for software-based license server.	Required	☐
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	☐

- **Upgrading the software-based license server**
  To upgrade 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.
- **Installing the software-based license server**
  To install the software-based license server, run the installer with the root user ID.
- **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.
- **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.
- **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.

Upgrading the software-based license server

To upgrade 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 upgrade 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.

To upgrade the software-based license server, complete the following steps:

1. Open the directory where the installation package zdt-install.tgz is stored.

```
cd <directory>
```

2. Change the authority of the installation package.

```
chmod 755 <package_name>
```

3. Decompress the installation package.

```
tar -xvf <package_name>
```

When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

License server installer
        zdt-license-server

4. Run the installer, and then follow instructions to upgrade.

```
./<installer_name>
```

Alternatively, you can upgrade the software-based license server silently with the following command.

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

5. After the upgrade completes, run the following commands to verify whether the upgrade is successful.

RHEL on x86

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

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

Ubuntu Linux® on x86

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

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

If the license server is upgraded successfully, information, including the right version number, of the installed application is returned.

What's next

After the software-based license server is upgraded, you need to enable the license server by running the steps that are described in section Setting up the software-based license server.

Installing the software-based license server

To install the software-based license server, run the installer with the root user ID.

1. Open the directory where the installation package zdt-install.tgz is stored.

```
cd <directory>
```

2. Change the authority of the installation package.

```
chmod 755 <package_name>
```

3. Decompress the installation package.

```
tar -xvf <package_name>
```

When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

License server installer
        zdt-license-server

4. Run the installer, and then follow instructions to complete the installation.

```
./<installer_name>
```

During the installation, you can choose to automatically install all necessary dependencies mentioned in topic Hardware and software requirements. Your must have access to the internet and software repository to install the dependencies. You can also choose to install the dependencies manually.
Alternatively, you can install the software-based license server silently with the following commands.

- To install the software-based license server, run the following command.

```
./<installer_name> --install
```

- To install the dependencies, run the following command.

```
<installer_name> --install --dep
```

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

RHEL on x86

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

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

Ubuntu Linux on x86

```
dpkg -l | grep zpdtldk

dpkg -l | grep zpdtuim
```

If the license server is installed successfully, information, including the right version number, of the installed application is returned.
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
```

6. Configure the iptables port for the license server by opening both UDP and TCP port 1947 that are listed in the license server port table. For more information about how to set up port 1947, see Firewalls.

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.
- **For the software-based license server**
  Learn about the steps to obtain an update file for 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 (LKC), and select your account. If you do not have access to LKC, 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 IBM 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 the software-based license server

Learn about the steps to obtain an update file for the 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 (LKC), and select your account. If you do not have access to LKC, 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 IBM 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 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 the 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


Thesse are the currently active sessions:

      KeyID          FID     FeatureName      Address      User     Machine       LoginTime         #
   432975633343422885 333    CPU License   9.26.158.161   ibmsys1   zdt-dev4   Mon Oct 30, 17:25:40   3
```

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.

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

IBM Z Development and Test Environment 14.2.x **63**

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_...98020495.zip	100

Generate

3. Download the generated update file.

View Licenses By Host

Host Name: sg1
Host ID: 951228536548436554

Generated By : Sanjay Ghanathey (sanjay@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

Optional: Installing ZD&T z/OS Extraction Utilities

To create components from z/OS®, either genuine z/OS on IBM Z® or an existing z/OS instance that runs on Linux®, you need to install ZD&T z/OS Extraction Utilities.

ZD&T z/OS Extraction Utilities is a component to be installed on one or more z/OS source environments to extract z/OS, CICS®, and Db2® resources.

To extract resources from an existing z/OS instance that runs on Linux, install ZD&T z/OS Extraction Utilities by transferring *<installation_directory>*/zSystem/zdtMainframeSFTP.jar to /usr/lpp/IBM/zdt of the Linux system.

To extract resources from genuine z/OS on IBM Z, install ZD&T z/OS Extraction Utilities by following the steps:
Note: If you have previously installed ZD&T z/OS Extraction Utilities and want to use the automated system volume discovery tool when you [create a volume component](#), make sure that you complete steps 4 to 6.

1. After the installer zdt-install.tgz runs on a Linux machine to install the web user interface, directory *<installation_directory>*/zSystem contains the IBM® Z components, the program directory file [HALME21.pdf](#), and other files, including jar file zdtMainframeSFTP.jar.
2. Install the IBM Z components by following the instructions in HALME21.pdf, specifically in section 6.0 "Installation Instructions".
   The data set hlq.IBM.HALME21.F2 obtained at step 6.1.3 in HALME21.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.HALME21.F2 to dsprefix.SFEUAUTH, and rename hlq.IBM.HALME21.F3 to dsprefix.SFEUEXEC.
   b. Go to step 6.1.11 by skipping steps from 6.1.4 to 6.1.10.
3. Transfer *<installation_directory>*/zSystem/zdtMainframeSFTP.jar to the z/OS UNIX System Services folder /usr/lpp/IBM/zdt that is located on the source z/OS.
4. Transfer *<installation_directory>*/zSystem/feucvoli from the Tools Server to the z/OS UNIX System Services folder /usr/lpp/IBM/zdt that is located on the source z/OS by using Secure Copy Protocol (SCP) instead of Secure File Transfer Protocol (SFTP).
5. Ensure that /usr/lpp/IBM/zdt/feucvoli is executable by running the **chmod 755 /usr/lpp/IBM/zdt/feucvoli** command in the UNIX System Services on z/OS.
6. To extract volumes, add export zdtAuth=STEPLIB_specified_in_feuc to your z/OS UNIX System Services login profile to point to the same STEPLIB as you specified in /usr/lpp/IBM/zdt/feuc.

Note:

- Instead of using the default directory /usr/lpp/IBM/zdt, you can use a different path and specify the value as the ZD&T z/OS Extraction tools installation directory with the web server. For instructions or more information about source environment types, see section [Adding source environments](#).
- Ensure that you have write access to the /tmp directory in the source z/OS UNIX System Services, because the utility will write a lot of logs into it. According to best practices, clean this folder periodically.

Configuring on the web server

Learn how to configure on the web server.

- **Connecting to your storage server**
  After you enabled an SFTP protocol, you can go to the Enterprise Edition web server UI to connect to the storage server.

- **Adding a software-based license server**
  The license servers that are used by each target ZD&T instance needs to be configured on the web server before the target instances are created. You must have administrator privileges to complete this task.
- **Managing users and roles**
  You can add and manage users the web server to control access to ZD&T Enterprise Edition.
- **Optional: Configuring IBM AD integrations**
  To make ZD&T Enterprise Edition use the knowledge base to locate the data sets of a specific application, configure IBM AD integrations by providing the details of your IBM® AD server. The integration configurations require administrator privileges.
- **Adding source environments**
  Source environments are configured to create components or images that are used to provision instances to target environments for development and testing.
- **Adding target environments**
  The target environment page is available for all users with different roles. You can provide the list of Docker, Linux®, or Red Hat® OpenShift® target environments that instances will be provisioned to.

Connecting to your storage server

After you enabled an SFTP protocol, you can go to the Enterprise Edition web server UI to connect to the storage server.

To connect to an SFTP server, complete the following steps:

1. Click Storage on the home page, or click the upper-right settings button to navigate to Storage.
2. Enter the qualified hostname of the storage server along with the port and directory.
3. Enter the credentials of the storage server.
4. Click Save.

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

Adding a software-based license server

The license servers that are used by each target ZD&T instance needs to be configured on the web server before the target instances are created. You must have administrator privileges to complete this task.

To add a software-based license server, complete the following steps:

1. Click Licensing on the home page, or click the upper-right settings button to navigate to Licensing.
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. Optional: Select if you want to set the license server as the primary license server.
4. Click Save.

After you add a license server, you can also edit or delete the license server at any time.

Managing users and roles

You can add and manage users the web server to control access to ZD&T Enterprise Edition.

Note: 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.

Configuring a user registry

To add a user to the web server, 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.

Adding a user

To add a new user who exists in the user registry to the web server, 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.
   - For the user to create z/OS® images or provision a z/OS images to a target environment, select the Developer/Tester role.
   - For the user to extract z/OS volumes for use on the web server, select the Builder role. Builder role includes the capabilities of the Developer/Tester role.
   - For the user to set up tooling on the web server, select the Administrator role. The Administrator role includes the capabilities of the Builder role.
4. Click Save.

Optional: Configuring IBM AD integrations

To make ZD&T Enterprise Edition use the knowledge base to locate the data sets of a specific application, configure IBM AD integrations by providing the details of your IBM® AD server. The integration configurations require administrator privileges.

To create images from an IBM AD project, you need to set up the connection to the IBM AD server:

HTTP/HTTPS
By default, the IBM AD web service is available via the HTTPS connection. If your IBM AD 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 AD.

Port
The port number of the IBM AD 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 AD 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 IBM AD server. Then, enter the hostname and port of the IBM AD server.
4. Enter the user ID and password to connect to the IBM AD server.
5. Click Save when the connection is successful.

When you configure the server, if errors occur, see Troubleshooting for configuring the IBM AD server for reference.

Related information

- ↪ IBM AD documentation

Adding 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 following source environment types are supported:

Genuine z/OS on IBM Z physical hardware
z/OS on IBM Z physical hardware with configured CICS® regions and Db2® subsystems. This environment type supports component creation by extracting, compressing, and transferring artifacts from an existing z/OS® environment on IBM Z®.
Existing z/OS instance run by ZD&T
Supports faster volume component creation by compressing and transferring an existing z/OS instance that ZD&T runs on a host Linux® system.
Extended ADCD
A pre-built Application Developers Controlled Distribution (ADCD) package that contains z/OS and many z/OS products and subsystems, from which you can create images.

Notes:

- For z/OS on IBM Z physical hardware source environments that are added, you can add, edit, or delete existing source systems, CICS regions, and Db2 subsystems. However, a source environment cannot be deleted if components created from this source environment exist.
- If you want to set up source environments from either genuine z/OS on IBM Z physical hardware or existing z/OS instances to extract and provision the necessary volumes or data sets, the z/OS system requirements must be met.
- Extracting from a z/OS instance initially provisioned from Extended ADCD is also considered as extracting from an existing z/OS instance; therefore, the z/OS system requirements apply.

- **Adding a source environment from a z/OS system on IBM Z physical hardware**
  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.
- **Adding a source environment from an existing z/OS instance run by ZD&T**
  You can create volume components quickly from an existing z/OS instance that runs on a host Linux system by adding this system as a source environment on the web server.
- **Adding a source environment from Extended ADCD**
  Learn how to add an Extended ADCD source environment on the web server. Administrator privileges are required to complete this task.

Adding a source environment from a z/OS system on IBM Z physical hardware

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 platform 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. Specify the ZD&T z/OS Extraction tools installation directory. The default path is /usr/lpp/IBM/zdt/.
7. 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.
8. 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.
9. 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.

Adding a source environment from an existing z/OS instance run by ZD&T

You can create volume components quickly from an existing z/OS instance that runs on a host Linux® system by adding this system as a source environment on the web server.

To add a Linux 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 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 z/OS® instance on Linux.
8. Specify the ZD&T z/OS Extraction tools installation directory. The default path is /usr/lpp/IBM/zdt/.
9. Click Connect ZD&T environment.
10. Enter the credentials to the Linux environment that you are connecting to, and click Submit.

Adding a source environment from Extended ADCD

Learn how to add an Extended ADCD source environment on the web server. Administrator privileges are required to complete this task.

Before you begin

Extended ADCD is a pre-built package that contains z/OS® and many z/OS products and subsystems from which you can create images. To create images from ADCD, you must configure Extended ADCD first. For more information, see Configuring Extended ADCD.

About this task

After the Extended ADCD files are successfully transferred to your storage server, you can add an ADCD source environment on the web server. To create an ADCD source environment, complete the following steps:

Procedure

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 instructions that are described in topic Configuring Extended ADCD are completed to discover Extended ADCD packages. Then, click Load ADCD. If duplicated Extended ADCD entries are found in the adcd directory location, you must ensure that only one copy of Extended ADCD exists in the location.
5. After the Extended ADCD packages are loaded successfully from the storage server, click Back to manage page.

Adding target environments

The target environment page is available for all users with different roles. You can provide the list of Docker, Linux®, 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.
- **Configuring a Linux target environment**
  Learn about the steps to configure, or delete a Linux target 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.

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 environment requirements.

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 Automated or Manual. If you select Automated, ZD&T Enterprise Edition will automatically configure the advanced settings during the provisioning process. If you select Manual, 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:
     - Modify the number of volumes that you want to extract and transfer simultaneously.
     - 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.

     - 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.

     - 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.

     - Specify the emulated z/OS IP address.
     - 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.

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

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.

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.
- **Enabling z/OS software to run on target environments**
  This section describes the steps to create components and images, and provision a z/OS instance from a created image by using the web server.
- **Using REST APIs**
  ZD&T Enterprise Edition supports REST APIs usage. You can access all functions of ZD&T Enterprise Edition without the web user interface, including creating components and images, provisioning instances from created images, and monitoring images and instances.

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 storage 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.
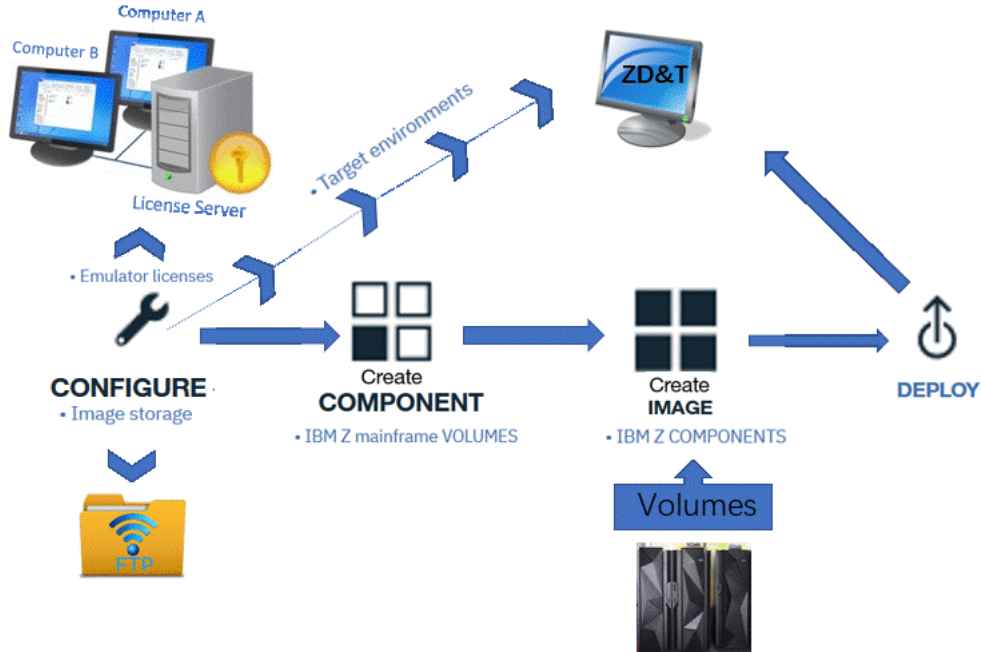
To transfer data sets either from z/OS or existing ZD&T machine, optionally, you need to validate the configuration of the storage server, License Server, and target machine that you must have configured when you provision the base target machine. Then, you can create components by selecting the data 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.
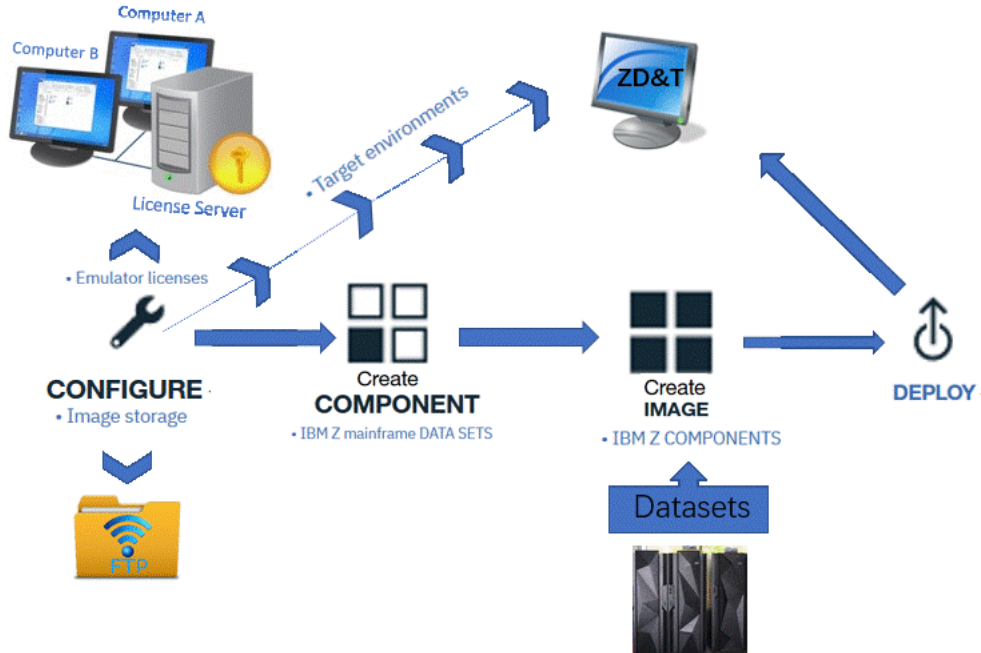
ZD&T Enterprise Edition can be integrated with IBM Application Discovery for IBM Z (IBM AD). If you have built your project in IBM AD, and want to transfer artifacts such as data sets from the source Z machine to the target environment, you need to configure the connection to your IBM AD web services. Then, create images from the created components, and provision a z/OS instance to the target environment from the created images.
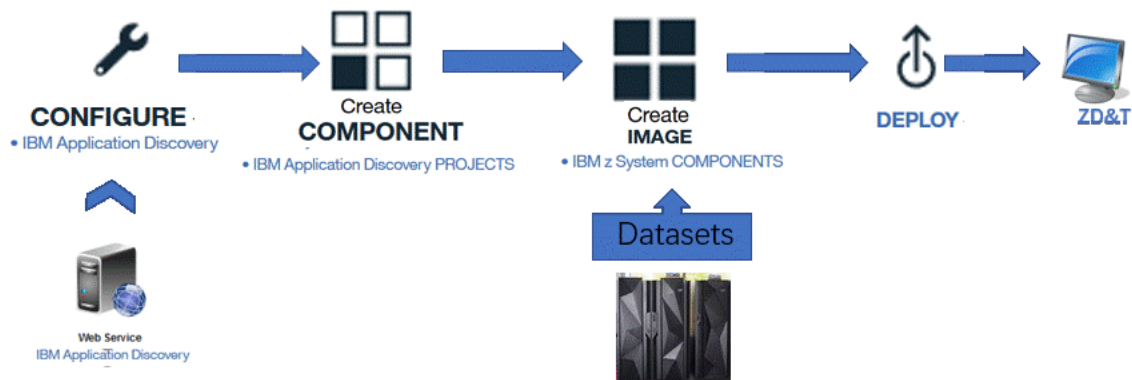
Images creation from ADCD

Before you provision ZD&T from ADCD, you need to configure the storage 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 the storage 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.
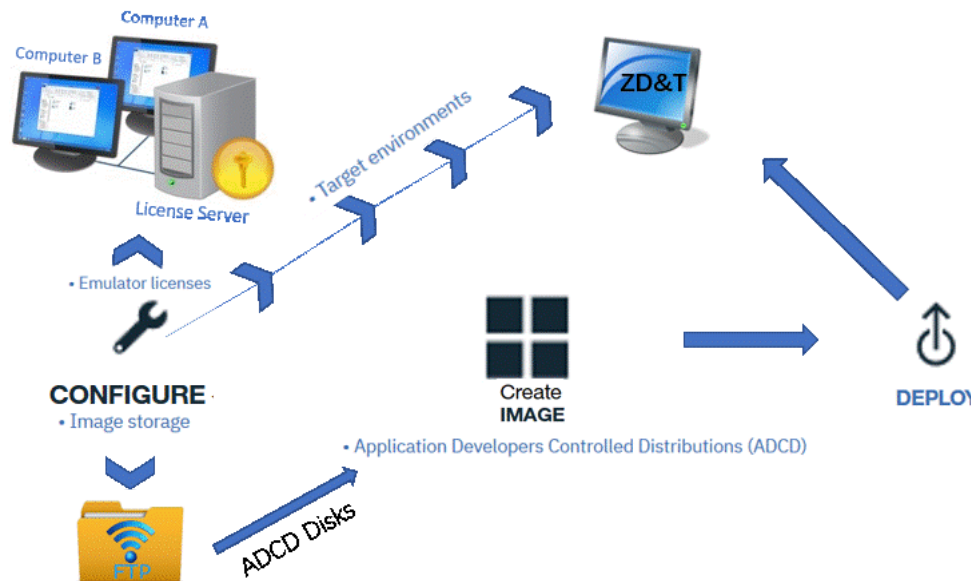
Cloud-ready images

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.

Enabling z/OS software to run on target environments

This section describes the steps to create components and images, and provision a z/OS® instance from a created image by using the web server.

- **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.
- **Creating components**
  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.
- **Monitoring and managing the created components**
  After you created a component, you can check the status of the extraction, and the details of your created components.
- **Creating images**
  An image is a collection of components that are packaged together from your source environment for deploying to your target environment. When creating an image, you must select to include one component of type system residence volume and any number of other components that 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 runs on emulated IBM Z hardware from the image..

- **Monitoring and managing the 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.
- **Provisioning instances to target environments**
  Learn how to provision z/OS instances to target environments.
- **Monitoring and managing the provisioned instances**
  After you provision a z/OS instance 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.

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.

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.
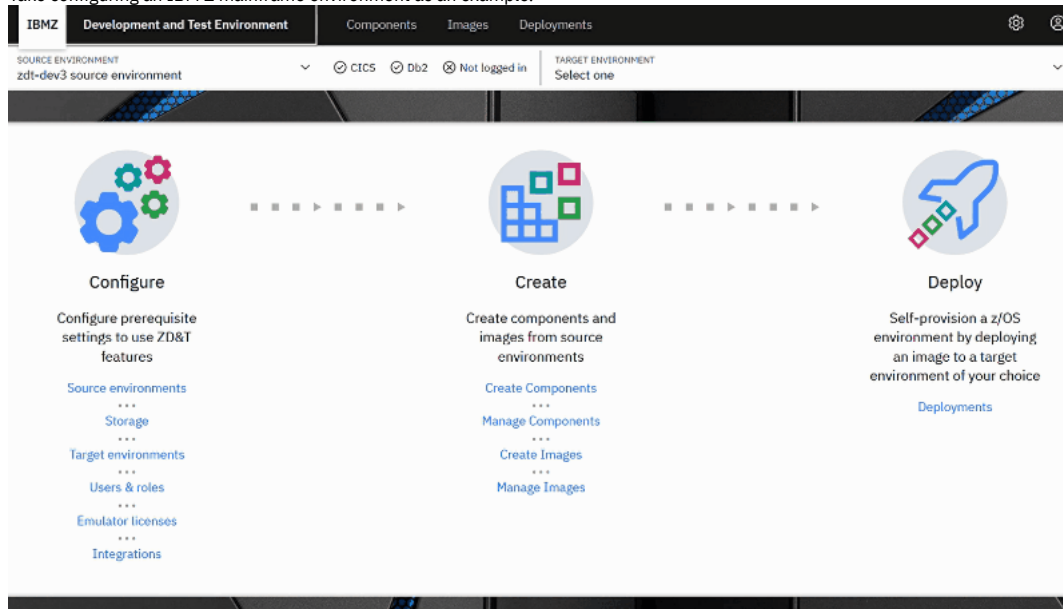
Target environment configuration

The target environment configuration page provides the following two options: Linux and Docker.

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.
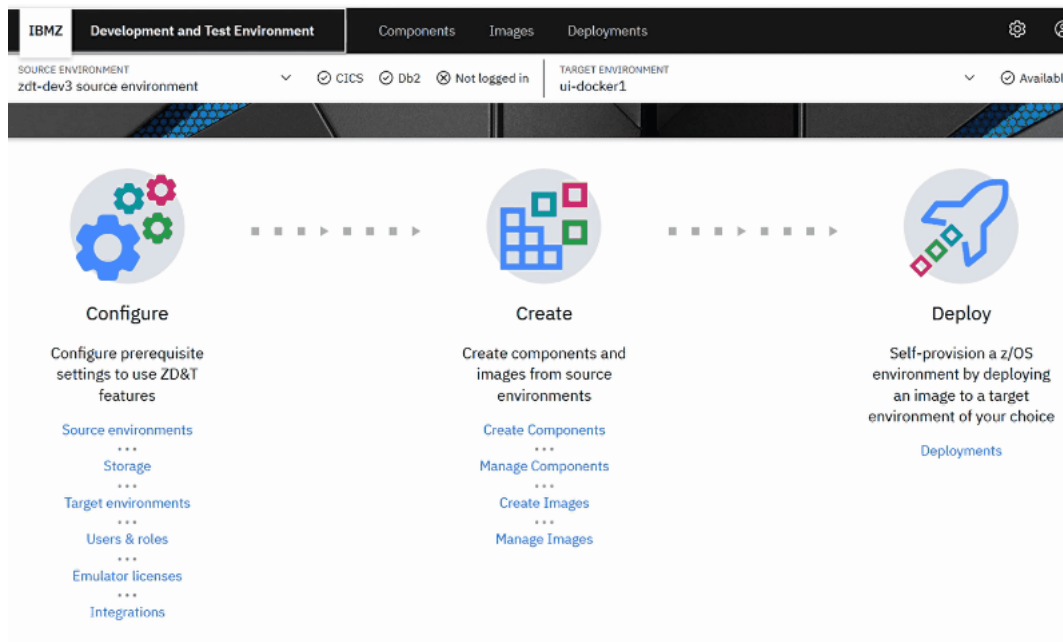
Component creation

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 component as an example.
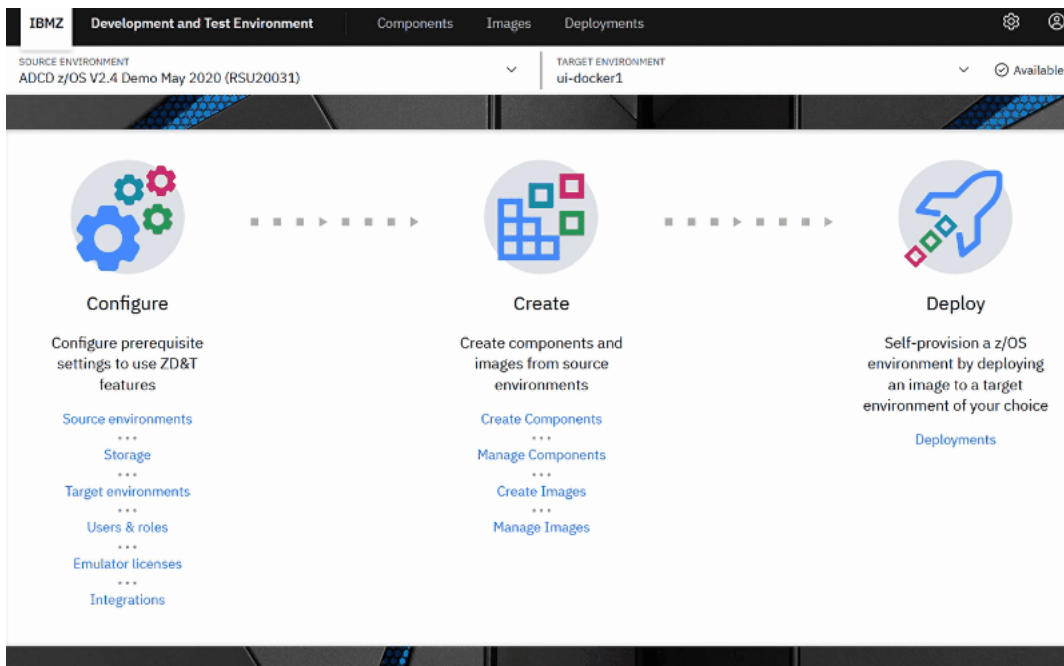
Image creation

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.
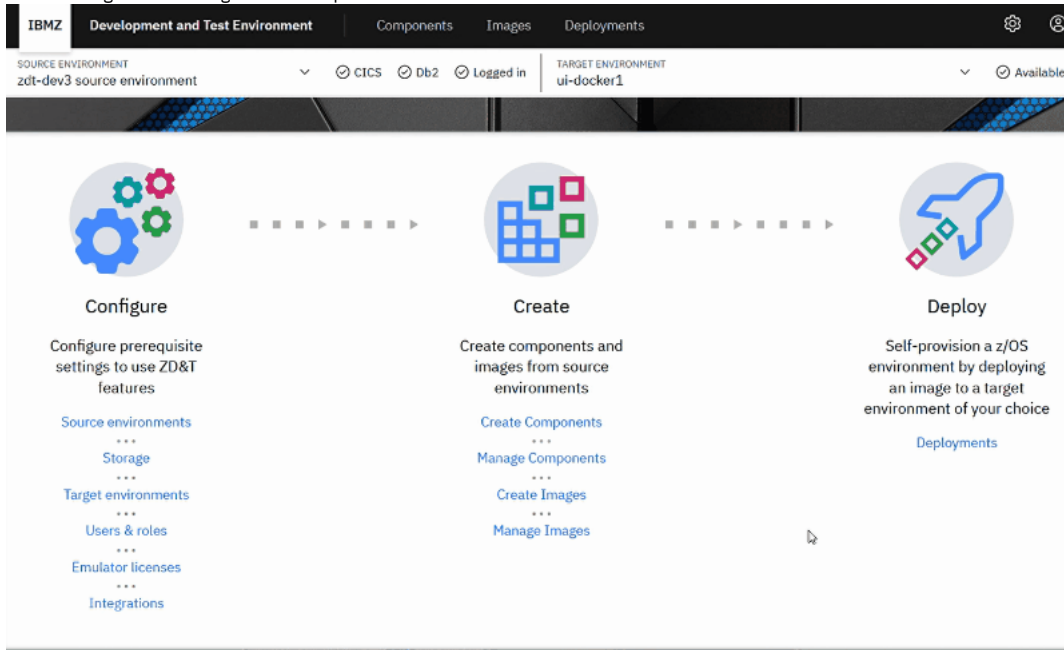
Assets monitoring and management

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.
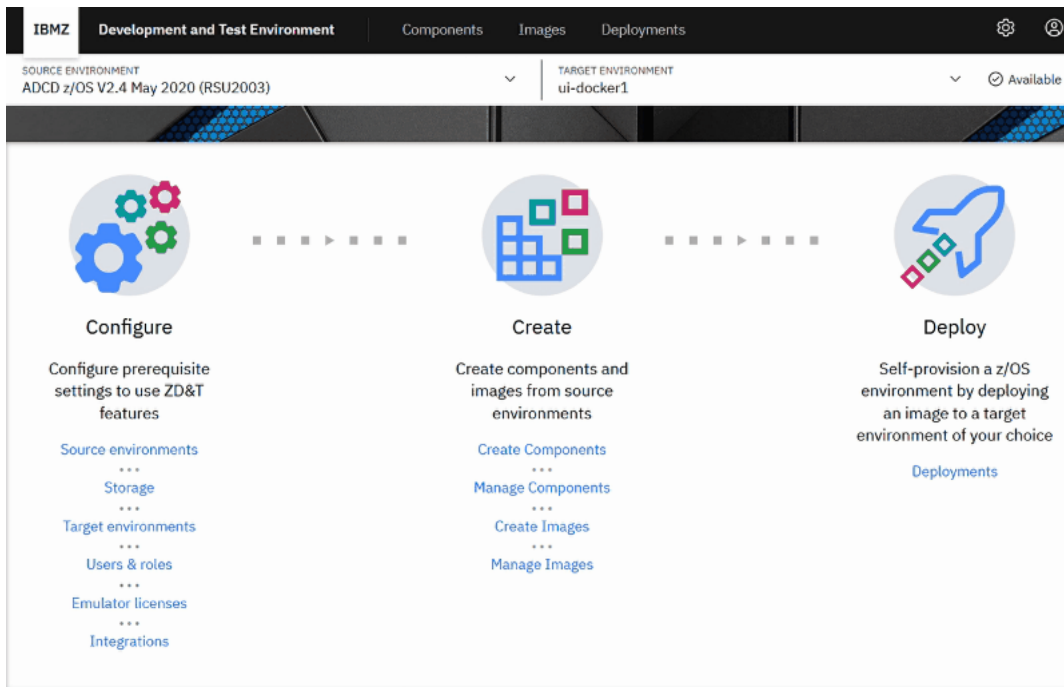
Tooltips

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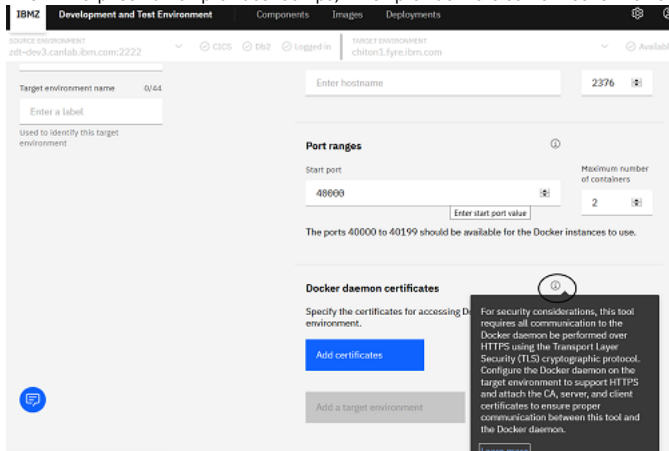
Creating components

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
- Volumes from an existing z/OS instance that ZD&T runs on a host Linux system.

Notes:

- Restriction: TCP/IP profile data sets that are sequential or PDSE are not supported. Only PDS is supported now.
- For any extraction from z/OS, the z/OS system requirements must be met. Additionally, for Db2 extraction, the Db2 extraction requirements must be met.

- **Creating a data set component**
  Learn how to create a data set component.
- **Creating a volume component**
  Learn how to create a volume component.

- **Creating a CICS component**
  After you configured an IBM Z mainframe system with CICS regions, you can create CICS components.
- **Creating a Db2 component**
  After you configured an IBM Z mainframe system with Db2 subsystems, you can create Db2 components.
- **Creating an IBM AD component**
  Learn how to create a component from an IBM Application Discovery for IBM Z (IBM AD) project.
- **Creating a volume component from an existing z/OS instance that ZD&T runs on Linux**
  After you configure an existing z/OS instance that ZD&T runs on a host Linux® system, you can create a volume component from the source environment with faster speed.

Creating a data set component

Learn how to create a data set component.

To create a data set component, 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. DASDVOL is checked for only DASD level operations, for example, dumping or restoring an entire volume. It is not checked for data set level operations.

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. To extract GDS data sets, make sure that the GDG base is also included.
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.

    If you try to extract a data set that has been opened in the modification mode, the extraction might fail with an error message like this one: `0ADR412E (001)-DTDSC(03), DATA`
    `SET xxx.xxx.xxx IN CATALOG xxx.xxx.xxx ON VOLUME xxxxxx FAILED SERIALIZATION`. The solution is to add the following line in *<installation_directory>*/zdt-server/server.env, and then restart the web server. You must be granted the appropriate access to set `TOL(ENQF)`.

    ```
    DSN_EXTRACT_TOLENQF=true
    ```

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 SOURCE ENVIRONMENT list.
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. From the Component type list, select Volume.
5. Enter a unique name that can be used to identify the volume component.
6. Optional: If you want system residence files to be contained in the component, select the Contains the system residence file(s) checkbox, and then optionally minimize the system residence volumes by using the automated system volume discovery tool.
   Note: The automated system volume discovery tool can identify the system residence volumes that are considered necessary to IPL the base z/OS® based on critical system data sets, such as the system link list, authorized program facility list, master catalog, and z/OS UNIX System Services file systems. To use this tool, make sure that you have completed all the steps in [Optional: Installing ZD&T z/OS Extraction Utilities](#).
   a. Click Discover system volumes.
   b. In the Enable automated system volume discovery window:
      i. Optional: Expand Discovery settings.
      ii. Optional: If you want to exclude data sets with particular high-level qualifiers (HLQs) from the discovered system volume list, select these HLQs in the High-level qualifiers table.
      iii. Optional: Select the Override the current selection with the discovered values checkbox to override the volumes that are being selected if there is any.
      iv. Click Run.
         The system residence volumes identified by the automated system volume discovery tool are highlighted in the Volumes table with an additional Reasons for selection column. In this column, you can see the corresponding system residence volume is important to the IPL of which system data sets and use it as a reference for selecting volumes that you need.
7. Select all the volumes that are needed to create the component.
8. Optional: Enable the Adjust advanced settings option, and then 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.
9. Optional: Enter some notes that can help you identify 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 the Schedule extraction icon to specify the date and time that you want to create your component.

After the volume component is created:

- All extracted artifacts will be stored on the system that you configured for storage.
- The type of the volume component that contains system residence files is System residence volume. Otherwise, the type of the volume component is Volume.

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. At least one file control table (FCT), source data set, or load module data set must be selected.
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 [Hardware and software requirements](#) and [Limitations](#).

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 IBM AD component

Learn how to create a component from an IBM® Application Discovery for IBM Z (IBM AD) project.

Before you create components from IBM AD 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.
- The connection is set up and tested through the Integrations page. For more information, see Optional: Configuring IBM AD integrations.

To create components from IBM AD 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 only supports data sets whose host names are entered. The data sets from IBM AD 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 AD projects, if errors occur, see Troubleshooting for creating components from IBM AD projects for reference.

Creating a volume component from an existing z/OS instance that ZD&T runs on Linux

After you configure an existing z/OS instance that ZD&T 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 such a source environment, ensure that the following two requirements are met.

- The z/OS instance on Linux is up and running.
- The latest z/OS® Extraction Utilities is installed in the source environment. For detailed instructions, see Optional: Installing ZD&T z/OS Extraction Utilities.

To create a volume component, complete the following steps.

1. Select a Linux running a ZD&T instance environment that you want to connect to 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 existing z/OS instance on Linux.

   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 .

Monitoring and managing the 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 Manage components on the home page, or click the Components tab on the header.
2. In the Components table, find the component that you want to update, and then click the Edit icon.
3. Change the component name if you want.
4. Optional: Enable the Adjust advanced settings option, and then enter an integer in the box of Concurrent extractions to define the number of artifacts that you want to extract simultaneously.
5. Modify the volumes or data sets that you selected, or reselect the volumes or data sets for the component. If you are updating a volume component that contains system residence files, you can optionally minimize the system residence volumes by using the automated system volume discovery tool.

   Note: The automated system volume discovery tool can identify the system residence volumes that are considered necessary to IPL the base z/OS® based on critical system data sets, such as the system link list, authorized program facility list, master catalog, and z/OS UNIX System Services file systems.
   a. Click Discover system volumes.
   b. In the Enable automated system volume discoverywindow, complete the following steps:
      i. Optional: Expand Discovery settings.
      ii. Optional: If you want to exclude data sets with particular high-level qualifiers (HLQs) from the discovered system volume list, select these HLQs in the High-level qualifiers table.
      iii. Optional: Select the Override the current selection with the discovered values checkbox to override the volumes that are being selected if there is any.
      iv. Click Run.

         The system residence volumes identified by the automated system volume discovery tool are highlighted in the Volumes table with an additional Reasons for selection column. In this column, you can see the corresponding system residence volume is important to the IPL of which system data sets and use it as a reference for selecting volumes that you need.
6. Click Update component, or click the Schedule extraction icon 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.

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 are 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 is created from a previous version contains mini volumes, the migration icon is 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 are created from IBM Z volumes or data sets support editing, creating new versions, or duplicating functions. For the components that are created from IBM Z CICS® regions and Db2® subsystems, these functions are not supported.

Creating images

An image is a collection of components that are packaged together from your source environment for deploying to your target environment. When creating an image, you must select to include one component of type system residence volume and any number of other components that 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 runs on emulated IBM Z® hardware from the image..

You can use the web server to create images from various source environments. To create an image from a different source environment, change your source environment in the source environment drop-down list.

- Existing IBM® Z components.
- 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 z/OS instance that ZD&T runs on Linux.

- **Creating z/OS images**
  After you create components from IBM Z mainframe system or existing z/OS instances, you can create z/OS images by combining the created components into a single package.
- **Creating ADCD images**
  Learn how to create images from Extended ADCD packages.
- **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.

Creating z/OS images

After you create components from IBM Z® mainframe system or existing z/OS instances, you can create z/OS® images by combining the created components into a single package.

To create an image, complete the following steps:

1. Select an IBM Z mainframe or an existing z/OS 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 emulator 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. If the data sets included in the selected components are related to user catalogs that are not part of the image, a warning message is displayed. Update components to ensure that the necessary user catalogs are extracted and included.
   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 Extended ADCD packages.

ZD&T Enterprise EditionZVDT supports creating an image by selecting the existing Extended ADCD 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 Extended 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 an emulator 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 Extended 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 Extended 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 Extended 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 Extended 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 covert. 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:
      - 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.
      - Override the IPL command if you want. Specify the attributes of the device address, IODF volume device address, and LOADxx suffix.
      - Specify the port of the service that is running on the z/OS that you want to expose.
11. Click Convert image.

Monitoring and managing the 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 Extended 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 advanced 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:

- You can edit the properties of only the images that are created from z/OS environments extracted from your on-premise IBM Z systems.
- 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. If a different system residence volume component is selected in step 5, the "Systems residence volume selection" dialog is displayed after you click Create image. Confirm whether you want to keep the image properties from the previous version or import the properties from the selected component, and then click OK.

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 Extended 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 instance, the components that are used in the image will also be deleted.

Migrating images to the current version

For images that are created in V12.0.1 or earlier versions with data set or CICS® components, migration is needed. Go to the Images page to verify whether the migration icon ⤷ is displayed for an image. If the icon exists, the image cannot be provisioned until it is migrated.

To migrate images, complete the following steps:

1. On the Images page, select Perform image extraction from the action menu ⋮ .
2. Authentication might be needed. Enter your user ID and password for the selected source environment, and then click Submit. After authentication, retry step 1.
3. The "Perform image extraction" dialog is displayed. Click Yes, perform extraction.

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 content 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 Extended 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.

Provisioning instances to target environments

Learn how to provision z/OS instances to target environments.

The provisioning options in the web server will install and configure the 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 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 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 SFTP as the storage server, the port number must be 22.
7. User and group configurations
   If you did not select to automatically configure users and groups when you configured the Linux target environments, the pre-provisioning validation function will check the user and group configurations by using zdt_congfig_user_byRoot.sh.
   - Group
     The function will check whether the group 'zpdt' 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 'zpdt'.
     - 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 'zpdt'.
8. Dependent libraries
   - If you did not select to install all dependencies when you ran the ZD&T installer, this function will check the following dependent libraries by using `zdt_install_dependencies_byRoot.sh`.
     - REHL operating system
       iptables
       sftp
       libstdc++.i686
       perl
       zip
       unzip
       gzip
     - Ubuntu operating system
       iptables
       libc6:i386 libncurses5:i386 libstdc++6:i386 lib32z1 lib32stdc++6
       sftp
       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`.
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.
- **Provisioning instances to a Linux environment**
  Learn about the steps to provision instances to a Linux target environment.
- **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.

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:
     - 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.
     - Override the IPL command if you want. Specify the attributes of the device address, IODF address, and LOADxx suffix.
     - Specify the private port of your Docker environment.
8. Click Provision instance.

Provisioning instances to a Linux 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 then 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 then 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 /[home directory]/.ssh/authorized_keys file of the user ID in the target environment.
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.

8. To provision an instance that contains data set components, 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 then click Provision instance to start the provisioning.
   - Enable the Advanced settings option, and then complete the following optional steps:
     - 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.
     - 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.

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 then 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:
     - Creating z/OS images or Creating ADCD images.
     - If the image has multiple versions, then select a version.
     - 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 ready image by following the steps at Converting an image to a cloud-ready image.
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:
     - 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.
     - Override the IPL command if you want. Specify the attributes of the device address, IODF volume device address, and LOADxx suffix.
     - Specify the port of the service that is running on the z/OS that you want to expose. The SSH port name must be ssh, which is used to run all post-IPL tasks that need to connect to the target environment. If you use another SSH port number other than 22, set Expose SSH port (:22) to off, and then define your SSH port with a name that is exactly "ssh" in the Custom ports section.
6. Click Provision instance to start the provisioning.

Monitoring and managing the provisioned instances

After you provision a z/OS® instance 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. Follow the instructions to enter credentials and configuration information. For more information about provisioning instructions, see Provisioning instances to target environments.
4. Click Provision instance.

Starting the z/OS instance

If the z/OS instance is offline or stopped, start the z/OS instance by completing the following steps:

1. Click the action icon beside the instance, and select Start z/OS.
2. Enter the required credentials, and click Submit.

Stopping the z/OS instance

To stop a running z/OS instance, complete the following steps:

1. Click the action icon beside the instance, and select Stop emulator.
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.

Using REST APIs

ZD&T Enterprise Edition supports REST APIs usage. You can access all functions of ZD&T Enterprise Edition without the web user interface, including creating components and images, provisioning instances from created images, and monitoring images and instances.

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 web 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 [REST API error codes](#) to get the corresponding error messages. The parameters that are 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/zdt-server/bootstrap.properties.
  2. Run the following command where you install the web server.

     ```
     /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/zdt-server/server.xml. Then, replace the text in **bold** with the new encoded password value.

     ```
     <!-- user authentication -->
     <basicRegistry
     id="_home_markdown_jenkins_workspace_Transform_in_SSTQBD_14.2.0_com.ibm.zsys.rdt.tools.user.guide.doc_topics_api_usage_zdt" realm="zdt">
     <user name="zdtadmin"
     password="{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB"/>
     </basicRegistry>
     ```

- **Adding REST APIs through Zowe API Gateway**
  The ZD&T Enterprise Edition installer contains a YAML file that can be used to add ZD&T Enterprise Edition REST APIs through Zowe™ API Gateway.
- **Connecting to target environments with REST APIs**
  To connect to target environments, you can use 'curl' to call REST APIs.
- **Creating and managing target environments with REST APIs**
  You can call REST APIs to create and manage target environments.
- **Creating components with REST APIs**
  If you want to extract data from z/OS® Systems, you must create components before you create images.
- **Checking the status of the created components with REST APIs**
  After the components are created, you can call the REST API to check the status of the created components.
- **Editing created components with REST APIs**
  After you create volume or data set components from z/OS systems, you can edit the component, or create a new version and duplicate for the component.
- **Creating images with REST APIs**
  Learn about the steps to create images by using REST API. After the images are created, you can store the image data in the specified storage server.
- **Checking the status of the created images with REST APIs**
  After the images are created, you can call the REST APIs to check the status of the created images.
- **Editing the created images with REST APIs**
  After you create images, you can edit the image, or create a new version or a duplicate of the image.
- **Provisioning instances to target environments with REST APIs**
  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® or Docker target environment.
- **REST API error codes**
  If an error occurs when you use REST APIs, refer to the response error codes for detailed description.

Adding REST APIs through Zowe API Gateway

The ZD&T Enterprise Edition installer contains a YAML file that can be used to add ZD&T Enterprise Edition REST APIs through Zowe™ API Gateway.

To add ZD&T Enterprise Edition REST APIs through Zowe API Gateway, complete the following steps.

1. Run installer zdt-install.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 ZD&T Enterprise Edition 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](#).

Connecting to target environments with REST APIs

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

Configure the connection to the target environments if you want to provision instances from created images to Linux® environments.

```
curl -X 'POST' \
 'https://your_tools_server_host_name:9443/zdtrest/zdtrs/target-environment-services/v1/linux-x86_64' \
 -H 'accept: application/json;charset=utf-8' \
 -H 'Content-Type: application/json;charset=utf-8' \
 -d '{
"hostname": "mytarget.domain.com",
"ssh-port": 22,
"label": "zdtForTestingCics",
"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": "172.26.1.2",
"tcp-port-rules": "0:21 \n 23:2021 \n 2022>22 \n 2023:3269 \n 3271:9449 \n 9452:65535",
"udp-port-rules": "111",
"zdt-managed": true,
"has-root-access": true,
"description": "string"
}'
```

Creating and managing target environments with REST APIs

You can call REST APIs to create and manage target environments.

- **Creating and managing Linux target environments with REST APIs**
  You can call REST APIs to create and manage Linux® target environments.
- **Creating and managing Docker target environments with REST APIs**
  You can call REST APIs to create and manage Docker target environments.
- **Creating and managing Red Hat OpenShift target environments with REST APIs**
  You can call REST APIs to create and manage Red Hat® OpenShift® target environments.

Creating and managing Linux target environments with REST APIs

You can call REST APIs to create and manage Linux® target environments.

About this task

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' \
  'https://your_tools_server_host_name:9443/zdtrest/zdtrs/target-environment-services/v1/linux-x86_64' \
  -H 'accept: application/json;charset=utf-8' \
  -H 'Content-Type: application/json;charset=utf-8' \
  -d '{
  "hostname": "mytarget.domain.com",
  "ssh-port": 22,
  "label": "zdtForTestingCics",
  "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": "172.26.1.2",
  "tcp-port-rules": "0:21 \n 23:2021 \n 2022>22 \n 2023:3269 \n 3271:9449 \n 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/target-environment-services/v1/linux-x86_64/<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 with REST APIs

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://zyour_tools_server_host_name:9443/zdtrest/zdtrs/target-environment-services/v1/docker-system' \

-H 'accept: application/json;charset=utf-8' \

-H 'Content-Type: multipart/form-data' \

-F 'docker-system={

"hostname": "string",

"port": 0,

"lower-port-value": 0,

"passphrase": "string",

"max-containers": 5,

"label": "string",

"description": "string",

"concurrent-transfers": 1

}' \

-F 'ca-certificate=' \

-F 'server-certificate=' \

-F 'client-key='
```

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": "string",

"creation-time": "2024-03-25T07:11:02.624Z",

"deletable-details": {

"deletable": true,

"explanations": [

"string"

]

},

"label": "string",
```

```
    "description": "string",
    "hostname": "string",
    "port": 0,
    "concurrent-transfer": 0,
    "type": "DOCKER",
    "status": "NEW",
    "online": false,
    "user": {
        "uuid": "string",
        "creation-time": "2024-03-25T07:11:02.624Z",
        "name": "string",
        "show-welcome": true,
        "show-usage-collection": true,
        "login-time": "2024-03-25T07:11:02.624Z",
        "login-count": 0,
        "modifiable": true,
        "roles": [
            {
                "uuid": "string",
                "creation-time": "2024-03-25T07:11:02.624Z",
                "name": "string"
            }
        ]
    },
    "support-multiple-instances": true,
    "lower-port-value": 0,
    "upper-port-value": 0,
    "max-containers": 0
}
```

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
  }s
}
```

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/zdtrs/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/target-environment-services/v1/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 with REST APIs

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/target-environment-services/v1/openshift-system' \
```

```
  -H 'accept: application/json;charset=utf-8' \

  -H 'Content-Type: application/json;charset=utf-8' \

  -d '{

  "url": "string",

  "token": "string",

  "label": "string",

  "namespace": "string",

  "storage-class": "string",

  "cloning": false,

  "description": "string",

  "concurrent-transfers": 1

}'
```

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": "string",

  "creation-time": "2024-03-25T08:35:38.637Z",

  "deletable-details": {

    "deletable": true,

    "explanations": [

      "string"

    ]

  },

  "label": "string",

  "description": "string",

  "hostname": "string",

  "port": 0,

  "concurrent-transfer": 0,

  "type": "DOCKER",

  "status": "NEW",

  "online": false,

  "user": {

    "uuid": "string",

    "creation-time": "2024-03-25T08:35:38.637Z",

    "name": "string",

    "show-welcome": true,

    "show-usage-collection": true,

    "login-time": "2024-03-25T08:35:38.637Z",

    "login-count": 0,

    "modifiable": true,

    "roles": [

      {

        "uuid": "string",

        "creation-time": "2024-03-25T08:35:38.637Z",

        "name": "string"

      }

    ]

  },

  "support-multiple-instances": true,
```

```
  "token": "string",
```

```
    "token": "string",

    "console-url": "string",

    "protocol": "string",

    "namespace": "string",

    "storage-class": "string",

    "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": "OPENSHIFT",
    "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": "OPENSHIFT",
    "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/target-environment-services/v1/openshift-system/uuid?cleanup=true' \

  -H 'accept: application/json;charset=utf-8'
```

If the REST API call is successful, the response displays 200 (OK) status code.

Creating components with REST APIs

If you want to extract data from z/OS® Systems, you must create components before you create images.

- **Creating volume components from z/OS system with REST APIs**
  Learn about the steps to create components from IBM Z® volumes by using REST API.
- **Creating data set components from z/OS system with REST APIs**
  Learn about the steps to create components from IBM® Z data sets by using REST API.
- **Creating CICS components from z/OS system with REST APIs**
  Learn about the steps to create CICS® components from z/OS system by using REST API.
- **Creating Db2 components from z/OS system with REST APIs**
  Learn about the steps to create Db2® components from z/OS system by using the REST API.
- **Creating data set components from z/OS for projects defined in IBM AD with REST APIs**
  Learn about the steps to create components from IBM Application Discovery by using REST API.
- **Creating volume components from existing ZD&T or instances**
  Learn about the steps to create volume components from existing ZD&T or instances.

Creating volume components from z/OS system with REST APIs

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":1892010960},
{"id":0,"name":"A27CCC","extractionTime":"1960-01-01
23:03:20","deviceNumber":"04BE","deviceType":"3390","progress":0,"compressed":0,"size":2838016440},
{"id":0,"name":"A45SYS","extractionTime":"1960-01-01
23:03:20","deviceNumber":"0492","deviceType":"3390","progress":0,"compressed":0,"size":8514049320},…]
```

3. Create the component by including 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 with REST APIs

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 with REST APIs

Learn about the steps to create CICS® components from z/OS® system by using REST API.

Procedure

1. Configure CICS to add the information of any CICS regions that you want to create components from by using the web server. For more information, see Adding a source environment from a z/OS system on IBM Z physical hardware.
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 with REST APIs

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",
        "sdsnexit": "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",
        "sdsnexit": "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","SYSTOO
LS"]
```

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":"PLANR
TAB","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","selectedTab
les":
[{"name":"PLAN_TABLE","tableType":"TABLE","schemaName":"your_db2_schema_name","databaseName":"your_db2_database_name"}],"i
ntegrity":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 IBM AD with REST APIs

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":"dXNlcjM1NTg5OTg9OTgwNA==","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 volume components from existing ZD&T or instances

Learn about the steps to create volume components from existing ZD&T or 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 or 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 the created components with REST APIs

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 z/OS instances to replace the following *italic* text.

```
curl -k -u zdtadmin: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.

Editing created components with REST APIs

After you create volume 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":"VVNFUjM1NTE4ODczMjY3NjU=","port":22}
```

2. Run the following command to submit the request.

```
curl -k -u zdtadmin:password -X PUT --header 'Content-Type: application/json' --header 'Accept: application/json' --header
'SSHAuthorization: {"token":"dXNlcjM1NTg5OTg4OTgwNA==","hostname":"YOUR_Z_OS_HOST_NAME","port":22} ' -d ' {"name": "name
of the 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'
```

Creating images with REST APIs

Learn about the steps to create images by using REST API. 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 with REST APIs**
  Learn about the steps to create images from z/OS® volumes or data sets.
- **Creating images from Extended ADCD with REST APIs**
  Learn about the steps to create images from Extended ADCD by using REST APIs.

Creating images from z/OS volumes or data sets with REST APIs

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 zdtadmin: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 REST API error codes.

2. Optional: Try again when an error code is displayed at step 1.

```
curl -k -u zdtadmin: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 images from Extended ADCD with REST APIs

Learn about the steps to create images from Extended ADCD by using REST APIs.

Procedure

1. Copy your Extended 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 Extended ADCD package and provision an instance from the created image.
2. Get the Extended 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/zdtrs/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 Extended ADCD package, and provide the sample subfolder name *<version>* as the source host name at the end.

```
curl -k -u zdtadmin:password -X GET 'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/componentServices/components?
host=<version>' --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": "<version>",
       "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 the Extended ADCD package**
     If you want to create an image from the Extended ADCD package, select the components as found from the above call (provide the sample subfolder name `<version>` 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": "<version>",     "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 the Extended ADCD package and data sets from other z Systems®**
     If you want to create an image from the Extended 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": "<version>",     "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 the Extended ADCD package and CICS® components**
     If you want to create an image from the Extended 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": "<version>",     "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 the created images with REST APIs

After the images are created, you can call the REST APIs to check the status of the created images.

Procedure

Run the following command, and use the variables of your z/OS instances to replace the following *italic* text.

```
curl -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 the created images with REST APIs

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":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
```

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":{},"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":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 to target environments with REST APIs

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® or Docker target environment.

- **Provisioning instances to a Linux target environment with REST APIs**
  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.
- **Provisioning instances to a Docker target environment with REST APIs**
  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.
- **Provisioning instances to a Red Hat OpenShift target environment with REST APIs**
  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.

Provisioning instances to a Linux target environment with REST APIs

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

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, see Creating and managing Linux target environments with REST APIs.

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 zdtadmin: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 zdtadmin: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 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","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 zdtadmin: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":"cm9vdC0xMzM5NzAzMjQ2Q2","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":"cm9vdC0xMzM5NzAzMjQ2Q2","hostname":"your_target_system_host_name","port":22}' -d '{"targetSystem":
    {"hostname": "your_target_system_host_name","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 zdtadmin:password -X GET --header 'Accept: application/json'
'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/targetEnvServices/targetSystem/your_target_system_host_name'
```

If the status is `"status": "DEPLOYED"`, the 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 zdtadmin:password -O -X GET --header 'Accept: application/zip'
'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/deployServices/zipAllLogs?
targetHostname=your_target_system_host_name&sourceHostname=your_appl_source_host_name&sourcePort=22&applicationN
ame=testZdtAppl'
```

Then, you can get a .zip file that contains all related logs.

   ii. Extract the file to read the logs.

Provisioning instances to a Docker target environment with REST APIs

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/zdtrs/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 with REST APIs

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 with REST APIs.

- **Provisioning instances from a created image with REST APIs**
  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.
- **Converting a created image to a cloud-ready image with REST APIs**
  To provision an instance from a cloud-ready image, you need to covert a created image to a cloud-ready image first.
- **Provisioning instances from a cloud-ready image with REST APIs**
  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.

Provisioning instances from a created image with REST APIs

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 with REST APIs

To provision an instance from a cloud-ready image, you need to covert 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 with REST APIs

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}'
```

REST API 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 variables. The variables will be replaced by the contents that are displayed in the messages.

Table 1. Response error codes

Error ID	Description
403	This account is locked out for *{0}* 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 *{0}* already exists.
10008	The Licensing system *{0}* 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.
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.

Error ID	Description
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.
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}*.
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.

Error ID	Description
20003	User already exists.
20004	User does not exist.
21001	The machine running the ZD&T web 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.
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.
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, 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.

Error ID	Description
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}.
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.
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.

Error ID	Description
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.
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.
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.
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.

Error ID	Description
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 'sftp' 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.
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}*.
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.

Error ID	Description
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.
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.
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.
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.

Error ID	Description
31750	Privilege command option already exists.
31751	The privilege command syntax is required to be defined on the *<a href|*.
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.
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 /etc/resolve.
50116	Failed to read the content of /etc/hosts file in container.
50117	Could not resume provision for *{0}* because it is not a failed provision.
50118	The *{0}* cannot be deprovisioned with status *{1}*.
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.
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.

Error ID	Description
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}.
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.

Scenarios

Learn about the scenarios of ZD&T Enterprise Edition for development and test activities.

- **Provisioning a CICS subsystem from a source Z**
  Learn how to provision a CICS® subsystem from a source z System.

Provisioning a 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 Optional: Configuring IBM AD integrations.
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 IBM AD component.
4. Create the image. For more information, see Creating z/OS images.
5. Provision an instance from a created image to the target environment. For more information, see Provisioning instances to target environments.

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

- BPXPRMxx has a list of 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
```

- LPALSTxx has a list of LPA data sets. It is defined in IEASYSxx as LPA parameter. For example:

  ```
  LPA=00, SELECT LPALST
  ```

- PROGxx has a list of other APF that authorizes module load libraries that might not be in the CICS startup JCL. It is defined in IEASYSxx as follow:

  ```
  PROG=(AB,AM,A0,A2, DYNAMIC APF LIST
  SY,LB,LM,L0,L2,LL), LINKLIST LL SUFFIX SHOULD BE LAST
  ```

Troubleshooting and workarounds

Learn how to troubleshoot ZD&T Enterprise Edition for development and test activities.

- **Finding logs**
  You can refer to log files for troubleshooting. This topic provides an overview of the key log files that are associated with ZD&T.
- **Troubleshooting for configuring the IBM AD server**
  Learn how to troubleshoot the application components creation from IBM Application Discovery for IBM Z (IBM AD) projects.
- **Troubleshooting for creating components from IBM AD projects**
- **Troubleshooting for provisioning issues**
  Learn how to troubleshoot for provisioning issues.
- **Restoring empty VSAM data sets**
  Learn about the workaround of restoring empty VSAM data sets.

Finding logs

You can refer to log files for troubleshooting. This topic provides an overview of the key log files that are associated with ZD&T.

On the Linux system where the web server is installed

/var/log/zdt*.log
    Records events and activities of the web server installer.
<installation_path>/zdt/zdt-server/logging.xml
    Specifies the logging level configurations of the web server.
<installation_path>/zdt/zdt-server/logs/*.log
    Records events and activities within the web server.
<installation_path>/zdt/zdt-server/postgresql/logfile
    Records database-related issues that might occur when attempting to start the web server.
<installation_path>/zdt/zdt-server/postgresql/zdt_db_conf.log
    Records database-related issues that might occur when attempting to start the web server.

On the target Linux system where z/OS instances are provisioned

<deployment_directory>/zdt/log/
    Records provisioning events and activities.
/home/ibmsys1/z1090/logs/
    For provisioning by the root user ID. Records events and activities of the emulator, with the target z/OS console logs stored in files that are named log_console*.txt.
<home_directory>/z1090/logs/
    For provisioning by non-root user IDs. Records events and activities of the emulator, with the target z/OS console logs stored in files that are named log_console*.txt.

OpenShift pod logs

Important logs for Sandbox are located on the operator pod and each WaziSandboxSystem instance pod or WaziSandboxVolumeCopy pod. The general pod logs can be accessed through the OpenShift web console or the `oc logs` command. To view the logs, you must have access to the OpenShift pod according to role-based access control (RBAC). If you do not have access, a subset of logs can be retrieved by following the commands in topic Collecting MustGather logs.

For WaziSandboxSystem instance pods, depending on your configuration, one or two additional logs might be generated for init containers.

For WaziSandboxSystem instances, the following logs might also be available:

/zdt/wlp/usr/servers/zdt-instance/logs/
    Records events and activities of the instance controller (IC) server.
/zdt/zvolumes/log.txt
    IPL log file. Records container startup and shutdown events and activities.
/home/ibmsys1/z1090/logs/log_console*.txt
    Stores the target z/OS console logs.

/zdt/zvolumes/ipl-zos-*.log
    IPL log files. Records messages that are specific to IPL.

/zdt/zvolumes/shutdown-emulator-*.log
    IPL log files. Records messages from shutting down the emulator.

/zdt/zvolumes/shutdown-zos-*.log
    IPL log files. Records messages from attempts to stop z/OS with the shutdown command.

/zdt/zvolumes/start-emulator-*.log
      IPL log files. Records status of the running pod with messages from the emulator and z/OS.

Docker container logs

/zdt/wlp/usr/servers/zdt-instance/logs/
      Records events and activities of the instance controller (IC) server.
/zdt/zvolumes/*.log
      IPL log files. Records startup and shutdown events and activities of the Pod.
/zdt/zvolumes/log.txt
      IPL log file. Records container startup and shutdown events and activities.

On the Linux system where the license server is installed

/home/ibmsys1/*.log
/home/ibmsys1/UIMserver/
/var/hasplm/error.log
/var/hasplm/log/

On the source z/OS system where ZD&T z/OS Extraction Utilities are installed

/tmp/feucdebug_log.HH.MM.SS.ssssss or /tmpDir/feucdebug_log.HH.MM.SS.ssssss
      Records events and activities of the volume discovery tool.
/tmp/ZDT_DB2X_PARMFILE_XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
      Records Db2 extraction events and activities.
/tmp/zdtdb2x_XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
      Records Db2 extraction events and activities.

On the provisioned z/OS system

/tmp/ZDT_DB2X_PARMFILE_XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
      Records Db2 restoration events and activities.
/tmp/zdtdb2x_XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
      Records Db2 restoration events and activities.

Troubleshooting for configuring the IBM AD server

Learn how to troubleshoot the application components creation from IBM® Application Discovery for IBM Z (IBM AD) projects.

Table 1. Troubleshooting the configuration of the IBM AD server

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

Troubleshooting for creating components from IBM AD projects

Table 1. Troubleshooting the components creation from IBM Application Discovery for IBM Z (IBM AD) 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 Managing Application Projects with IBM AD Build to refresh or rebuild the project.
The selected data set list is required.	Select the data set list on the right panel of the page.
The port number is not valid.	No character is allowed.

Troubleshooting for provisioning issues

Learn how to troubleshoot for provisioning issues.

Symptom	Problem determination
When your target environment is set up without the automated system configuration enabled, and the target environment uses a non-sudo privilege management platform, attempted provision fails at step configuring users and groups.	If the automated system configuration is not enabled, you need to manually configure users and groups. For instructions, see User and group configurations. Or you can enable the automated system configuration when adding your target environment on the web server. For instructions, see Adding target environments.
When you provision an instance with a valid license server hostname, somewhere in the meantime, the license server hostname becomes invalid, for example, by the license server going down. After connecting the license server, the provision cannot be resumed.	Deprovision the instance and provision again for the corrected license server hostname to take effect.
When no offline DASD devices are available in the source z/OS during extraction, although the extraction might be successful, the provision of a new z/OS instance using this extracted image fails.	During the provision of a new z/OS instance, a volume called ZDTDPY is created. This volume needs to be mounted on an offline DASD device in the source z/OS. Therefore, there must be at least one offline DASD device available when provisioning. You can follow the steps to solve the issue: 1. Add an offline DASD device to the source z/OS volume list. 2. Create a new version of the SYSRES component that contains the offline DASD device. 3. Create a new version of the image to replace the SYSRES component with the version created in step 2. 4. Update the provision with the image created in step 3.

Restoring empty VSAM data sets

Learn about the workaround of restoring empty VSAM data sets.

Symptom	Problem determination
When you try to restore an empty VSAM data set that contains a lot of unused but occupied space, or has a smaller size and was heavily used, the restoration fails with the "Broken pipe" error message.	When ZD&T calls DFDSS to dump the data set, the dumped image contains a lot of unused space, while the DFDSS restoring does not use the space. So the restoration thread completes, while the transfer thread is still trying to send data to the restoration thread. You can follow the steps to solve the issue: 1. For an empty VSAM data set that contains a lot of unused but occupied space, redefine the VSAM file size to a smaller value. For an empty VSAM data set that has a smaller size and was heavily used, redefine the VSAM file size to the exact value. 2. Re-extract the VSAM file by creating a new component version. 3. Create a new image version to contain the new component version. 4. Update the provision with the new image version.

Sharing feedback

You can share anonymous data on the web server to help shape the product.

Your feedback matters! To provide feedback, log in to the web server, and then click the upper-right Information menu > Feedback.

Uninstalling

Learn how to uninstall ZD&T.

- **Uninstalling the web server**

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 IBM License Key Center, and then generate a new one. Returning the license entitlements is a process of telling the IBM 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.

- **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 IBM License Key Center, and then generate a new one. Returning the license entitlements is a process of telling the IBM 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 IBM License Key Center.
- **Returning Rational Tokens for software-based license server**
  To generate a new update file, you must first return the existing license entitlement in the IBM License Key Center, and then generate a new update file. Returning the license entitlements is a process of telling the IBM 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 IBM License Key Center.

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 IBM License Key Center, and then generate a new one. Returning the license entitlements is a process of telling the IBM 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 IBM 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 IBM License Key Center.
3. Log in to the IBM 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 IBM 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 IBM 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 IBM 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 IBM License Key Center.

You can return a license entitlement in the IBM 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 more 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 IBM License Key Center.
2. Obtain a license key file that invalidates the current license key from the IBM License Key Center.
   a. Log in to the IBM 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 IBM 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 IBM 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 IBM License Key Center.

4. Use the verification file to complete the return process in the IBM License Key Center.

   a. Log in to the IBM 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 IBM License Key Center, and then generate a new update file. Returning the license entitlements is a process of telling the IBM 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 IBM License Key Center.

1. Log in to IBM 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.

Uninstalling the software-based license server

Before you uninstall the software-based license server, you must return the existing license key by following the steps in Returning a software-based license key.

To uninstall the software-based license server, complete the following steps:

1. Stop the software-based license server with the root user ID.

   `service hasplmd stop`

2. Stop the UIM server with a non-root user ID.

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

3. Open the directory where the installer zdt-install.tgz is stored.

   `cd <directory>`

   License server installer
       zdt-license-server

4. Run the installer with the root user ID, and then follow instructions to uninstall.

   `./<installer_name>`

   Alternatively, you can uninstall the software-based license server silently by using the following command.

   `./<installer_name> --uninstall`

5. After the uninstallation completes, run the following commands to verify whether the uninstallation is successful.

   RHEL on x86

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

Ubuntu Linux on x86

```
dpkg -l | grep zpdtldk

dpkg -l | grep zpdtuim
```

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

Uninstalling the web server

Before you uninstall the web server, you must complete the following steps first.

1. Return the existing license key by following the steps in Returning a software-based license key.
2. Stop the UIM server with a non-root user ID.

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

To uninstall the web server, complete the following steps:

1. Open the directory where the installation package zdt-install.tgz is stored.

```
cd <directory>
```

2. Change the authority of the installation package.

```
chmod 755 <package_name>
```

3. Decompress the installation package.

```
tar -xvf <package_name>
```

When the decompression is complete, the installers can be found in the target directory. If you don't specify a target directory, the installers can be found in the same directory where the installation package is.

Web server installer
    zdt-install

4. Run the installer with the root user ID, and then follow instructions to uninstall.

```
./<installer_name>
```

Alternatively, you can uninstall the web server silently by using the following command:

```
./<installer_name> --uninstall --zdtee
```

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 1. The software-based license server and various components of Parallel Sysplex

Software Based License Server
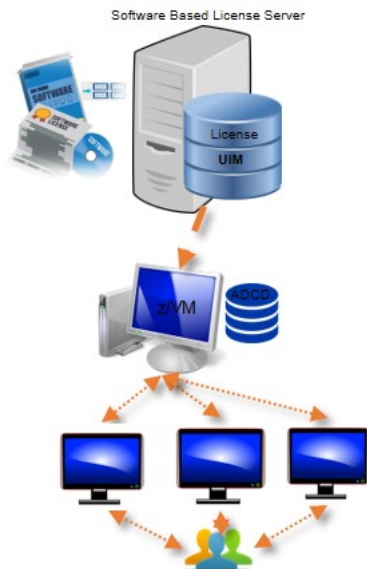
hardware-based license server

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 IBM 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 2. The hardware-based license server and various components of Parallel Sysplex

Hardware Based License Server

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 IBM 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 3. The hardware-based license server with Rational Tokens, and various components of Parallel Sysplex
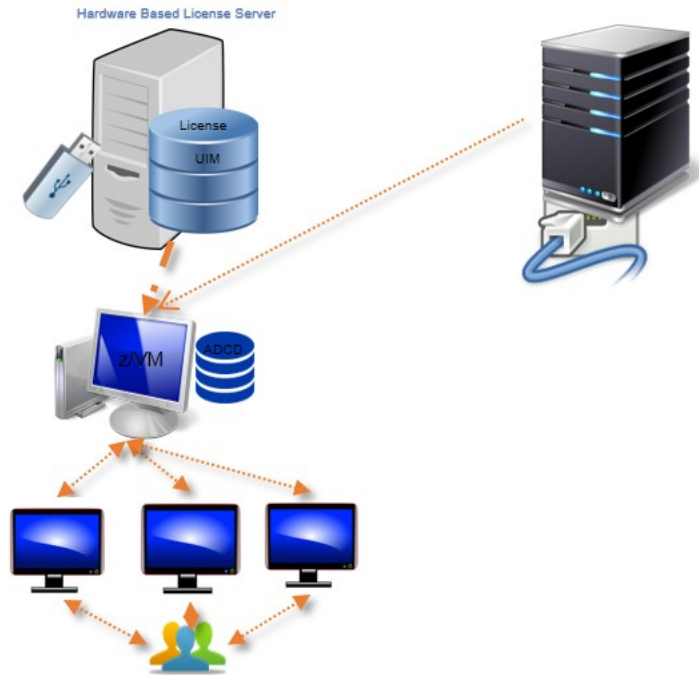
Figure 4. The software-based license server with Rational Tokens, and various components of Parallel Sysplex
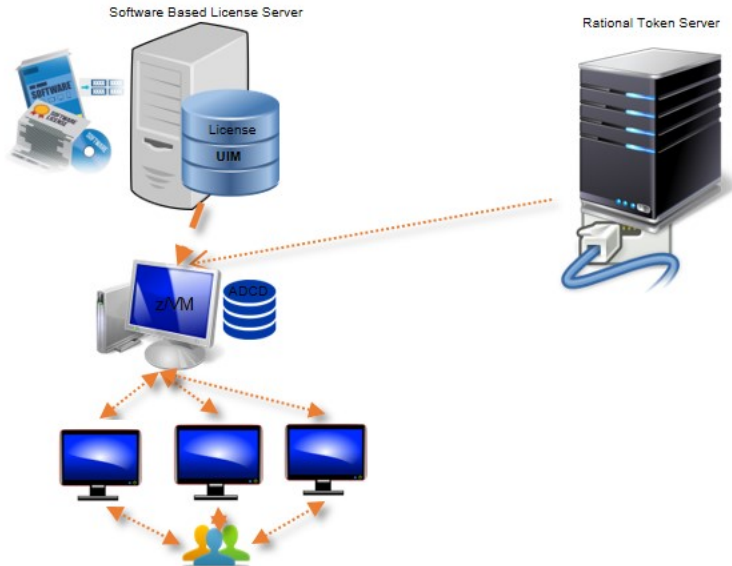
Application Developers Controlled Distribution (ADCD)

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.
- **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.

- **Installation packages**
  Learn about how to download installation packages.
- **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.
- **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.
- **Installing Parallel Sysplex**
  To install ZD&T Parallel Sysplex, run the installer from the root user ID.
- **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.
- **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.
- **Starting z/VM**
  To start z/VM, issue `awsstart` under `ibmsys1` user ID or equivalent home by using directory structure /usr/ibmsys1.
- **Starting Z instances**
  You need to start Z instances from an active 3270 session that is connected to z/VM.
- **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.
- **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.
- **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.

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 1. New installation checklist for Parallel Sysplex

	Installing Parallel Sysplex	Required/Optional	Complete
1	Prerequisites Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	☐
2	Obtaining installation packages Describes the steps to download all required packages to install Parallel Sysplex.	Required	☐
3	Enabling the license servers Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐
4	Installing Parallel Sysplex Describes the steps to install Parallel Sysplex.	Required	☐
5	Activating ZD&T Parallel Sysplex instances Describes the steps to activate ZD&T instances	Required	☐
6	Configuring Z instances Describes the steps to configure ADCD to start Z instances.	Required	☐
7	Starting z/VM Describes the steps to start z/VM.	Required	☐
8	Starting Z instances Describes the steps to start Z instances.	Required	☐
9	Stopping Z instances and z/VM Describes the steps to stop Z and z/VM.	Optional	☐

Table 2. Update checklist for Parallel Sysplex

	Updating Parallel Sysplex	Required/Optional	Complete
1	Returning a software-based license key or Returning a hardware-based license key Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	☐
2	Prerequisites Describes the software and hardware requirements, and the steps to set up the basic Linux system.	Required	☐
3	Obtaining installation packages Describes the steps to download all required packages to install Parallel Sysplex.	Required	☐

4	• Updating the installed Parallel Sysplex.
	  Describes the steps to update Parallel Sysplex if you installed ZD&T V11.0 or later versions.
	
	• Uninstalling the earlier versions of ZD&T and Installing Parallel Sysplex.
	  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	☐

5	Enabling the license servers Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐
6	Activating ZD&T Parallel Sysplex instances Describes the steps to activate ZD&T instances	Required	☐

Updating Parallel Sysplex		Required/Optional	Complete
7	Configuring Z instances Describes the steps to configure ADCD to start Z instances.	Required	☐
8	Starting z/VM Describes the steps to start z/VM.	Required	☐
9	Starting Z instances Describes the steps to start Z instances.	Required	☐

Table 3. Update checklist for license server

Updating license server		Required/Optional	Complete
1	Returning a software-based license key or Returning a hardware-based license key Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	☐
2	Updating the software-based license server or Updating the hardware-based license server Describes the steps to update the software-based license server or hardware-based license server.	Required	☐
3	Enabling the license servers Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐

Table 4. Update checklist for license

Updating license		Required/Optional	Complete
1	Returning a software-based license key or Returning a hardware-based license key Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	☐
2	Enabling the license servers Describes the steps to install, enable, update or uninstall the software-based license server or hardware-based license server.	Required	☐

Table 5. Uninstallation checklist for Parallel Sysplex

Uninstalling Parallel Sysplex		Required/Optional	Complete
1	Returning a software-based license key or Returning a hardware-based license key Describes the steps to return a software-based or hardware-based license or Rational Tokens.	Required	☐
2	Uninstalling the software-based license server or Uninstalling the hardware-based license server Describes the steps to uninstall the software-based license server or hardware-based license server.	Required	☐
3	Uninstalling Parallel Sysplex Describes the steps to uninstall Parallel Sysplex.	Required	☐

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

  - Additional `YUM` commands on RHEL 8 operating system

    ```
    yum -y install ncurses-libs
    yum -y install libnsl
    ```

- **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 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 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

Learn about how to download installation packages.

Two types of packages are provided for installation: one for Extended ADCD, which can be downloaded optionally; one for the other components of ZD&T Parallel Sysplex. To download the installation packages, 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 then click Continue.
4. Enter the part description or part number, and then 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 lists in the following tables.

After a package is downloaded, you can verify the integrity of the downloaded package by using the .md5 checksum file. To do a checksum on a downloaded package, use command `md5sum -c <checksum_file_name>`.

Table 1. ZD&T Parallel Sysplex V14.0.0

	Name	Part No.	Image Classification	File Name
1	IBM Z Development and Test Environment 14.2.2 with Parallel Sysplex Installation eAssembly	G0BGDML	Required	Customer File Name
2	IBM Z Development and Test Environment Parallel Sysplex 14.2.2 Installation Multilingual	M0HGNML	Required	ZDT_Install_PS_V14.2.2.tgz
3	IBM Z Development and Test Environment Parallel Sysplex 14.2.2 Installation Checksum Multilingual	M0HGPML	Required	ps.md5
4	z/VM 7.3 Part 1 of 5 - RES volume Multilingual	M0HGQML	Required	M01RES.gz
5	z/VM 7.3 Part 2 of 5 – Page volume Multilingual	M0HGRML	Required	M01P01.gz
6	z/VM 7.3 Part 3 of 5 – System volume Multilingual	M0HGSML	Required	M01S01.gz
7	z/VM 7.3 Part 4 of 5 – Common volume Multilingual	M0HGTML	Required	VMCOM1.gz
8	z/VM 7.3 Part 5 of 5 – Release volume Multilingual	M0HGVML	Required	730RL1.gz
9	z/VM 7.3 - Checksum Multilingual	M0HGWML	Required	adcd_zvm73_checksum.txt
10	Devmap for z/VM 7.3 Multilingual	NA	Required	devmap.txt

	Name	Part No.	Image Classification	File Name
11	IBM Z Development and Test Environment 14.0 with Parallel Sysplex Installation eAssembly	G06Z7ML	Required	N/A
12	IBM Z Development and Test Environment Parallel Sysplex 14.0 Installation Multilingual	M083BML	Required	ZDT_Install_PS_V14.0.tgz
13	IBM Z Development and Test Environment Parallel Sysplex 14.0 Installation Checksum Multilingual	M083CML	Required	ps.md5
14	z/VM® 7.1 #1 of 5 - RES volume	G00PPZX	Required	M01RES.gz
15	z/VM 7.1 #2 of 5 – Page volume	G00PQZX	Required	M01P01.gz
16	z/VM 7.1 #3 of 5 – System volume	G00PRZX	Required	M01S01.gz
17	z/VM 7.1 #4 of 5 – Common volume	G00PSZX	Required	VMCOM1.gz
18	z/VM 7.1 #5 of 5 – Release volume	G00PTZX	Required	710RL1.gz
19	z/VM 7.1 - Checksum	G00PVZX	Required	adcd_zvm71_checksum.txt
20	IBM Common Licensing V9.0 Setup Multiplatform Multilingual	CC996ML	Optional	N/A
21	IBM License Key Server V9.0 for Windows Multiplatform Multilingual	CC997ML	Optional	N/A
22	IBM License Key Server V9.0 for Linux x86_64 Multiplatform Multilingual	CC998ML	Optional	N/A
23	IBM License Key Server V9.0 for AIX ppc64 Multiplatform Multilingual	CC999ML	Optional	N/A
24	IBM License Key Administration and Reporting Tool V9.0 Multiplatform Multilingual	CC99AML	Optional	N/A
25	IBM License Key Administration Agent V9.0 Multiplatform Multilingual	CC99BML	Optional	N/A
26	IBM Db2 Standard Edition - VPC Option - Activation 11.5 for Linux	CC36WML	Optional	N/A
27	IBM Db2 Server 11.5 for Linux on AMD64 and Intel EM64T system	CC1U0ML	Optional	N/A
28	IBM Db2 Server 11.5 for Linux on System z Multilingual	CC1U1ML	Optional	N/A
29	IBM Db2 Server 11.5 for Linux on POWER systems (little endian)	CC1U2ML	Optional	N/A
30	IBM Db2 Server 11.5 for AIX Multilingual	CC1U3ML	Optional	N/A
31	IBM Db2 Server 11.5 for Windows on AMD64 and Intel EM64T system	CC1U4ML	Optional	N/A

Table 2. Extended ADCD z/OS V3.1 December Edition of 2023, uploaded in March 2024

	Name	Part No.	Image Classification	File Name
1	ADCD 1Q24 z/OS V3.1 for IBM Z Development and Test 14.2.2 RSU 2309 Multilingual eAssembly	G0BGBML	Optional	N/A
2	ADCD 1Q24 z/OS 3.1 Part 1 of 20 - RES volume 1 Multilingual	M0HDDML	Optional	A3RES1.ZPD
3	ADCD 1Q24 z/OS 3.1 Part 2 of 20 - RES volume 2 Multilingual	M0HDFML	Optional	A3RES2.gz
4	ADCD 1Q24 z/OS 3.1 Part 3 of 20 - System volume Multilingual	M0HDGML	Optional	A3SYS1.gz
5	ADCD 1Q24 z/OS 3.1 Part 4 of 20 - Configuration volume Multilingual	M0HDHML	Optional	A3CFG1.gz
6	ADCD 1Q24 z/OS 3.1 Part 5 of 20 - UNIX System Services volume 1 Multilingual	M0HDJML	Optional	A3USS1.gz
7	ADCD 1Q24 z/OS 3.1 Part 6 of 20 - UNIX System Services volume 2 Multilingual	M0HDKML	Optional	A3USS2.gz
8	ADCD 1Q24 z/OS 3.1 Part 7 of 20 - UNIX System Services volume 3 Multilingual	M0HDLML	Optional	A3USS3.gz
9	ADCD 1Q24 z/OS 3.1 Part 8 of 20 - z/OS Products volume 1 Multilingual	M0HDMML	Optional	A3PRD1.gz
10	ADCD 1Q24 z/OS 3.1 Part 9 of 20 - z/OS Products volume 2 Multilingual	M0HDNML	Optional	A3PRD2.gz
11	ADCD 1Q24 z/OS 3.1 Part 10 of 20 - z/OS Products volume 3 Multilingual	M0HDPML	Optional	A3PRD3.gz
12	ADCD 1Q24 z/OS 3.1 Part 11 of 20 - z/OS Products volume 4 Multilingual	M0HDQML	Optional	A3PRD4.gz
13	ADCD 1Q24 z/OS 3.1 Part 12 of 20 - z/OS Products volume 5 Multilingual	M0HDRML	Optional	A3PRD5.gz
14	ADCD 1Q24 z/OS 3.1 Part 13 of 20 - z/OS Distribution Libraries volume 1 Multilingual	M0HDSML	Optional	A3DIS1.gz
15	ADCD 1Q24 z/OS 3.1 Part 14 of 20 - z/OS Distribution Libraries volume 2 Multilingual	M0HDTML	Optional	A3DIS2.gz
16	ADCD 1Q24 z/OS 3.1 Part 15 of 20 - z/OS Distribution Volume Multilingual	M0HDVML	Optional	A3DIS3.gz
17	ADCD 1Q24 z/OS 3.1 Part 16 of 20 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M0HDWML	Optional	A3PAGA.gz
18	ADCD 1Q24 z/OS 3.1 Part 17 of 20 - z/OS Page Datasets (Local) volume 2 Multilingual	M0HDXML	Optional	A3PAGB.gz
19	ADCD 1Q24 z/OS 3.1 Part 18 of 20 - z/OS Page Datasets (Local) volume 3 Multilingual	M0HDYML	Optional	A3PAGC.gz
20	ADCD 1Q24 z/OS 3.1 Part 19 of 20 - z/OS User volume Multilingual	M0HDZML	Optional	A3USR1.gz
21	ADCD 1Q24 z/OS 3.1 Part 20 of 20 - Stand Alone RES volume Multilingual	M0HF0ML	Optional	SARES1.ZPD
22	ADCD 1Q24 Engineering Workflow Management 7.0.2 Multilingual	M0HF1ML	Optional	A3BLZ1.gz
23	ADCD 1Q24 DB2 V12 Part 1 of 2 Multilingual	M0HF2ML	Optional	A3DBC1.gz
24	ADCD 1Q24 DB2 V12 Part 2 of 2 Multilingual	M0HF3ML	Optional	A3DBC2.gz
25	ADCD 1Q24 DB2 V13 Part 1 of 2 Multilingual	M0HF4ML	Optional	A3DBD1.gz
26	ADCD 1Q24 DB2 V13 Part 2 of 2 Multilingual	M0HF5ML	Optional	A3DBD2.gz
27	ADCD 1Q24 CICS 5.6 Multilingual	M0HF6ML	Optional	A3C560.gz
28	ADCD 1Q24 CICS 6.1 Multilingual	M0HF7ML	Optional	A3C610.gz
29	ADCD 1Q24 IMS 15.2 Multilingual	M0HF8ML	Optional	A3IMF1.gz
30	ADCD 1Q24 IBM Installation Manager 1.9 Multilingual	M0HF9ML	Optional	A3INM1.gz
31	ADCD 1Q24 DB2 Archive Logs Multilingual	M0HFBML	Optional	A3DBAR.gz
32	ADCD 1Q24 IBM Z Monitoring Suite v1.2	M0HFCML	Optional	A3KAN1.gz
33	ADCD 1Q24 WAS V9.0 Part 1 of 2 Multilingual	M0HFDML	Optional	A3W901.gz
34	ADCD 1Q24 WAS V9.0 Part 2 of 2 Multilingual	M0HFFML	Optional	A3W902.gz
35	ADCD 1Q24 Zowe Open Source Project Multilingual	M0HFGML	Optional	A3ZWE1.gz
36	ADCD Metadata for Tools Multilingual	M0HFHML	Optional	ADCDTOOLS.XML
37	ADCD 1Q24 Inventory for Sandbox Multilingual	M0HFJML	Optional	inventory.txt

	Name	Part No.	Image Classification	File Name
38	ADCD 1Q24 Devmap for Sandbox Multilingual	M0HFKML	Optional	devmap.txt
39	ADCD 1Q24 Extended ADCD Notices Multilingual	M0HFLML	Optional	notices_ADCD_RSU 2309
40	ADCD 1Q24 Inventory with SHA256 Checksums for Sandbox Multilingual	M0HFMML	Optional	inventory_sha256.txt
41	ADCD 1Q24 IPL Configuration Script Multilingual	M0HFNML	Optional	zdt_config_adcd_ipl.sh
42	ADCD z/OS V2.5 for IBM Z Development and Test 14.2 RSU 2303 Multilingual eAssembly	G08N2ML	Optional	N/A
43	ADCD 3Q'23: z/OS 2.5 Part 1 of 20 - RES volume 1 Multilingual	M0FDQML	Optional	D5RES1.ZPD
44	ADCD 3Q'23: z/OS 2.5 Part 2 of 20 - RES volume 2 Multilingual	M0FDRML	Optional	D5RES2.gz
45	ADCD 3Q'23: z/OS 2.5 Part 3 of 20 - System volume Multilingual	M0FDSML	Optional	D5SYS1.gz
46	ADCD 3Q'23: z/OS 2.5 Part 4 of 20 - Configuration volume Multilingual	M0FDTML	Optional	D5CFG1.gz
47	ADCD 3Q'23: z/OS 2.5 Part 5 of 20 - UNIX System Services volume 1 Multilingual	M0FDVML	Optional	D5USS1.gz
48	ADCD 3Q'23: z/OS 2.5 Part 6 of 20 - UNIX System Services volume 2 Multilingual	M0FDWML	Optional	D5USS2.gz
49	ADCD 3Q'23: z/OS 2.5 Part 7 of 20 - UNIX System Services volume 3 Multilingual	M0FDXML	Optional	D5USS3.gz
50	ADCD 3Q'23: z/OS 2.5 Part 8 of 20 - z/OS Products volume 1 Multilingual	M0FDYML	Optional	D5PRD1.gz
51	ADCD 3Q'23: z/OS 2.5 Part 9 of 20 - z/OS Products volume 2 Multilingual	M0FDZML	Optional	D5PRD2.gz
52	ADCD 3Q'23: z/OS 2.5 Part 10 of 20 - z/OS Products volume 3 Multilingual	M0FF0ML	Optional	D5PRD3.gz
53	ADCD 3Q'23: z/OS 2.5 Part 11 of 20 - z/OS Products volume 4 Multilingual	M0FF1ML	Optional	D5PRD4.gz
54	ADCD 3Q'23: z/OS 2.5 Part 12 of 20 - z/OS Products volume 5 Multilingual	M0FF2ML	Optional	D5PRD5.gz
55	ADCD 3Q'23: z/OS 2.5 Part 13 of 20 - z/OS Distribution Libraries volume 1 Multilingual	M0FF3ML	Optional	D5DIS1.gz
56	ADCD 3Q'23: z/OS 2.5 Part 14 of 20 - z/OS Distribution Libraries volume 2 Multilingual	M0FF4ML	Optional	D5DIS2.gz
57	ADCD 3Q'23: z/OS 2.5 Part 15 of 20 - z/OS Distribution Volume Multilingual	M0FF5ML	Optional	D5DIS3.gz
58	ADCD 3Q'23: z/OS 2.5 Part 16 of 20 - z/OS Page Datasets (PLPA, Common, Local) volume 1 Multilingual	M0FF6ML	Optional	D5PAGA.gz
59	ADCD 3Q'23: z/OS 2.5 Part 17 of 20 - z/OS Page Datasets (Local) volume 2 Multilingual	M0FF7ML	Optional	D5PAGB.gz
60	ADCD 3Q'23: z/OS 2.5 Part 18 of 20 - z/OS Page Datasets (Local) volume 3 Multilingual	M0FF8ML	Optional	D5PAGC.gz
61	ADCD 3Q'23: z/OS 2.5 Part 19 of 20 - z/OS User volume Multilingual	M0FF9ML	Optional	D5USR1.gz
62	ADCD 3Q'23: z/OS 2.5 Part 20 of 20 - Stand Alone RES volume Multilingual	M0FFBML	Optional	SARES1.ZPD
63	ADCD 3Q'23: Engineering Workflow Management 7.0.2 Multilingual	M0FFCML	Optional	D5BLZ1.gz
64	ADCD 3Q'23: DB2 V12 Part 1 of 2 Multilingual	M0FFDML	Optional	D5DBC1.gz
65	ADCD 3Q'23: DB2 V12 Part 2 of 2 Multilingual	M0FFFML	Optional	D5DBC2.gz
66	ADCD 3Q'23: DB2 V13 Part 1 of 2 Multilingual	M0FFGML	Optional	D5DBD1.gz
67	ADCD 3Q'23: DB2 V13 Part 2 of 2 Multilingual	M0FFHML	Optional	D5DBD2.gz
68	ADCD 3Q'23: CICS 5.6 Multilingual	M0FFJML	Optional	D5C560.gz
69	ADCD 3Q'23: CICS 6.1 Multilingual	M0FFKML	Optional	D5C610.gz
70	ADCD 3Q'23: IMS 15.2 Multilingual	M0FFLML	Optional	D5IMF1.gz
71	ADCD 3Q'23: IBM Installation Manager 1.9 Multilingual	M0FFMML	Optional	D5INM1.gz
72	ADCD 3Q'23: DB2 Archive Logs Multilingual	M0FFNML	Optional	D5DBAR.gz
73	ADCD 3Q'23: IBM Z Monitoring Suite v1.2	M0FFPML	Optional	D5KAN1.gz
74	ADCD 3Q'23: WAS V9.0 Part 1 of 2 Multilingual	M0FFQML	Optional	D5W901.gz
75	ADCD 3Q'23: WAS V9.0 Part 2 of 2 Multilingual	M0FFRML	Optional	D5W902.gz
76	ADCD 3Q'23: Zowe Open Source Project Multilingual	M0FFSML	Optional	D5ZWE1.gz
77	ADCD Metadata for Tools Multilingual	M0FFTML	Optional	ADCDTOOLS.XML
78	ADCD 3Q'23 Inventory for Sandbox Multilingual	M0FFVML	Optional	inventory.txt
79	ADCD 3Q'23 Devmap for Sandbox 1.4 Linux Multilingual	M0G01ML	Optional	devmap.txt
80	ADCD 3Q'23 Extended ADCD Notices Multilingual	M0FFXML	Optional	notices_ADCD_RSU 2303
81	ADCD 3Q'23 Inventory with SHA256 Checksums for Sandbox Multilingual	M0G7CML	Optional	inventory_sha256.txt
82	ADCD 3Q'23 IPL Configuration Script Multilingual	M0FFZML	Optional	zdt_config_adcd_ipl.sh

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.

The ZDT_Install_PS_V14.2.2.tgz files that you download from IBM are compressed installation packages. Each ZD&T Parallel Sysplex Edition installation package contains the following signed installers and the corresponding signature files.

Installer	Signature file
zdt-install-ps	zdt-install-ps.cosign.sig

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

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

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

```
openssl dgst -sha256 -verify signature/public.PSm -signature zdt-install-ps.cosign.sig zdt-install-ps.cosign.sig
```

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.

Table 1. Enablement checklist for software-based licensing

Software-based license	Required/Optional	Complete
1 Installing the software-based license server Describes the steps to install the software-based license server	Required	☐
2 Obtaining an update file Describes the steps to obtain an update file for software license server.	Required	☐
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	☐

Table 2. Enablement checklist for hardware-based licensing

Hardware-based license	Required/Optional	Complete
1 Installing the hardware-based license server Describes the steps to install the hardware-based license server.	Required	☐
2 Obtaining an update file for hardware-based license server Describes the steps to obtain an update file for hardware-based license server.	Required	☐
3 Applying the update file to the hardware-based license server Describes the steps to apply the update file to ZD&T Parallel Sysplex.	Required	☐

Table 3. Switching from Hardware-based License Server to Software-based License Server

Hardware-based license to Software-based license	Required/Optional	Complete
1 Returning a hardware-based license key Describes the steps to return a hardware-based license key.	Required	☐
2 Uninstalling the hardware-based license server Describes the steps to uninstall hardware-based license server.	Required	☐
3 Installing the software-based license server Describes the steps to install software-based license server.	Required	☐
4 Obtaining an update file for software-based license server Describes the steps to obtain an update file for software-based license server.	Required	☐
5 Applying the update file to the software-based license server Describes the steps to apply the update file to software-based license server.	Required	☐

- **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.
- **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.

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 IBM 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 LKC, is a generated unique identifier for each software-based license serverlicense 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 LKC, 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 IBM 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.
- **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.
- **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.
- **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.
- **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.
- **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.

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_V14.0.0.tgz.
2. Change the authority.

    ```
    chmod 755 ZDT_Install_PS_V14.0.0.tgz
    ```

3. Decompress the installer.

    ```
    tar -xzvf ZDT_Install_PS_V14.0.0.tgz
    ```

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

    ```
    ./ZDT_Install_PS_V14.0.0.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. You need to have access to internet and software repository to install the dependencies. Otherwise, installation will complete without dependencies, and you need to install the dependencies manually.

11. After the installation completes, run the following commands to verify whether the installation is successful.
    - RHEL operating system

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

    - Ubuntu operating system

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

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

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

    To delete the user ID, use the following command:

    ```
    userdel -r ibmsys2
    ```

Alternatively, after you complete steps 1 - 4, you can install the software-based license server silently.

- To install the software-based license server on the virtual machine, run the following command:

    ```
    ./ZDT_Install_PS_V14.0.0.x86_64 --install --zdtswlicense --ackVM
    ```

- To install the dependencies, run the following command:

    ```
    ./ZDT_Install_PS_V14.0.0.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.
- **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.

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 (LKC), and select your account. If you do not have access to LKC, 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 IBM 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 IBM License Key Center.
3. Log in to Rational License Key Center, and select your account. If you do not have access to LKC, 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 IBM 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 IBM 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. 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


Thesse are the currently active sessions:

       KeyID         FID      FeatureName        Address       User          Machine      LoginTime         #
 432975633343422885 333       CPU License    9.26.158.161    ibmsys1       zdt-dev4  Mon Oct 30, 17:25:40    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 IBM License Key Center.
- **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.
- **Returning Rational Tokens for software-based license server**
  To generate a new update file, you must first return the existing license entitlement in the IBM License Key Center, and then generate a new update file. Returning the license entitlements is a process of telling the IBM 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 IBM License Key Center.

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 IBM License Key Center.

You can return a license entitlement in the IBM 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 IBM License Key Center.
3. Log in to the IBM 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 IBM 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 IBM License Key Center.

You can return a license entitlement in the IBM 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 more 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 IBM License Key Center.
2. Obtain a license key file that invalidates the current license key from the IBM License Key Center.
   a. Log in to the IBM 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 IBM 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 IBM 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 IBM License Key Center.
4. Use the verification file to complete the return process in the IBM License Key Center.
   a. Log in to the IBM 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 IBM License Key Center, and then generate a new update file. Returning the license entitlements is a process of telling the IBM 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 IBM License Key Center.

1. Log in to IBM 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.

To update the software-based license server, complete the following steps:

1. Open the directory that contains the installer ZDT_Install_PS_V14.0.0.tgz.
2. Change the authority.

   chmod 755 ZDT_Install_PS_V14.0.0.tgz

3. Decompress the installer.

   tar -xzvf ZDT_Install_PS_V14.0.0.tgz

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

   ./ZDT_Install_PS_V14.0.0.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_V14.0.0.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.

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.
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_V14.0.0.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_V14.0.0.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.
In figure 1, the entire serial number is 02-00222.

Capacity of the hardware device

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

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

- **Low capacity devices**
  Low Capacity USB hardware devices can activate up to three CPs. These devices have a blue label. As of ZD&T v8.5, low capacity devices are no longer available for order.

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

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.
- **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.
- **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.
- **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.
- **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.
- **Uninstalling the hardware-based license server**
  To uninstall the hardware-based license server, you need to 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.

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_V14.0.0.tgz.
2. Change the authority.

   ```
   chmod 755 ZDT_Install_PS_V14.0.0.tgz
   ```

3. Decompress the installer.

   ```
   tar -xzvf ZDT_Install_PS_V14.0.0.tgz
   ```

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

```
./ZDT_Install_PS_V14.0.0.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](#). 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_V14.0.0.x86_64 --install --zdthwlicense
```

To install the dependencies silently, run the following command:

```
./ZDT_Install_PS_V14.0.0.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.
- **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.

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](#) (LKC), and select your account. If you do not have access to LKC, 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 IBM 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 (LKC), and select your account. If you do not have access to LKC, 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.
- **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.

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 IBM 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 IBM License Key Center (LKC), 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 IBM License Key Center, and then generate a new update file. Returning the license entitlements is a process of telling the IBM 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 IBM License Key Center.

1. Log in to IBM License Key Center (LKC), 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.

To update the hardware-based license server, complete the following steps:

1. Open the directory that contains the installer ZDT_Install_PS_V14.0.0.tgz.
2. Change the authority.

   chmod 755 ZDT_Install_PS_V14.0.0.tgz

3. Decompress the installer.

   tar -xzvf ZDT_Install_PS_V14.0.0.tgz

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

   ./ZDT_Install_PS_V14.0.0.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_V14.0.0.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 <u>Enabling the license servers</u>.

Uninstalling the hardware-based license server

To uninstall the hardware-based license server, you need to 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 <u>Returning a hardware-based license key</u>.
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_V14.0.0.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_V14.0.0.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_V14.0.0.tgz.
2. Change the authority.

   `chmod 755 ZDT_Install_PS_V14.0.0.tgz`

3. Decompress the installer.

   `tar -xzvf ZDT_Install_PS_V14.0.0.tgz`

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

   `./ZDT_Install_PS_V14.0.0.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 <u>Prerequisites</u>. 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_V14.0.0.x86_64 --install --zdtps
```

To install the dependencies, run the following command:

```
./ZDT_Install_PS_V14.0.0.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.
- **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.

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.
- **Creating the device map**
  A sample program `create_devmap.pl` is available in the `/opt/ConfigGuideSample` directory where you have installed ZD&T.

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.
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>/A3RES1.ZPD <output path of volume>/A3RES1
```

For example, `./Z1091_ADCD_install /home/ibmsys1/volumes/A3RES1.ZPD /home/ibmsys1/volumes/A3RES1`.

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.

The sample program only creates a device map for the z/OS disk volumes. You must append the z/VM disk volumes to the device map manually. There is a sample device map for z/VM shipped with the z/VM package.

After generating the device map for the z/OS Volumes you can add the following devices to the awsckd stanza:

```
device 0123 3390 3390 M01RES
```

```
device 0124 3390 3390 VMCOM1
```

```
device 0125 3390 3390 730RL1
```

```
device 0126 3390 3390 M01S01
```

```
device 0127 3390 3390 M01P01
```

To create the device map from a non-root user ID, run the following command:

```
perl <pathtocommand>/create_devmap.pl pathtodisks > 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](#).

Starting z/VM

To start z/VM®, issue `awsstart` under `ibmsys1` user ID or equivalent home by using directory structure /usr/ibmsys1.

Restriction: You cannot use the `runzpdt` script when you perform a sysplex activation.
To start z/VM, complete the following steps:

1. Switch to a non-root user ID.
2. Switch to the path where you defined your devmap. For example, if your devmap is located in /usr/ibmsys1/volumes, enter the following command:

```
cd /usr/ibmsys1/volumes
```

3. Enter the following command.

```
awsstart generateddevmap
```

```
generated
devmap
```
is the name of your devmap.
4. This is an optional step and applicable only if you are using 3270 console when ipling z/VM. The ipl command given in step 6 uses parm conssysc which does not require the terminal session to be open.

Open the terminal emulator, and enter the IP and port (3270) of the machine where you have installed Parallel Sysplex® to configure the emulator. If you use Linux® 3270, use the following command to start.

```
x3270 -port 3270 <session_name_in_devmap>@localhost &
```

5. Open the devmap, and find the address of system residence volume for z/VM (M01RES).
6. IPL the z/VM system by entering the following command, where address **123** is the device address in the devmap of the system residence volume for z/VM (M01RES).

```
ipl 123 parm conssysc
```

Note that using parm conssysc uses the Linux host as the master console and eliminates the need to logon to 3270 device and load z/VM manually. Logon to device 700 via 3270 emulation and use the appropriate parm if you are using the 3270 console interface.

7. After the IPL completion, from the Linux host where you issued awsstart and ipl command, enter the following command:

```
oprmsg 'enable all'
```

8. To start the Coupling Facility, from the z/VM logon screen, log in with user CFCONSOL and password ZVM710.
   After logging in successfully enter:

```
xautolog cfcc1
```

Messages resembling the following appears:

```
CFCC1 : 14:16:34 CF0009I IBM Confidential
CFCC1 : Licensed Internal Code - Property of IBM
CFCC1 : 3931, 3932 Licensed Internal Code
CFCC1 : Coupling facility control code
CFCC1 : (C) Copyright IBM Corp. 1993, 2022
CFCC1 : All rights reserved.
CFCC1 : US Government Users Restricted Rights -
CFCC1 : Use, duplication or disclosure restricted
CFCC1 : by GSA ADP Schedule Contract with IBM Corp.
CFCC1 : 14:16:34 CF0280I CFCC Release 25.00, Service Level 02.43
CFCC1 : Built on 05/17/2022 at 15:26:00
CFCC1 : Code Load Features:
CFCC1 : Facility Operational Level: 25
CFCC1 : 14:16:34 CF0011I Coupling Facility is active with:
CFCC1 : 1 CP
CFCC1 : 0 CF Receiver Channels
CFCC1 : 0 CF Sender Channels
CFCC1 : 4652 MB of allocatable storage
CFCC1 : 0 MB of Total SCM storage
```

9. Optionally start a second coupling facility using command:

```
xautolog cfcc2
```

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

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

Starting Z instances

You need to start Z instances from an active 3270 session that is connected to z/VM®.

To start Z instances, complete the following steps:

1. When the z/VM screen is displayed, enter a valid z/VM user ID and password. For example, use `BASEAD` as the user ID and `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.Z25C.PARMLIB(SHUT**).

To shut down Z instances, complete the following steps:

1. Enter `S SHUTXX` in the master console.
   XX is the last 2 characters of shutdown script that is generally equivalent to loadparms. To reply to any pending message, enter this command from the master console. If you do not have a master console, enter `/S SHUTXX` in SDSF, or enter `oprmsg "S SHUTXX"` in the Linux® machine.

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

Note: Do not make any change in the ADCD.Z31A.* library. If you want to edit any parm or proc, ADCD has USER.Z31A.* 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](#).
- For the steps to return the hardware-based license, see [Returning a hardware-based license key](#).

- **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..
- **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.

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	• Stop any currently active instance of RD&T.
	   ○ To stop a base instance of RD&T, see [Stopping Z instances](#).
	   ○ If you are running a product license server:
	     1. Stop the UIM server by using the command from a non-root user ID.
	
	     `cd /usr/z1090/bin`
	     `./uimserverstop`
	
	     2. Enter these two commands to stop the product license server:
	
	     `# cd /opt/safenet_sentinel/common_files/sentinel_keys_server`
	     `# ./loadserv stop`
	
	• 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	• 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	• 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.

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.

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_V14.0.0.tgz.
2. Change the authority.

   chmod 755 ZDT_Install_PS_V14.0.0.tgz

3. Decompress the installer.

   tar -xzvf ZDT_Install_PS_V14.0.0.tgz

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

   ./ZDT_Install_PS_V14.0.0.x86_64

6. Select **2** for ZD&T Parallel Sysplex.
7. Select **2** to update.
8. Press ENTER, and read the license agreements carefully. At the end of the license, enter `yes` to accept or `no` to decline the terms.
9. After the update completes, run the following command to verify whether the update is successful.
   - RHEL operating system

     rpm -qa | grep z1091

   - Ubuntu operating system

     dpkg -l | grep z1091

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

   z1091-1-10.55.12.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_V14.0.0.x86_64 --update --zdtps

Uninstalling Parallel Sysplex

To uninstall Parallel Sysplex®, you need to return the license key, and uninstall the license server first. Then, run the installer from the root user ID.

To uninstall ZD&T Parallel Sysplex, complete the following steps:

1. Stop the z/VM® and Z instances. For more information, see Stopping Z instances and z/VM
2. Run the installer from the root user ID.

   ./ZDT_Install_PS_V14.0.0.x86_64

3. Select **2** for ZD&T Parallel Sysplex.
4. Select **3** to uninstall.

Alternatively, you can run uninstall ZD&T Parallel Sysplex silently by using the following command.

./ZDT_Install_PS_V14.0.0.x86_64 --uninstall --zdtps

Configuring Extended ADCD

Learn how to configure Extended ADCD.

Before you begin

- You must have a storage server set up according to the storage server prerequisites. The storage server will be used to store Extended ADCD files.
- Make sure that the web server is installed, and it is connected to the storage server through the web server UI. For instructions, see section Setting up the web server and topic Connecting to your storage server.
- Download the Extended ADCD package according to instructions in topic Downloading installation packages.

About this task

Application Developers Controlled Distribution (ADCD) is a customized bundle of z/OS and related products. Extended ADCD is a pre-built package that contains a set of mainframe software, such as CICS®, Db2®, IMS, z/OS 2.5. Also, Extended ADCD includes host components that are needed by Wazi Code, such as Git, z/OS Explorer, RSE API, IBM® z/OS Debugger, Dependency Based Build, ZOA Utilities, and z/OS Explorer Extensions. Developers and testers can download Extended ADCD, develop and test their applications with these software packages without any installation. The Extended ADCD package is updated approximately twice a year and is provided as a part of the product. To use Extended ADCD for image creation and z/OS provisioning, follow the steps to transfer the Extended ADCD files to your storage server.

Procedure

1. Confirm your storage server base directory from the storage server settings page on the web server UI. In the base directory, create a directory labeled adcd, if it does not already exist.
2. In the adcd directory created in the previous step, create a directory with a unique label for each Extended ADCD version you want to make available in the Enterprise Edition, for example, adcd/*<version>*.
3. Transfer the following files from the downloaded Extended ADCD package to the directory specified in the previous step:

   ADCDTOOLS.XML
   > This file describes the metadata for the Enterprise Edition.
   zdt_config_adcd_ipl.sh
   > This file is used to customize z/OS based on your selections. For example, if you prefer an Extended ADCD image with only CICS, this script adjusts z/OS configurations accordingly. Similarly, if you choose different software or combinations of software for an Extended ADCD image, this script updates z/OS configurations to start only those subsystems after IPL.

4. Create a volumes directory in the directory specified in step 2, for example, adcd/*<version>*/volumes. Then transfer all the volume files from the downloaded Extended ADCD package to this volumes directory.
5. If you want to customize Extended ADCD with extra configurations, to make upgrading easier, see the following tips:
   - Script all customization that is done to Extended ADCD, customize it by batch, or combine the two methods, so that this process can be easily repeatable.
   - When possible, keep all customization separate from z/OS. For example, keep it on your own disk volumes instead of the Extended-ADCD-supplied volumes.

What to do next

After configuring Extended ADCD, you can create images from it by adding an ADCD source environment on the web server UI. For instructions, see topic Adding a source environment from Extended ADCD.

- **Extended ADCD reference**
  Learn about general release information of ADCD z/OS V3R1 December Edition of 2023, hints and tips, and the customized ADCD package (Extended ADCD) for ZD&T.

Extended ADCD reference

Learn about general release information of ADCD z/OS® V3R1 December Edition of 2023, hints and tips, and the customized ADCD package (Extended ADCD) for ZD&T.

- **ADCD z/OS V3R1 December Edition of 2023**
  Learn what's new in ADCD z/OS V3R1 December Edition of 2023 and the products that are contained in this release.
- **Extended ADCD for ZD&T**
  Extended ADCD is built upon the general release of ADCD z/OS V2R5 May Edition of 2023. Extended ADCD contains some changes to support IBM Cloud Pak® for Applications, and improve the z/OS system start performance for Java™ heavy subsystems.

ADCD z/OS V3R1 December Edition of 2023

Learn what's new in ADCD z/OS V3R1 December Edition of 2023 and the products that are contained in this release.

Note: The SMS section on this page for ADCD is replaced by the SMS that is provided for Extended ADCD. For more information, see Customized SMS configuration in Extended ADCD.

- What's New in ADCD z/OS V3R1 December Edition of 2023
- Products contained in this release
- Reported Problems, Fixes, Maintenance and Observations
- Build Structure
- Migration Guidelines
- LoadParm Options
- Console PF Key Settings
- USERIDS
- Maintenance Service Levels

What's New in ADCD z/OS V3R1 December Edition of 2023

Note: As of ADCD z/OS 3.1 December 2023 Edition, distribution via DVDs has been discontinued and all future distribution of ADCD Editions will only be available via download.
The ADCD Release Guide z/OS 3.1 December 2023 Edition, also available on the download site, contains more information about this edition. It also provides some additional notes and tips to effectively manage the ADCD system.

The following Workflow Provisions have been added to the Market Place:

- Db2 v12 Full Instance
- CICS v6.1

The following products have been added:

- IBM Z Development & Test Environment Enterprise Edition V14.0.0
- IBM Open Enterprise SDK for GO V1.2.1
- IBM Z Multi-Factor Authentication V2.2.0
- IBM Application Delivery Foundation for z/OS V4.0.2
- IBM z/OS Connect Enterprise Edition V3.0.0
- IBM Open Enterprise SDK for Python V3.11.0
- IBM Urban Code Deploy for z/OS V7.3.2

The following products have been updated to PUT2309 / RSU2309

- z/OS 3.1 and all incorporated products that provided maintenance at the above stated level.

All of the z/OS base, z/OS products, and most middleware volume sizes have been increased to mod-9 (10,017 cylinders) to provide more free space for expansion. Additionally, volumes for Db2 V13 and Installation Manager have been increased to 18,000 and 20,000 cylinders respectively.

Products contained in this release

Table 1. **z/OS 3.1 December 2023 Edition**

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-ZOS	Hardware Configuration Definition Base	HCS77E0	HCS77E0	SYS1
5655-ZOS	GDDM PGF	HGD3201	HGD3201	SYS1
5655-ZOS	XL C/C++ Base	HLB77C0	HLB77C0	SYS1
5655-ZOS	XL C/C++ Base	HLB77C1	HLB77C1	SYS1
5655-ZOS	IBM Open XL C/C++	HLB77C2	HLB77C2	SYS1
5655-ZOS	Infoprint Server - IP PrintWay Base	HMOS705	HMOS705	SYS1
5655-ZOS	Infoprint Server - NetSpool Base	HNET7D0	HNET7D0	SYS1
5655-ZOS	Infoprint Server - Print Interface Base	HOPI7D0	HOPI7D0	SYS1
5655-ZOS	SDSF Base	HQX77E0	HQX77E0	SYS1
5655-ZOS	Security Server - RACF Base	HRF77E0	HRF77E0	SYS1
5655-ZOS	RMF	HRM77E0	HRM77E0	SYS1
5655-ZOS	DFSORT Base	HSM1310	HSM1310	SYS1
5655-ZOS	C/C++ HOST PERFORMANCE ANALYZE	H24P111	H24P111	SYS1
5655-ZOS	High Level Assembler Toolkit	HMQ4160	JMQ416A	SYS1
5655-ZOS	XL C/C++ JPN	HLB77C0	JLB77CJ	SYS1
5655-ZOS	Infoprint Server - IP PrintWay JPN	HMOS705	JMOS7J5	SYS1
5655-ZOS	Infoprint Server - NetSpool JPN	HNET7D0	JNET7DJ	SYS1
5655-ZOS	Infoprint Server - Print Interface JPN	HOPI7D0	JOPI7DJ	SYS1
5655-ZOS	Security Server - RACF JPN	HRF77E0	JRF77EJ	SYS1
5655-ZOS	RMF Japanese	HRM77E0	JRM77EJ	SYS1
5655-ZOS	C/C++ HOST PERFORMANCE ANALYZE	H24P111	J24P112	SYS1
5655-HZ1	IBM Container Hosting Foundation for z/OS	HHZ1100	HHZ1100	SYS1
5655-P97	Encryption Facility DFSMSdss Encryption	HCF773D	HCF773D	SYS1
5655-P97	Encryption Facility Encrypt Ser	HCF7740	HCF7740	SYS1
5655-TF1	Print Transform Accessible PDF	HTFA124	HTFA124	SYS1
5655-TF1	IBM Signed Secure PDF for z/OS	HTFS125	HTFS125	SYS1
5655-TF1	Print Transform Common	HTFX120	HTFX120	SYS1
5655-TF1	Print Transform AFPxPDF	HTFX121	HTFX121	SYS1
5655-NOJ	IBM Open Enterprise SDK for Node.js	HAMIH00	HAMIH00	SYS1
5655-ZCX	IBM zCX Foundation for Red Hat OpenShift	HHRH110	HHRH110	SYS1
5655-ZOS	ICKDSF - Device Support Facilities, Base	EDU1H01	EDU1H01	SYS1
5655-ZOS	Environmental Record Editing and Printing	EER3500	EER3500	SYS1
5655-ZOS	MICR/OCR	EMI2220	EMI2220	SYS1
5655-ZOS	TIOC	ETI1106	ETI1106	SYS1
5655-ZOS	ICKDSF - Device Support Facilities, ISMF/MODS	EDU1H01	FDU1H07	SYS1
5655-ZOS	ICKDSF - Device Support Facilities, ISMF/ENU	EDU1H01	FDU1H08	SYS1
5655-ZOS	BCP Base	HBB77E0	HBB77E0	SYS1
5655-ZOS	Hardware Configuration Manager	HCM1J10	HCM1J10	SYS1
5655-ZOS	Cryptographic Services - System SSL Base	HCPT510	HCPT510	SYS1
5655-ZOS	Cryptographic Support - ICSF	HCR77E0	HCR77E0	SYS1
5655-ZOS	Network File System Server and Client	HDZ331N	HDZ331N	SYS1
5655-ZOS	Data Facility System Managed Storage Base & ENU	HDZ3310	HDZ3310	SYS1
5655-ZOS	z/OS Font Collection	HFNT140	HFNT140	SYS1
5655-ZOS	FFST	HFST101	HFST101	SYS1
5655-ZOS	PC 3270 FILE TRANSFER	HFX1112	HFX1112	SYS1
5655-ZOS	GDDM BASE	HGD3200	HGD3200	SYS1
5655-ZOS	IBM HTTP Server	HHAP90P	HHAP90P	SYS1
5655-ZOS	ISPF Base	HIF83A2	HIF83A2	SYS1

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-ZOS	IOCP	HIO1106	HIO1106	SYS1
5655-ZOS	Communications Server IP	HIP6310	HIP6310	SYS1
5655-ZOS	JES2 BASE	HJE77E0	HJE77E0	SYS1
5655-ZOS	HKCZ300	HKCZ300	HKCZ300	SYS1
5655-ZOS	Cryptographic Services - PKI Services	HKY77E0	HKY77E0	SYS1
5655-ZOS	Language Environment Base	HLE77E0	HLE77E0	SYS1
5655-ZOS	SMP/E Base	HMP1K00	HMP1K00	SYS1
5655-ZOS	High Level Assembler Base	HMQ4160	HMQ4160	SYS1
5655-ZOS	OpenSSH for z/OS	HOS3310	HOS3310	SYS1
5655-ZOS	z/OS UNIX System Services Application Services Bas	HOT77E0	HOT77E0	SYS1
5655-ZOS	CIM - Common Interface Model	HPG77E0	HPG77E0	SYS1
5655-ZOS	BCP - Program Management Binder	HPM77E0	HPM77E0	SYS1
5655-ZOS	BCP - Capacity Provisioning	HPV77E0	HPV77E0	SYS1
5655-ZOS	zOS Data Gatherer	HRG77E0	HRG77E0	SYS1
5655-ZOS	IBM Tivoli Directory Server for z/OS Base	HRSL510	HRSL510	SYS1
5655-ZOS	Metal C Runtime Library	HSD7780	HSD7780	SYS1
5655-ZOS	Z/OSMF Network Configuration Assistant	HSMA31A	HSMA31A	SYS1
5655-ZOS	z/OSMF zERT Network Analyzer	HSMA31E	HSMA31E	SYS1
5655-ZOS	z/OSMF Core Functions	HSMA310	HSMA310	SYS1
5655-ZOS	z/OSMF ISPF	HSMA311	HSMA311	SYS1
5655-ZOS	z/OSMF Resource Monitoring	HSMA312	HSMA312	SYS1
5655-ZOS	zOSMF WLM	HSMA313	HSMA313	SYS1
5655-ZOS	z/OSMF Software Deployment	HSMA314	HSMA314	SYS1
5655-ZOS	z/OSMF Incident Log	HSMA315	HSMA315	SYS1
5655-ZOS	z/OSMF Capacity Provisioning	HSMA316	HSMA316	SYS1
5655-ZOS	z/OSMF Workflow	HSMA317	HSMA317	SYS1
5655-ZOS	ESCON DIRECTOR	HSWF100	HSWF100	SYS1
5655-ZOS	Integrated Security Services-Network Authenticatio	HSWK510	HSWK510	SYS1
5655-ZOS	TSO/E Base	HTE77E0	HTE77E0	SYS1
5655-ZOS	Runtime Library Extensions Base	HTV77C0	HTV77C0	SYS1
5655-ZOS	BCP - Support for Unicode Base	HUN77E0	HUN77E0	SYS1
5655-ZOS	Communications Server SNA	HVT6310	HVT6310	SYS1
5655-ZOS	z/OS Liberty Embedded	HWLPEM0	HWLPEM0	SYS1
5655-ZOS	Web Toolkit	HWT0500	HWT0500	SYS1
5655-ZOS	XML Toolkit for z/OS	HXML1B0	HXML1B0	SYS1
5655-ZOS	HZAI310	HZAI310	HZAI310	SYS1
5655-ZOS	z/OS Container Extensions	HZDC7C0	HZDC7C0	SYS1
5655-ZOS	z/OS File System Base	HZFS510	HZFS510	SYS1
5655-ZOS	z/OS Security Level 3 - System SSL Security Level	HCPT510	JCPT511	SYS1
5655-ZOS	GDDM NLS ENU	HGD3200	JGD3219	SYS1
5655-ZOS	ISPF UpperCase English - ENP	HIF83A2	JIF83A6	SYS1
5655-ZOS	Communications Server Security Level 3	HIP6310	JIP631K	SYS1
5655-ZOS	Communications Server X11R4 XWindows	HIP6310	JIP631X	SYS1
5655-ZOS	z/OS Security Level 3 - IBM TDS for z/OS Security	HRSL510	JRSL511	SYS1
5655-ZOS	z/OS Security Level 3 - Network Srv Security L3	HSWK510	JSWK511	SYS1
5655-ZOS	TSO/E ENU	HTE77E0	JTE77EE	SYS1
5655-ZOS	ICKDSF - DEVICE SUPPORT FACILITIES, ISMF/ENU JPN	EDU1H01	FDU1H09	SYS1
5655-ZOS	z/OS Font Collection - Chinese, Japanese, Korean	HFNT14J	HFNT14J	SYS1
5655-ZOS	BCP JPN	HBB77E0	JBB77EJ	SYS1
5655-ZOS	Cryptographic Services - System SSL JPN	HCPT510	JCPT51J	SYS1
5655-ZOS	Hardware Configuration Definition JPN	HCS77E0	JCS77EJ	SYS1
5655-ZOS	Network File System Server and Client JPN	HDZ331N	JDZ331J	SYS1
5655-ZOS	Data Facility System Managed Storage JPN	HDZ3310	JDZ331K	SYS1
5655-ZOS	GDDM NLS JPN	HGD3200	JGD3227	SYS1
5655-ZOS	ISPF Japanese - JPN	HIF83A2	JIF83A4	SYS1
5655-ZOS	JES2 JPN	HJE77E0	JJE77EJ	SYS1
5655-ZOS	Language Environment JPN	HLE77E0	JLE77EJ	SYS1
5655-ZOS	SMP/E JPN	HMP1K00	JMP1K11	SYS1
5655-ZOS	z/OS UNIX System Services Application Services JPN	HOT77E0	JOT77EJ	SYS1
5655-ZOS	IBM Tivoli Directory Server for z/OS JPN	HRSL510	JRSL51J	SYS1
5655-ZOS	Integrated Security Services-Network Authenticatio	HSWK510	JSWK51J	SYS1
5655-ZOS	TSO/E JPN	HTE77E0	JTE77EJ	SYS1
5655-ZOS	Runtime Library Extensions JPN	HTV77C0	JTV77CJ	SYS1
5655-ZOS	BCP - Support for Unicode JPN	HUN77E0	JUN77EJ	SYS1
5655-ZOS	z/OS File System JPN	HZFS510	JZFS51J	SYS1

Table 2. **z/OS 3.1 December 2023 Edition Products**

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-U59	HOURGLASS	HAD5710	HAD5710	AGG710
5655-M32	UPLOAD FOR z/OS	HPRF417	HPRF417	APS470
5655-M32	PSF for z/OS Base	HPRF470	HPRF470	APS470
5655-M32	Download for z/OS	HPRF470	JPRF472	APS470
5655-M32	ACIF	HQN4470	HQN4470	APS470
5655-M32	AFP Download Plus	HPRF470	JPRF471	APS470
5698-SA4	IBM Z SA Base Automation	HWRE430	HWRE430	AUT430
5698-SA4	IBM Z SA CICS Automation	HWRE430	JWRE43C	AUT430
5698-SA4	IBM Z SA Extended Automation	HWRE430	JWRE43F	AUT430
5698-SA4	IBM Z SA IMS Automation	HWRE430	JWRE43I	AUT430
5655-MA1	AZF	HMFA220	HMFA220	AZF220
5655-CE5	ZCONNECT - Base	HZC3000	HZC3000	BAQ30E
5655-CE5	ZCONNECT - CICS	HZC3000	JZC3002	BAQ30E
5655-CE5	ZCONNECT - Unlimited	HZC3000	JZC3003	BAQ30E
5724-V04	EWM - Build Agent	HRBA702	HRBA702	BLZ702
5724-V04	EWM - Build System Toolkit	HRBT702	HRBT702	BLZ702
5724-V04	Common Components	HRCC702	HRCC702	BLZ702
5724-V04	IBM Developer for z/OS Subset	HRDV702	HRDV702	BLZ702
5724-V04	UC DEPLOY FOR Z/OS	HRUC732	HRUC732	BUZ732
5655-TC1	IBM TOOLS CUSTOMIZER	HTCZ110	HTCZ110	CCQ110
5655-Y22	CICS INTERDEPENDENCY ANALYZER BASE COMPONENT	H274550	H274550	CIU550
5655-Y22	CICS INTERDEPENDENCY ANALYZER - ENGLISH	H274550	J27455E	CIU550
5655-Y22	CICS INTERDEPENDENCY ANALYZER - JAPANESE�	H274550	J27455K	CIU550
5655-Y20	CICS Transaction Gateway for z/OS	HCTG930	HCTG930	CTG930
5655-GOZ	IBM Open Enterprise SDK for Go	HAMF1L0	HAMF1L0	CVG121
5655-PYT	IBM Open Enterprise SDK for Python	HAMB3B0	HAMB3B0	CYP3B0
5655-X11	IBM Sterling Connect:Direct for z/OS	HDGA630	HDGA630	DGA630
5655-X11	IBM STERLING CONNECT DIRECT FOR Z/OS STANDARD EDIT	HDGA630	JDGA630	DGA630
5655-103	DITTO/ESA base	H0GB310	H0GB310	DIT130
5655-Y24	CICS VSAM Transparency for z/OS - Base	HCCV520	HCCV520	DWW520
5655-Y24	CICS VSAM Transparency for z/OS - English	HCCV520	JCCV52E	DWW520
5655-R29	IBM Rational COBOL Runtime	HACZ601	HACZ601	ELA601
5755-A01	File Manager Base Component - English	HADLF10	HADLF10	ISM402
5755-A01	Fault Analyzer - English	HADQF10	HADQF10	ISM402
5755-A01	z/OS Debugger Base	HADRG00	HADRG00	ISM402
5755-A01	Application Performance Analyzer - Base Component	HAD0F10	HAD0F10	ISM402
5755-A01	IBM z/OS Source Code Analysis	HAKGG00	HAKGG00	ISM402
5755-A01	IBM Explorer for z/OS	HALG330	HALG330	ISM402
5755-A01	Rocket Git for z/OS	HAL2111	HAL2111	ISM402
5755-A01	Z Open Automation Utilities	HAL5120	HAL5120	ISM402
5755-A01	IBM z/OS Dynamic Test Runner	HAL6100	HAL6100	ISM402
5755-A01	IBM Remote System Explorer API (RSE API)	HAMA110	HAMA110	ISM402
5755-A01	Wazi Deploy	HAMO300	HAMO300	ISM402
5755-A01	IBM Dependency Based Build	HBGZ200	HBGZ200	ISM402
5755-A01	IBM z/OS Explorer Extensions	HHOPG00	HHOPG00	ISM402
5755-A01	Application Delivery Foundation Common Components	HVWR190	HVWR190	ISM402
5755-A01	File Manager DB2 Component - English	HADLF10	JADLF12	ISM402
5755-A01	File Manager IMS Component - English	HADLF10	JADLF14	ISM402
5755-A01	File Manager CICS Component - English	HADLF10	JADLF16	ISM402
5695-013	IBM Compiler for REXX on zSeries	HWK0140	HWK0140	FAN140
5695-014	IBM Library for REXX on zSeries	HWJ9140	HWJ9140	FAN140
5695-014	IBM Alternate Library for REXX on zSeries	HWJ9143	HWJ9143	FAN140
5725-G39	IBM ZDT EE	HALME10	HALME10	FEUE00
5698-MG5	IBM Migration Utility for z/OS	HMGU510	HMGU510	FSY510
5698-MG5	REMOTE MODERNIZATION UTILITY	HRMU110	HRMU110	FSY510
5655-WSS	IBM Installation Manager install kit	HGIN140	HGIN140	GIN143
5655-Y31	ODM FOR Z/OS	HBR8A00	HBR8A00	HBR8A0
5737-B16	IBM AD Connect for Mainframe	HALT612	HALT612	IAY612
5655-PL6	IBM Enterprise PL/I for z/OS	H270610	H270610	IEL610
5655-PL6	IBM Enterprise PL/I for z/OS FEATURE	H270610	J270611	IEL610
5655-EC6	IBM COBOL FOR Z/OS BASE	HADB640	HADB640	IGY640
5655-EC6	IBM COBOL FOR Z/OS HFS	HADB640	JADB64H	IGY640
5655-EC6	IBM COBOL FOR Z/OS US ENGLISH	HADB640	JADB641	IGY640
5698-DSC	IBM DATA SET COMMANDER FOR Z/OS	HIQI910	HIQI910	IQI910
5655-DGG	IBM 31-Bit SDK for z/OS Java Technology Edition	HJVA800	HJVA800	JVA800
5655-DGH	IBM 64-Bit SDK for z/OS Java Technology Edition	HJVB800	HJVB800	JVB800
5655-DGJ	IBM Semeru Runtime Certified Edition for z/OS	HJVBB00	HJVBB00	JVBB00

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-UA1	IBM Semeru Runtime Certified Edition for z/OS	HJVBH00	HJVBH00	JVBH00
5698-CL3	IBM CL/SUPERSESSION FOR Z/OS	HKLS310	HKLS310	KLS310
5697-NV6	IBM Z NetView	HNV640B	HNV640B	NET640
5697-NV6	IBM Z NetView English	HNV640B	JNV640E	NET640

Table 3. **CICS V6.1**

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-YA1	Service Flow Runtime - Base	HCIZ300	HCIZ300	DFH610
5655-YA1	CICS - Base	HCI7400	HCI7400	DFH610
5655-YA1	CICS TS z/OS Activation Module	HCTS610	HCTS610	DFH610
5655-YA1	CICS REXX Runtime Facility	H0B5110	H0B5110	DFH610
5655-YA1	CICS REXX Development System	H0B7110	H0B7110	DFH610
5655-YA1	CICS REXX COMM FOR z/OS	H0Z2110	H0Z2110	DFH610
5655-YA1	CICS - WAS Liberty Profile	HCI7400	JCI740L	DFH610
5655-YA1	CICS - System Manager	HCI7400	JCI740M	DFH610
5655-YA1	CICS - WS Security	HCI7400	JCI740W	DFH610
5655-YA1	CICS - COBOL feature	HCI7400	JCI7401	DFH610
5655-YA1	CICS - PL/1 feature	HCI7400	JCI7402	DFH610
5655-YA1	CICS - C feature	HCI7400	JCI7403	DFH610

Table 4. **CICS V5.6**

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-Y04	Service Flow Runtime - Base	HCIZ300	HCIZ300	DFH560
5655-Y04	CICS - Base	HCI7300	HCI7300	DFH560
5655-Y04	CICS TS z/OS Activation Module	HCTS560	HCTS560	DFH560
5655-Y04	CICS REXX Runtime Facility	H0B5110	H0B5110	DFH560
5655-Y04	CICS REXX Development System	H0B7110	H0B7110	DFH560
5655-Y04	CICS REXX COMM FOR z/OS	H0Z2110	H0Z2110	DFH560
5655-Y04	CICS - WAS Liberty Profile	HCI7300	JCI730L	DFH560
5655-Y04	CICS - System Manager	HCI7300	JCI730M	DFH560
5655-Y04	CICS - WS Security	HCI7300	JCI730W	DFH560
5655-Y04	CICS - COBOL feature	HCI7300	JCI7301	DFH560
5655-Y04	CICS - PL/1 feature	HCI7300	JCI7302	DFH560
5655-Y04	CICS - C feature	HCI7300	JCI7303	DFH560

Table 5. **DB2 V12**

Product Number	Description	Fmid	Dependent FMID	HLQ
5697-Q05	DB2 Adapter for z/OS Connect	H2AZ330	H2AZ330	DSNC10
5697-Q05	DB2 INT COMP UNICODE	H2AF410	H2AF410	DSNC10
5697-Q05	DB2 Spatial Support	H2AG410	H2AG410	DSNC10
5697-Q05	JSON capabilities for DB2 for z/OS	H2AS410	H2AS410	DSNC10
5770-AF4	IBM DB2 UTILITY SUITE FOR Z/OS	HDBCC1K	HDBCC1K	DSNC10
5770-AF3	DB2 BASE Z/OS	HDBCC10	HDBCC10	DSNC10
5770-AF3	DB2 RACF EXIT Z/OS	HDREC10	HDREC10	DSNC10
5770-AF3	IRLM	HIR2230	HIR2230	DSNC10
5770-AF3	DB2 IMS ATTACH Z/OS	HIYCC10	HIYCC10	DSNC10
5770-AF3	DB2 SUBSYS INIT Z/OS	HIZCC10	HIZCC10	DSNC10
5770-AF3	DB2 JDBC/SQLJ Z/OS	HDBCC10	JDBCC12	DSNC10
5770-AF3	DB2 ENGLISH Z/OS	HDBCC10	JDBCC14	DSNC10
5770-AF3	DB2 ODBC Z/OS	HDBCC10	JDBCC17	DSNC10
5770-AF3	DB2 KANJI PANELS - JPN	HDBCC10	JDBCC11	DSNC10
5655-CH1	IBM DB2 Change Management Solution Pack	HAQG110	HAQG110	ADBC10
5655-CH1	IBM DB2 Administration Tool for z/OS	H0IHC10	H0IHC10	ADBC10
5655-CH1	IBM DB2 Object Comparison Tool for z/OS	H25GC10	H25GC10	ADBC10
5770-AF3	Z/OS APPLICATION CONNECTIVITY TO DB2 FOR Z/OS	HDDA211	HDDA211	DDC211
5697-QM2	IBM DB2 QMF HIGH PERFORMANCE OPTION	HHPCC10	HHPCC10	QMFC10
5697-QM2	DB2 QMF SERVICE	HQDCC10	HQDCC10	QMFC10
5697-QM2	IBM DB2 QMF - ENGLISH	HSQCC10	HSQCC10	QMFC10
5697-QM2	IBM DB2 QMF FOR Z/OS	HSQCC10	JSQCC1Q	QMFC10
5697-QM2	DB2 QMF - UPPERCASE ENGLISH	HSQCC10	JSQCC51	QMFC10
5697-QM2	QMF APPLICATIONS	HSQCC10	JYQCC10	QMFC10

Table 6. **DB2 V13**

Product Number	Description	Fmid	Dependent FMID	HLQ
5697-Q05	DB2 INT COMP UNICODE	H2AF410	H2AF410	DSND10
5697-Q05	DB2 Spatial Support	H2AG410	H2AG410	DSND10
5697-Q05	JSON capabilities for DB2 for z/OS	H2AS410	H2AS410	DSND10
5698-DUT	Db2 UTILITIES SUITE	HDBDD1K	HDBDD1K	DSND10
5698-DB2	Db2 BASE Z/OS	HDBDD10	HDBDD10	DSND10
5698-DB2	Db2 RACF EXIT Z/OS	HDRED10	HDRED10	DSND10

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-DB2	IRLM	HIR2230	HIR2230	DSND10
5698-DB2	Db2 IMS ATTACH Z/OS	HIYDD10	HIYDD10	DSND10
5698-DB2	Db2 SUBSYS INIT Z/OS	HIZDD10	HIZDD10	DSND10
5698-DB2	IBM DB2 JDBC/SQLJ Z/OS	HDBDD10	JDBDD12	DSND10
5698-DB2	Db2 ENGLISH Z/OS	HDBDD10	JDBDD14	DSND10
5698-DB2	Db2 ODBC Z/OS	HDBDD10	JDBDD17	DSND10
5697-Q05	SQL Tuning Services	H2AT110	H2AT110	DSND10
5698-AT3	IBM DB2 Change Management Solution Pack	HAQG120	HAQG120	ADBD10
5698-AT3	IBM Db2 Administration Tool for z/OS	H0IHD10	H0IHD10	ADBD10
5698-AT3	IBM Db2 Object Comparison Tool for z/OS	H25GD10	H25GD10	ADBD10
5698-AT3	IBM Db2 Administration Tool for z/OS Enhanced Func	H0IHD10	J0IHD10	ADBD10
5698-DB2	Z/OS APPLICATION CONNECTIVITY TO DB2 FOR Z/OS	HDDA211	HDDA211	DDD211
5698-QMF	IBM DB2 QMF Z CLIENT	HFQMD10	HFQMD10	QMFD10
5698-QMF	IBM DB2 QMF HIGH PERFORMANCE OPTION	HHPCC10	HHPCC10	QMFD10
5698-QMF	DB2 QMF SERVICE	HQDCC10	HQDCC10	QMFD10
5698-QMF	IBM Db2 QMF - ENGLISH	HSQDD10	HSQDD10	QMFD10
5698-QMF	IBM Db2 QMF FOR Z/OS	HSQDD10	JSQDD1Q	QMFD10
5698-QMF	IBM Db2 QMF APPLICATIONS	HSQDD10	JYQDD10	QMFD10

Table 7. **IMS V15.3**

Product Number	Description	Fmid	Dependent FMID	HLQ
5635-A06	IRLM	HIR2230	HIR2230	DFSF10
5635-A06	IMS SYSTEM SERVICES	HMK1500	HMK1500	DFSF10
5635-A06	IMS DATABASE MANAGER	HMK1500	JMK1501	DFSF10
5635-A06	IMS JAVA ON DEMAND FEATURES	HMK1500	JMK1506	DFSF10
5635-A06	IMS EXTENDED TERMINAL OPTION	HMK1500	JMK1503	DFSF10
5635-A06	IMS TRANSACTION MANAGER	HMK1500	JMK1502	DFSF10

Table 8. **IBM Z Monitoring Suite v2.1.0**

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-013	ITCAM for Application Diagnostics Common Services	HAAD71C	HAAD71C	ZMS210
5698-013	ITCAM for Application Diagnostics	HAAD710	HAAD710	ZMS210
5698-013	IBM Z Monitoring Suit IZSAM ID	HFZT210	HFZT210	ZMS210
5698-013	IBM Z Common Data Provider Embedded Liberty	HHBO51L	HHBO51L	ZMS210
5698-013	IBM Z Common Data Provider	HHBO510	HHBO510	ZMS210
5698-013	IBM z Service Management Explorer	HIUW631	HIUW631	ZMS210
5698-013	IBM DISCOVERY LIBRARY ADAPTER FOR Z/OS	HIZD320	HIZD320	ZMS210
5698-013	Installation/Configuration Assistance Tool	HKCI310	HKCI310	ZMS210
5698-013	IBM Z OMEGAMON for CICS	HKC5560	HKC5560	ZMS210
5698-013	IBM OMEGAMON Performance Expert for Db2 on z/OS	HKDB550	HKDB550	ZMS210
5698-013	Tivoli Enterprise Monitoring Server	HKDS630	HKDS630	ZMS210
5698-013	Apache Kafka for IBM Z	HKFK110	HKFK110	ZMS210
5698-013	IBM Z OMEGAMON for CICS TG	HKGW560	HKGW560	ZMS210
5698-013	OMEGAMON for IMS on z/OS	HKI5550	HKI5550	ZMS210
5698-013	OMEGAMON AI for JVM	HKJJ610	HKJJ610	ZMS210
5698-013	TMS:Engine	HKLV630	HKLV630	ZMS210
5698-013	OMEGAMON for IBM MQ Monitoring	HKMQ750	HKMQ750	ZMS210
5698-013	OMEGAMON Data Provider	HKOA110	HKOA110	ZMS210
5698-013	OMEGAMON Enhanced 3270 User Interface	HKOB750	HKOB750	ZMS210
5698-013	OMEGAMON for IBM Integration Bus Monitoring	HKQI750	HKQI750	ZMS210
5698-013	SHARED PROBES	HKSB750	HKSB750	ZMS210
5698-013	OMEGAMON FOR STORAGE ON Z/OS	HKS3550	HKS3550	ZMS210
5698-013	ITCAM for Application Diagnostics, TEMA	HKYN710	HKYN710	ZMS210
5698-013	OMEGAMON Integration Monitoring for DE	HRKD560	HRKD560	ZMS210
5698-013	OMEGAMON AI for Networks	HRKN610	HRKN610	ZMS210
5698-013	OMEGAMON AI for z/OS	HRKZ610	HRKZ610	ZMS210
5698-013	ITCAM for Application Diagnostics, Install	HKYN710	JKYN711	ZMS210

Table 9. **Tivoli Workload Scheduler 10.1.0**

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-T09	IBM Z Workload Scheduler agent BASE	HWSZA10	HWSZA10	TWSA10
5698-T09	IBM Z Workload Scheduler engine English	HWSZA10	JWSZA1B	TWSA10
5698-T09	IBM Z Workload Scheduler engine BASE	HWSZA10	JWSZA12	TWSA10
5698-T09	IBM Z Workload Scheduler E2E enabler	HWSZA10	JWSZA13	TWSA10

Table 10. **WebSphere Application Server ND for z/OS V9.0**

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-WAS	WebSphere App Server	HBBO900	HBBO900	WAS900

Table 11. **IBM z Distribution ZOWE V2.0.0**

Product Number	Description	Fmid	Dependent FMID	HLQ

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-ZWG	IBM Z Distribution for Zowe	AZWE002	AZWE002	ZWE200

Reported Problems, Fixes, Maintenance and Observations

- The screens for VT Terminal and TN3270 on Zowe Desktop fail to connect. A case has been opened to seek resolution.

Build Structure

Table 12. **Minumnim Set of Volumes for Simple IPL. Use LoadParm: 0XyyzzM**

Volume	Description
A3RES1,A3RES2	A3RES1 Base MVS system software target, or run-time libraries, and other system data sets required to use TSO/E and ISPF.A3RES2 An extension of the SYSRES volume **A3RES1 and A3RES2 both are required for IPL** A3RES1 is a 3390-9 volume that contains the base MVS system software target, or run-time libraries, and other system data sets required to IPL the system and use TSO/E and ISPF. This is the MVS IPL volume. A3RES2 is a 3390-9 volumes that are an extension of the SYSRES volume A3RES1. The volume is also required for IPL.
A3SYS1	System control data sets, such as SYS1.IPLPARM,IODF (I/O Definition File), and system's master catalog (CATALOG.Z31A.MASTER) reside on this volume. A3SYS1 contains the usercatalog USERCAT.Z31A.PRODS containing entries for z/OS relate products. **A3SYS1 is required for IPL** This 3390-9 volume contains data sets that you might change, either through normal use of the system or user customization. System control data sets, such as SYS1.IPLPARM,IODF (I/O Definition File), and system's master catalog (CATALOG.Z31A.MASTER) reside on this volume. This volume is required to IPL. A3SYS1 contains the usercatalog USERCAT.Z31A.PRODS containing entries for z/OS related products.
A3CFG1	The 'USER.**' data sets reside on this volume, they are cataloged in the master catalog. The RACF database resides on this volume. **A3CFG1 is required for IPL** This 3390-9 volume contains data sets for storing user configuration. The 'USER.**' data sets reside on this volume, they are cataloged in the master catalog. The RACF database reside on this volume.
A3USS1, A3USS2, A3USS3	A3USS1 Contains the Fonts, Version and zOSMF zFS files for UNIX System Services of z/OS. A3USS2 Contains root zFS and zFS files for UNIX System Services of z/OS A3USS3 Contains root zFS and zFS files for UNIX System Services of z/OS **A3USS1, A3USS2, and A3USS3 are all required for IPL** A3USS1 is a 3390-9 volume that contains the Fonts, Version and zOSMF zFS files for UNIX System Services of z/OS. A3USS2 is a 3390-9 volume that contains root zFS and zFS files for UNIX System Services of z/OS. A3USS3 is a 3390-9 volume that contains root zFS and zFS files for UNIX System Services of z/OS.
A3PAGA, A3PAGB, A3PAGC	A3PAGA contains the PLPA, common and one Local page data set. A3PAGB contains Contain one Local page data set A3PAGC contains Contain one Local page data set **A3PAGA is required for IPL** **Note: The loadparms that bring up DB2 or WAS have been defined to use A3PAGA A3PAGB and A3PAGC , all other loadparms use only the first two volumes. If you do not want to use all of these local page data sets, you can modify the PAGE parameter on IEASYSnn member, accordingly.** These volumes contain page data sets. The A3PAGA volume has the PLPA, Common and one Local page data set. Rest of the 2 volumes each have one Local page data set. The loadparms that bring up DB2 or WAS have been defined to use all the above 3 page volumes, all other loadparms use only the first two volumes. If you do not want to use all of these local page data sets, you can modify the PAGE parameter on IEASYSnn member, accordingly.
A3PRD1,A3PRD2,A3PRD3,A3PRD4,A3PRD5	A3PRD1 to A3PRD5 are 3390-9 volumes that contains all z/OS products.
SARES1	This 3390-9 volume contains a single volume stand alone system. This volume can be used to IPL and logon to a TSO/ISPF session. The disk volume can assist in building LPAR environments and correct errors that prevent system IPL. Recommended use of this volume is to install the volume and leave accessible. The volume should not be altered. The volume should stay in a background mode and be available for emergency type of situations. If other z/OS, OS/390, or LPAR partitions contain errors, the SARES1 volume could be used to solve the problem. The stand alone system does not contain TCPIP or Unix system service support. The system cannot be used to install products or apply maintenance via SMP/E. **No other volumes required to IPL this Standalone volume. Use the following information:** Load Address: **0Axx** Load Parms: **0AxxSAM**

Table 13. **Volumes for Supported Products**

Volume	Description
A3DIS1,A3DIS2,A3DIS3	These volumes contains the base MVS system software distribution libraries. They contain the DLIB (Distribution Library) and zone CSI's. These volumes are needed to install service or products on the MVS system using SMP/E. These volumes are all 3390-9.

A3DBD1,A3DB D2,A3DBAR	A3DBD1 is an 18,000 cylinder volume that contains DB2 V13 related data. A3DBD1 contains the DB2 usercatalog, USERCAT.Z31A.DB2VD. Also A3DBD1 contains all DB2 target libraries and DB2 V13 utility products. This volume is not required to IPL but is required to bring up DB2 V13. A3DBD2 is a 3390-9 volume that contains DB2 catalog and directories with HLQ DSNCD10, which are SMS managed. The volume is not required for IPL but is needed to bring up DB2. A3DBAR is a 3390-9 volume defined in a new esoteric called DBARCH. This esoteric has been defined in DB2, for allocating the archive logs into this storage volume. This volume is required, if you want to bring up DB2 V13.
A3DBC1,A3DB C2,A3DBAR	A3DBC1 is a 3390-9 volume that contains DB2 V12 related data. A3DBC1 contains the DB2 usercatlog, USERCAT.Z31A.DB2VC. Also A3DBC1 contains all DB2 target libraries and DB2 V12 utility products. This volume is not required to IPL but is required to bring up DB2 V12. A3DBC2 is a 3390-9 volume that contains DB2 catalog and directories with HLQ DSNCC10, which are SMS managed. The volume is not required for IPL but is needed to bring up DB2. A3DBAR is a 3390-9 volume defined in a new esoteric called DBARCH. This esoteric has been defined in DB2, for allocating the archive logs into this storage volume. This volume is required, if you want to bring up DB2 V12.
A3C560	A3C560 is a 3390-9 volume that contains CICS 5.6 Target, DLIB and all related data for CICS 5.6. USERCAT.Z31A.CICS560 resides on this volume. This volume is not required to IPL but is required to bring up CICS 5.6 and apply CICS maintenance.
A3C610	A3C610 is a 3390-9 volume that contains CICS 6.1 Target, DLIB and all related data for CICS 6.1. USERCAT.Z31A.CICS610 resides on this volume. This volume is not required to IPL but is required to bring up CICS 6.1 and apply CICS maintenance.
A3W901, A3W902	A3W901 and A3W902 are 3390-9 volumes that contains target and distribution libraries for Websphere Application Services.
A3IMF1	A3IMF1 is a 3390-9 volume that contains IMS Target, DLIB and all related data for IMS 15. USERCAT.Z31A.IMS15 resides on this volume. This volume is not required to IPL but is required to bring up IMS and perform maintenance on IMS.
A3KAN1	A3KAN1 is a 3390-9 volume that contains Target and DLIB libraries for IBM Z Monitoring Suite 1.3.0.
A3BLZ1	A3BLZ1 is a 3390-9 volume that contains Target and DLIB libraries for IBM Engineering Workflow Management 7.0.2. This volume is not required to IPL but is required to bring up RTC.
A3INM1	A3INM1 is a 20,000 cylinder volume that contains Target and DLIB libraries for Installation Manager 1.8.8.

Table 14. **Volumes for SMS Support**

Volume	Description
A3USR1	The data class DCEXTEAV is defined to allow larger than four Gigabyte dataset allocation. With this data class and the accompanying storage group and class, a dataset can be allocated on the provided SMS managed volume A3USR1. The ACS routines are defined so no filtering is needed. To allocate and SMS managed dataset, specify the data class (DCEXTEAV) in the allocation JCL and the dataset will be defined on the provided volume. Below are the names of the Table 14. **SMS Information** | Item	Description
DCEXTEAV	Data Class
SGEXTEAV	Storage Group
SCEXTEAV	Storage class
A3USR1	Volume defined to SGEXTEAV (Storage Group)
SYS1.SMS.CNTL	Location of the SMS routines	
A3ZCX1	The data class CXDC is defined to allow larger than four Gigabyte dataset allocation. With this data class and the accompanying storage group and class, a dataset can be allocated on the provided SMS managed volume A3ZCX1. The ACS routines are defined so no filtering is needed. To allocate and SMS managed dataset, specify the data class (CXDC) in the allocation JCL and the dataset will be defined on the provided volume. Below are the names of the Table 15. **SMS Information** | Item	Description
CXDC	Data Class
CXROOTSG	Storage Group
CXROOTSC	Storage class
A3ZCX1	Volume defined to CXROOTSG (Storage Group)
SYS1.SMS.CNTL	Location of the SMS routines	

Table 17. **Volume for zOWE Support**

Volume	Description
A3ZWE1	A3ZWE1 is a 3390-9 volume that contains ZOWE Target, and DLIB This volume is not required to IPL but is required to bring up zOWE .

Migration Guidelines

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

Only IBM-supplied system software should reside on A3xxxx volumes except for A3SYS1. A3CFG1 contains RACF, IPL, and catalog datasets that are user dependent.

The ADCD does not contain a generalized migration utility or process. Each user has unique requirements. It is recommended that you build a migration plan based on your unique needs. The ADCD does have some assistance. The ADCD uses a system of concatenated libraries (see below) The highest level of concatenation is USER.Z31A.xxxxxx The ADCD distributes these libraries empty. It is recommended that the user place changes or overrides in these libraries. It is further recommended that the USER.Z31A.xxxxxx libraries be backed up regularly. When a new ADCD release is installed the USER.Z31A.xxxxxx libraries can be copied from backups to the new USER.Z31A.xxxxxx libraries to provide assistance in migration of programs and parameters.

The ADCD is distributed with a standard library concatenation for the following:

- LINKLST
- PROCLIB (including TSO procedures)
- CLIST
- ISPPLIB (TSO panels)
- LPALIB

- PARMLIB
- VTAMLST
- VTAMLIB
- VTAM source

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

- USER.Z31A.LINKLIB
- ADCD.Z31A.LINKLIB
- SYS1.LINKLIB

USER.Z31A.xxxxxx libraries have all been built on A3CFG1 which is the only volume that should contain user data.

USER.Z31A.xxxxxx will not be changed by ADCD or System processes; thus, user updates in USER.Z31A.xxxxxx will be retained between release levels. ADCD.Z31A.xxxxxx libraries are allocated on A3SYS1 and are the libraries that are used by ADCD developers. No RACF rules exist on these libraries; however, changes to these libraries could destroy customization necessary to bring up many products. SYS1.xxxxxxx libraries should NEVER be updated by users or ADCD development. System libraries should ONLY be updated through SMP/E install, maintenance processes, or like processes.

NOTE: Changes to System libraries or ADCD pre-customization may corrupt your system and prevent or delay IBM support activites.

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

- Copy the system data from SYS1.xxxxx or ADCD.xxxx to USER.Z31A.xxxxxx
- Edit, compile, or run user program against the USER.Z31A.xxxxxx item
- Re-ipl, recycle system service, or re-logon to TSO

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

LOADPARMS Options

Note: JES2 should be COLD started the FIRST time you bring up the system.

Table 18. **Table Listing of available pre-configured distributed LOADPARMS**.

LOADPARM	Description
CS	CLPA and Cold start of JES2. Base z/OS system functions i.e. no CICS, DB2, IMS, WAS, etc.
WS	CLPA and Warm start of JES2. Base z/OS system functions i.e. no CICS, DB2, IMS, WAS, etc.
00	CLPA and Warm start of JES2. Base z/OS system functions i.e. no CICS, DB2, IMS, WAS, etc.
CI	CLPA and Warm start of JES2. Loads CICS 5.6 and 6.1 libraries. up Starts up CICS 6.1 and RDz.
DB	CLPA and Warm start of JES2. Loads DB2 V13 and V12 libraries. Starts up DB2 V13 and IDz.
IM	CLPA and Warm start of JES2. Loads IMS 15 libraries. up Starts up IMS 15 and IDz.
IZ	CLPA and Warm start of JES2. Starts up z/OSMF and IDz.
WA	CLPA and Warm start of JES2. Loads WAS 9.0 libraries.Starts up IDz. WAS needs to be manually started.
AL	CLPA and Warm start of JES2. Loads all middleware libraries. Starts up CICS 6.1, DB2 V13, IMS 15, z/OSMF and IDz.
DC	CLPA and Cold start of JES2. Loads CICS, DB2 libraries. Starts up CICS 6.1, DB2 V13, UCD 7.3.2, RTC 7.0.2 and IDz.
DW	CLPA and Warm start of JES2. Loads CICS, DB2 libraries. Starts up CICS 6.1, DB2 V13, UCD 7.3.2, RTC 7.0.2 and IDz.
ZE	CLPA and Warm start of JES2. Loads all middleware libraries. Starts up z/OSMF, Zowe, IDz, and UCD. Other subsystems need to be manually started if desired.

Console PF Key Settings

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

USERIDS

Table 19. **Table of Pre-Defined USERIDS. The following TSO Userids and Passwords have already been set up on your system:**

Userid	Password
ADCDMST (RACF special authority)	SYS1 or ADCDMST
IBMUSER (RACF special authority)	SYS1 or IBMUSER
SYSADM (DB2 and RACF special authority)	SYS1 or SYSADM
SYSOPR (DB2 and RACF special authority)	SYS1 or SYSOPR

Userid	Password
WEBADM	WEBADM

Maintenance Service Levels

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

- All functions of the base z/OS 3.1 December 2023 Edition are at PUT2309 / RSU2309
- All functions of the CICS TS 5.6 are at PUT2309 / RSU2309.
- All functions of the CICS TS 6.1 are at PUT2309 / RSU2309.
- The functions of the DB2 12 base are at PUT2309 / RSU2309.
- The functions of the DB2 13 base are at PUT2309 / RSU2309.
- All functions of the IMS 15.1.0 are at PUT2309 / RSU2309.
- IBM Z Monitoring Suite V1.3.0 is at PUT2309 / RSU2309.

Extended ADCD for ZD&T

Extended ADCD is built upon the general release of ADCD z/OS® V2R5 May Edition of 2023. Extended ADCD contains some changes to support IBM Cloud Pak® for Applications, and improve the z/OS system start performance for Java™ heavy subsystems.

To support IBM Cloud Pak for Applications, some new software packages are added or updated to ADCD z/OS V2R5 May Edition of 2023.

Software packages that are added:

- z/OS Explorer Extensions Version 16.0.3 (HLQ - ISM332)
- z/OS Explorer API Version 1.1 (HLQ - ISM332)
- IBM Z® Open Automation Utilities Version 1.2.5 (HLQ - ISM332)
- Rocket Git 2.42.0-6 (HLQ - ISM332)
- Rocket Bash 5.2.15 (HLQ - ISM332)
- Rocket Perl 5.32.1 (HLQ - ISM332)
- IBM® Python SDK 3.11.4 (HLQ - CYP390)

Software packages that are updated:

- IBM Dependency Based Build is updated to 2.0 (HLQ - ISM333)
- IBM Debugger is updated to 16.0.3 (HLQ - ISM332)
- z/OS Explorer is updated to 3.3 (HLQ - ISM332)

Java enhancements

The changes that are made in Extend ADCD do not significantly affect the ADCD packages. The only exception is to improve the z/OS system start performance for the following Java heavy subsystems, especially when you introduce extra layers of virtualization with OpenShift®.

- z/OS Explorer, Explorer API, and Explorer Extensions
- z/OS Debug Manager, Debug Profile, and remote Debug Service
- DBB
- z/OSMF
- z/OS Connect
- UrbanCode® Deploy

- **Hints and tips to manage Extended ADCD**
  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.
- **Customized TCP/IP network configurations**
  This section documents the TCP/IP configuration changes that are made to ADCD for ZD&T.
- **Customized SMS configuration in Extended ADCD**
  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 V2R5 May Edition of 2023.
- **Automated NetView**
  This section documents some system automations that are available to start or stop the system and certain z/OS subsystems.
- **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.

Hints and tips to manage Extended ADCD

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® V2R5 May Edition of 2023.

Note: The following topics from Application Developers Controlled Distribution (ADCD) Release Guide z/OS V2R5 May Edition of 2023 are changed.

- **ACS routines provided on ADCD**
  The SMS configuration that is documented in [Customized SMS configuration in Extended ADCD](#) 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](#).

- **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 [Customized SMS configuration in Extended ADCD](#).

Customized TCP/IP network configurations

This section documents the TCP/IP configuration changes that are made to ADCD for ZD&T.

TCP/IP addressing was symbolized in the system parmlib configuration. A few system symbolic parameters are defined in FEU.Z25C.PARMLIB. If Extended ADCD is used for different purpose, different members are selected. If Extended ADCD is used for the IPL in ZD&T, the member IEASYMAU is selected. If Extended 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.Z25C.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 DYNXCFIPADDRESS will be modified by ZD&TEnterprise 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.Z25C.TCPPARMS(ZPDTDEV1). When an instance is provisioned from a created ADCD image, the value of HOMEIPADDRESS1 will be modified by ZD&TEnterprise 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.Z25C.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&TEnterprise 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.Z25C.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.Z25C.PROCLIB(RESOLVER).

Customized SMS configuration in Extended ADCD

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® V2R5 May Edition of 2023.

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 **ALCCKD**.
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/ZDTDPY
 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 otion** 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 Extended 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.

Automated NetView

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.

Table 1. Valid NetView Commands

Commands	Description
`SHUTSYS`	Shut down all z/OS system tasks.
`GETSTAT`	Show the status of all z/OS system tasks that are managed by NetView.
`STRTTASK task name`	Start a managed task.
`STOPTASK task name`	Stop a managed task.
`SETSTAT task name`	Set a specific status for a managed task.
`DSPSTAT`	Display and update the status for all managed tasks.
`GETGLBVA task name`	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&TEnterprise 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&TEnterprise 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&TEnterprise Edition. However, when the manual IPL process of z/OS is done, ZD&TEnterprise 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 tasks 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.

Adding tasks 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.Z25C.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.

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:

RESTCASH
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 ZOSCSRV
- 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 tasks to NetView automation](#).

Troubleshooting and support

Learn how to troubleshoot the product.

- **[Troubleshooting for installation](#)**
- **[Troubleshooting for the installation, operating, and uninstallation of the software-based license server](#)**
Learn how to troubleshoot problems installing, operating, and uninstalling the software-based license server.
- **[Troubleshooting for USB Hardware device activation and licensing](#)**
- **[Troubleshooting for the installation of z/OS volume images](#)**
- **[Troubleshooting for defining the ZD&T machine characteristics (DEVMAP)](#)**
- **[Troubleshooting for the starting and stopping of ZD&T](#)**
- **[Troubleshooting for extraction issues](#)**
Learn how to troubleshoot for extraction issues.
- **[Troubleshooting for z/OS networking issues](#)**
- **[Troubleshooting for performance issues](#)**
- **[Troubleshooting for connecting with Rational License Key Servers](#)**
- **[Troubleshooting for software-based license server and USB hardware device activation](#)**
- **[Troubleshooting tips](#)**
- **[Obtaining the latest editions of Redbooks](#)**
IBM® Redbooks® provide information about major components of ZD&T.
- **[Contacting IBM Software Support](#)**
Learn how to contact IBM Software Support for help with problems, fixes, and other support information.

Troubleshooting for installation

Table 1. Troubleshooting the installation of the ZD&T

Symptom	Problem determination
Message CRIMA1076E in the install log when you are uninstalling a version of ZD&T	For the step-by-step process to perform if an uninstall fails., see <u>Uninstalling the earlier versions of</u>.

When you are attempting to run launchpad.sh or launchpad-console.sh, you get the following error:
 	
`./InstallerImage_linux.gtk.x86_64/`
    `tools/imcl: Permission denied`
	This issue can happen for several reasons:
	
	• 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.

When you are attempting to perform an Installation Manager update for ZD&T, or to run `launchpad.sh` or `launchpad-console.sh`, you get one of the following errors:
 	
`sntl-sud-xxxx is required`
    `by z1091-1-xxxx but is already`
    `installed`
 	
Or
 	
`zpdt-shk-server-xxxx is`
    `required by z1091-1-xxxx`
    `but is already installed`
	• 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 <u>Uninstalling the earlier versions of</u>.
	
	• If these messages are from running `launchpad.sh` or `launchpad-console.sh`, perform an uninstall of the previous version before you attempt a new installation. For the step-by-step process to perform an uninstall, see <u>Uninstalling the earlier versions of</u>.

You receive any error message when you are uninstalling a previous version of ZD&T using the steps outlined in <u>Uninstalling the earlier versions of</u>.	1. If you are uninstalling Version 9.0 or 9.1, run the `z1091_removeall` command
	  • Start a Linux console as user root.
	  • Change Directory (`cd`) to `/usr/z1090/bin`, the folder that contains the `z1091_removeall` command.
	  • Run the command as follows:
	    `./z1091_removeall`
	
	  • 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 (`zpdt-shk-server`) or Sentinel System Driver (`sntl-sud`), attempt these steps in the following order:
	  • Enter # `rpm -e --allmatches z1090`
	  • Enter # `rpm -e --allmatches z1091`
	  • Enter # `rpm -e --allmatches zpdt-shk-server`
	  • Enter # `rpm -e --allmatches sntl-sud`

When an installation fails for the ZD&T software-based license server, and the log displays the following entries:
 	
`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`
	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.

Troubleshooting for the installation, operating, and uninstallation of 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
```

Troubleshooting for USB Hardware device activation and licensing

Troubleshooting for USB Hardware device activation and licensing

Table 1. USB Hardware device activation and licensing. Troubleshooting USB Hardware device activation and licensing

Symptom	`SecureUpdateUtility` or `Z1091_token_update` does not apply the update file to the USB hardware device.
Problem Determination	• 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 `lsusb` 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 `SecureUpdateUtility` `-u` or the `z1091_token_update` command
Problem Determination	This issue can happen for several reasons: • 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 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.

Symptom	When you are starting ZD&T, (`awsstart`) you get any of the following messages at the Linux terminal that indicates a failure to get a license:
	
	• `Error: Failed to get license.`
	`        Return code: 312`
	
	• `Error: Unable to get expected`
	`        license: xxxx`
	
	or
	
	`Error: Unable to locate expected`
	`        license: xxxx`
	
	• `AWSEMI315E zPDTA License Unavailable`
	`        for CPU 0`
	`AWSEMI005I Waiting for 1090 license`
	
	• `CPU Address Out Of Range`
	
	• `SFNTGetLicense failed.`
	`        Return code: 312`

Problem Determination	This issue can happen for several reasons:
	• The local USB Hardware device is not plugged in properly. Attempt to "reseat" the USB Hardware device. Unplug the device for 15 seconds, then reinsert it.
	• The correct update file is not applied to the matching USB Hardware device, or it did not apply successfully.
	  ○ 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 `SecureUpdateUtility -u` or `Z1091_token_update`
	  `-u` command responded with a "`Success`" message.
	• The requested license type is not correct.
	  ○ Validate that if your device map includes the `cpuopt zvm_couplingfacility` directive, your USB hardware key includes sysplex enabling licenses. If no sysplex license is available, your device map cannot enable sysplex operation.
	  ○ Validate that if your device map includes the `RDTSERVER=`
	  `directive` or you have an `RDTSERVER` 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
	  ○ From a Linux terminal that is running as root, enter the `clientconfig` command that is found in `/usr/z1090/bin`. Ensure the
	  `License`
	  `ContactServer` is `localhost` if you are attempting to use a local USB. No other fields need be completed.
	• The `/usr/z1090/bin/sntlconfig.xml` file is not readable by the programs that need it. Verify that `/usr/z1090/bin/sntlconfig.xml` is "world readable". If it is not, use `chmod` to change the permission bits to `644`.

	• You do not have the correct type of USB Hardware Device. ○ Perform the following command from a Linux terminal that is running as root: `rpm -qa	grep z109`        If Rational Development and Test Environment for z System is properly installed, you should see an rpm of the form        `z1091-xxx and not z1090-xxx` ○ 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.

Problem Determination (continued)	• You do not have connectivity from the instance that is performing the `awsstart` to the product license server.
	   ○ Enter a `UIMCHECK` on the client instance to verify whether there is proper connectivity. You see the message
	
	      `The remote QUERY command executed successfully`
	
	   ○ If this procedure does not complete successfully, try the following procedure.
	     ■ From a Linux terminal that is running as root, enter the `clientconfig` command that is found in `/usr/z1090/bin`. Ensure the `License ContactServer` is the IP address of the product license server, that the `License PortNumber` is `9450`, and that `License ipv6` is `yes` or `no` as is appropriate for your installation.
	     ■ 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.
	        ■ Ensure that the server was started once by a non-root ID with the `uimserverstart` 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
	
	       `iptables -L -n`
	
	       Ensure ports `9450` and `9451` are allowed.
	     ■ 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
	
	     ■ `/opt/safenet_sentinel/common_files/`
	          `sentinel_keys_server/sntlconfigsrve.xml`
	
	       For more information, see paragraph 8.4.2, "Security" in chapter 8 of the [zPDT® Guide and Reference](#).
	
	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:
	
	`/opt/safenet_sentinel/common_files/sentinel_keys_server/loadserv`
	      `restart`

Symptom	When you are starting Rational Development and Test Environment for z Systems®, (`awsstart`) you get the following messages at the Linux terminal that indicate an expired license
	
	`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`

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 *ZD&T Activation Guide (SC27-6630)*.
Symptom	Time cheat errors
Problem Determination	• 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 for the installation of z/OS volume images

Table 1. Trouble shooting the installation of z/OS® volume images

Symptom	Problem Determination

You get the following errors while performing a z1091_ADCD_install command:
 	
`Error: Failed to get license.`
    `Return code:`
`312.`
`Error: Unable to`
	For information on possible causes, see the same symptom in "[Troubleshooting for USB Hardware device activation and licensing](#)".
	Also this can occur if the USB Hardware device, either local or in a product license server, has been updated with a `.upw` update file instead of a `.zip` update file. If you are migrating a ZD&T instance from version 9 to version 9.1, you must request a new update file from the Rational® License Key Center to install the ADCD `S1RES1` and `SARES1` volumes.

```
locate expected
     license: 0D98
```

Troubleshooting for defining the ZD&T machine characteristics (DEVMAP)

Table 1. 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.

Troubleshooting for the starting and stopping of ZD&T

Table 1. 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.shmmni = 8192` to `/etc/sysctl.conf` and activate it using the command `sysctl -p`.

The following message occurs when ipling 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.

Troubleshooting for extraction issues

Learn how to troubleshoot for extraction issues.

Symptom	Problem determination

When you create components from a z/OS system, the extraction fails with the following error message:
 	
`Exception in thread "main"`
`java/lang/OutOfMemoryError:`
`native memory exhausted`
	To run the extraction code, minimum region size of the z/OS user ID must be met, which is approximately 150MB for 32-big Java or 250MB for 64-bit Java. To solve the issue, try to increase the region size.
	The region size can be specified in the OMVS segment (**ASSIZEMAX**) for remote login, for example, SSH, and for TSO login through the TSO segment (**SIZE** or **MAXSIZE**). If OMVS or TSO segments do not limit sizes, global settings might need to be adjusted in the BPXPRM*xx* member of PARMLIB.
	
	Example
	
	`alu ibmuser OMVS(ASSIZEMAX(262144000))`

Troubleshooting for z/OS networking issues

Table 1. z/OS networking issues . Troubleshooting z/OS® networking issues

Symptom	General z/OS networking configuration errors
Problem Determination	• Check syslog (z/OS console messages) for errors during the start of TCP/IP or VTAM®. • Validate configuration files match for device numbers and names in the device map, VTAM definitions, and TCP/IP configuration profile. Specifically, follow the definitions to ensure that the tunnel and external addresses in the device map are defined through VTAM and TCP/IP to be the correct device names and addresses in z/OS. These configuration errors can be tricky because device names in the TCP/IP profile are arbitrary.
Symptom	Unable to establish a network tunnel on a `10.x.x.x` network.

Problem Determination	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:
	
	`name awsosa 0009 --path=A0`
	`        --pathtype=OSD`
	`        --tunnel_intf=y`
	`        --tunnel_ip=192.168.1.1`
	
	Configure z/OS TCP IP profile as a tunnel IP of
	
	`192.168.1.2 netmask 255.255.255.0`

Symptom	Cannot connect to ZD&T z/OS using a remote 3270 terminal emulator
Problem Determination	• Ensure that you have network connectivity to the Linux® adapter by pinging the Linux adapter IP address from the machine that is running your remote 3270 terminal emulator

	- Ensure that you have a tunnel OSA defined with an internal `10.x.x.x` address. Ensure that there is a matching route that sends all `10.x.x.x` traffic through the tunnel OSA - Ensure that the remote 3270 terminal emulator configuration is attempting to connect to the Linux IP address or host name, and port 3270 (or whatever port is defined on the 3270port statement) - Ensure that your firewall configuration allows traffic to the 3270port. See Firewall considerations in "Configuring the base Linux Server" above for how to configure the firewall to allow the appropriate ports.
Symptom	Cannot ping to a z/OS defined IP address on the ZD&T instance.
Problem Determination	- 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 <u>zPDT® Guide and Reference</u>. - 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 <u>zPDT Guide and Reference</u>, the z/OS IP address is advertised through the network using Linux Network Address Translation (NAT) functions. - Check your firewall configuration with the command `iptables -L -n` - 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 `firsthop` address is the Linux tunnel address if all traffic is being routed through the Linux. - For more information, see Chapter 7 "LANs" in the <u>zPDT Guide and Reference</u>.
Symptom	Cannot SFTP or Telnet to a z/OS defined IP address on the ZD&T instance
Problem Determination	- Verify that you can ping to the address from the remote location. If you cannot, follow the steps defined <u>here</u>. - If you can ping, check your firewall configuration with the command `iptables -L -n` - Ensure that the appropriate ports are allowed; typically ports 22 for SFTP and 23 for Telnet

Troubleshooting for performance issues

Table 1. Performance. Troubleshooting for performance issues

Symptom	High volume traffic through an emulated OSA-Express® ethernet adapter on ZD&T suffers poor throughput.

Problem Determination	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:
	
	<pre># ethtool -K eth0 rx off (disable RX checksumming offload)
	# ethtool -K eth0 tso off (disable TCP segmentation offload)
	# ethtool -K eth0 gso off (disable generic segmentation
	      offload)
	# ethtool -K etho 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>
	You might need to experiment with these commands.
	
	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.
	
	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
	
	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:
	
	`net.core.rmem_max=1048576`
	
	For more information on zPDT® performance and performance problems, see paragraph 2.3.4 "Performance" and 7.5.9 "Performance problems" in the <u>zPDT Guide and Reference</u>.

Symptom	ZD&T performs slowly. CPUs reach 100% utilization.
Problem Determination	Verify that you are meeting the minimum requirements for both the hardware and the software. See <u>Hardware and software requirements</u> and paragraph 2.3.2 "zPDT instances" in the <u>zPDT Guide and Reference</u>.

Troubleshooting for 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 for software-based license server and USB hardware device activation

Configuration and usage for ZD&T is described in Configuring Z instances and the zPDT® Guide and Reference.

Troubleshooting tips

Table 1. Troubleshooting tips

Symptom	Problem Determination
For any problem after installation of ZD&T (creating a devmap, installing z/OS®, starting or stopping ZD&T, z/OS networking issues, performance issues)	Validate the environment and the installation by running the **z1090instcheck** command and fixing errors. Do not worry about errors that are related to **kernel.core_pattern** unless the emulator is terminating abnormally.
AWScccnnns message that is issued from the Linux® terminal that is running ZD&T.	Any message beginning with AWS is a zPDT® message. Enter the following zPDT command **msgInfo message-number** 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.

Obtaining the latest editions of Redbooks

IBM® Redbooks® provide information about major components of ZD&T.

Refer to the latest edition of *IBM zPDT® Guide and Reference* for the use with ZD&T.

Contacting 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 Support Guide*.

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.

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 IBM 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 an IBM License Key Server**
- **Updating the device map to point to an IBM License Key Server**
  The IBM 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.
- **Troubleshooting connections with IBM 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 IBM 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 IBM 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.

Installing, configuring, and administering an IBM License Key Server

Setting up the IBM License Key Server is beyond the scope of this documentation, but additional information can be found in the IBM Common Licensing documentation.

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 an IBM License Key Server

The IBM 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 IBM 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 IBM 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 IBM 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 IBM 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 IBM 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 IBM 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 IBM 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 Terminology.
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 1 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 1. Options for obtaining zPDT licenses
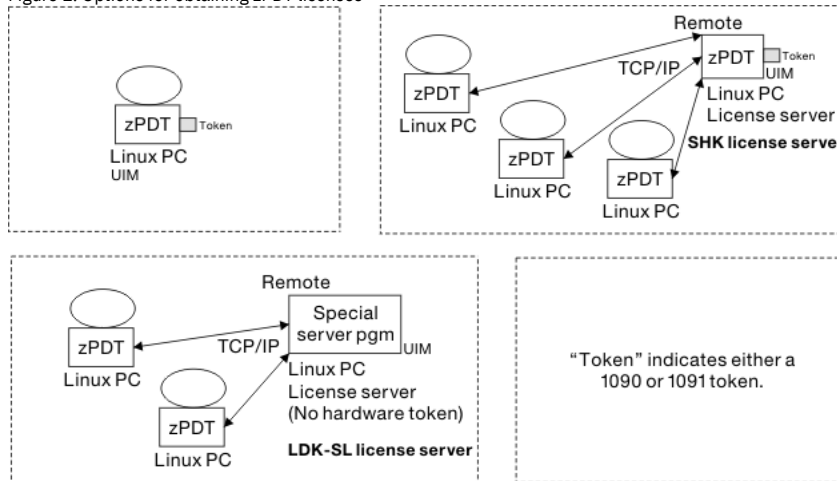
In a simple configuration, a *local token* is installed in a USB port on the base machine running zPDT. In this case (one token installed in a local USB port), the token supplies both the zPDT license and the serial number used for the z Systems CPs, assuming that the local zPDT system has never been 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 1 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.
- **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.
- **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).
- **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.
- **Scenarios**
  Learn about common usage scenarios.
- **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.
- **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.
- **Allocating multiple licenses**
 Learn how the zPDT system allocates multiple licenses to clients.

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 1 illustrates the conceptual operation.
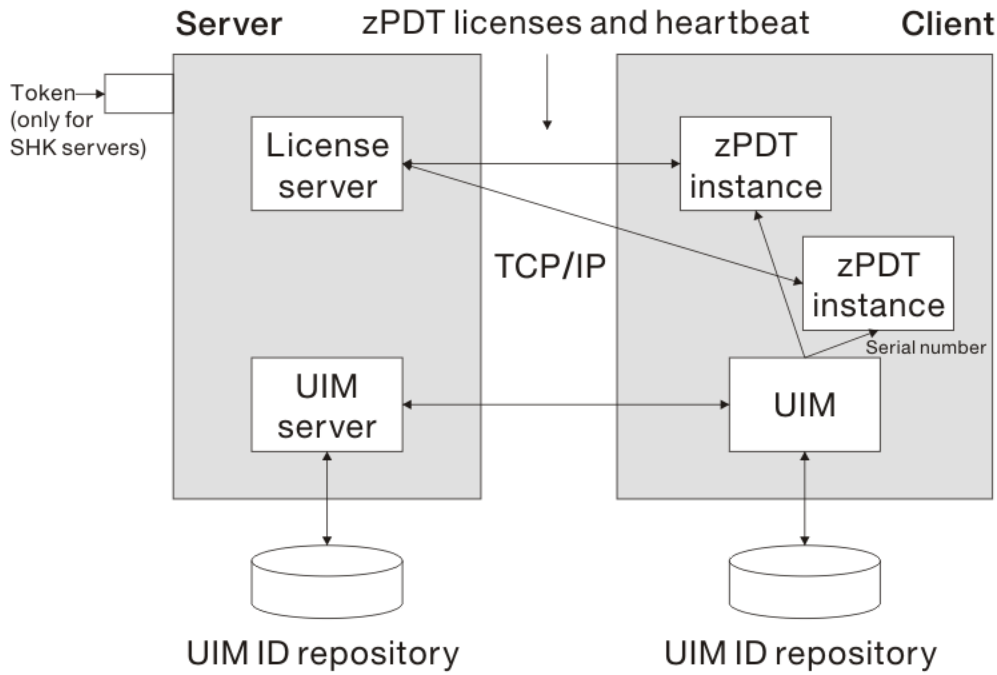
Figure 1. Simple local operation
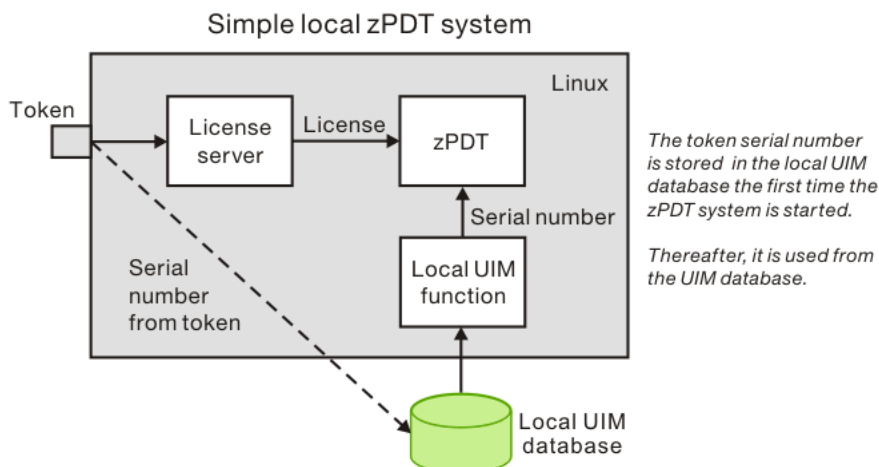
When a remote license server is used (or if multiple local tokens are used) there needs to be a method of assigning unique serial numbers that do not change after they are assigned. Figure 2 illustrates the general concept.

Figure 2. 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 1, 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 2:

- 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.

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 nomal 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.
- **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.
- **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.

SHK client configuration

After a nomal 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 and LDK client configuration

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.
- **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.
- **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.

UIM server

The UIM server is automatically installed when installing either the SHK or the LDK-SL license server packages provided for zPDT®.

Once installed, the remote UIM server must initially be started manually; thereafter it is automatically managed by cron. It must not run as root. It runs under a normal Linux® userid and places its database in the home directory of that userid. It also places small log files in the home directory. For this reason, the same Linux userid (not

root) should always be used to run the UIM server.

Two commands are associated with running the UIM server:

```
$ uimserverstart
    Start the UIM server.
$ uimserverstop
    Stop the UIM server.
```

The **uimserverstart** command, in addition to starting the server, places entries in the Linux cron files such that the UIM server is restarted automatically (after 10 minutes) if it fails. It is also started automatically during a Linux reboot. The **uimserverstop** command stops the server and removes these cron entries.

No other configuration is needed for the UIM server. You must not edit the UIM database file that is created in a subdirectory of the home directory of the userid running the UIM server.

SHK license server

The SHK license server is part of the standard zPDT® package and is installed as if you were installing a zPDT client. It is activated by the actions of the two token "driver" components that are part of zPDT installation.

One or more 1090 or 1091 tokens must be installed in the license server machine before it can be used. The license server configuration file is located in: /opt/safenet-sentinel/common_files/sentinel_key_server/sntlconfigsrvr.xml

This file typically does not require any additional configuration. If you want to change the license server port number, you can edit and change this file. You would then need to restart the server by using these commands:

```
#        cd /opt/safenet_sentinel/common_files/sentinel_keys_server
```

```
#        ./loadserv restart
```

Several security functions may be specified in the sntlconfigsrve.xml file.

LDK-SL License server

Several steps are involved in preparing an LDK- SL license server. The license server (and the associated UIM server) are supplied in a file with a name similar to that shown in the following command.

Place this file in a convenient directory and, working as root, execute this file:

```
#         ./zPDT_LS-1.6.49.20L-x86_64
```

This file must be executable. This might require a **chmod u+x** operation. Also, the exact file name may change slightly to match newer levels of zPDT®.

The installation process causes an Internet search for the latest version of the 32-bit glibc library, as described in LDK client configuration. 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.
- **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.

- **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.
- **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.

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 **SercureUpdateUtility** 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.
- **Temporarily switch from server to local**
  Learn how to temporarily switch a client from a remote license and UIM server to local license.
- **Switch from remote server to local**
  Learn how to switch from a remote license and UIM server to a local license.
- **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.
- **Switching tokens**
  Learn how to switch tokens in a local UIM database.
- **Change from single token to multiple tokens**
  Learn how to switch from a single token to one of several other tokens.
- **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.
- **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.
- **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.
- **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.

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](#) or [LDK client configuration](#).
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 IDd 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.
- **LDK-SL server**
  You can use the browser interface to control access to the server.
- **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.

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 canot 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 sntlconfigsrve.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.

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.

Terminology

Find terms that are used in this documentation.

activation
> See USB hardware device activation.

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.

clientconfig
> A program (Linux® command) that can be used to change parameters in the client XML file.

component
> A collection of z/OS® assets, such as volumes, data sets, CICS® resources, or DB2® resources, from an existing z/OS, z/OS instance, or the Extended ADCD package.
> A component is reusable with different components to create more than one image.

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.

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).

Extended Application Developers Controlled Distribution (ADCD)
> ZD&T Enterprise Edition also comes with the Extended ADCD package. The Extended 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, and JES2. You can select the software packages that you need, and provision the packages out of Extended ADCD to the target environment. For more information about Extended ADCD, see Extended ADCD reference.

IBM License Key Server
> The license server used exclusively to share Rational® Token license entitlements among a set of IBM products.

IBM License Server
> In zPDT, provides controlled access to multiple IBM software products and might be used along with zPDT license servers. The IBM License Server has no relation to zPDT license servers.

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 is used to provision z/OS instances on target environments.

instance
> A z/OS instance that is running on a target environment from a created image.

instance number
> A number in the range of 1 and 255 assigned to each z/OS instance on a base Linux machine. Each z/OS 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 z Systems.

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 IBM 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 IBM 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 IBM 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 license.

product license server
See license server.

provisioning tools
Refers to a web UI with role-based access control and REST APIs. The provisioning tools ease the process of provisioning and managing your z/OS instances. 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 z/OS instances of selected images to target environments.
- Manage and monitor components, images, and instances (also known as ZD&T emulator instances).

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.

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 documentation.

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.

software-based license server update file
The specific license key file that is generated in the IBM 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 z/OS on IBM Z physical hardware, Linux systems that run a z/OS instance, and Extended ADCD.

storage server
ZD&T Enterprise Edition stores extracted information on the intermediary storage machine, for example, 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 prepare such a storage server and ensure enough storage on the storage server to hold the required artifacts.

target environment
An Intel x86-compatible environment that is capable of running one or more z/OS instances.
Several types of target environments are supported, including Linux, Docker, and Red Hat® OpenShift®.

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 that the last recorded time in the token, the license is unusable.

token
See USB hardware device.
A type of IBM Rational product entitlement that allows great flexibility in the deployment and use of associated, purchased products. The product documentation uses the phrase Rational Token to refer to IBM Rational Token licensing.
In zPDT, another term for a SafeNet Sentinel Key. The terms token, key, SafeNet key, and Sentinel key are used interchangeably.

token serial number
The license information in the token contains a unique serial number that is assigned by IBM. This serial number might be used as the basis for the z Systems CP serial number in some cases.

UIM 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 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 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 IBM License Key Center and applied to the USB hardware device to enable zPDT 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.

web server

A server that enables [ZD&T provisioning tools](#). It is a component of ZD&T Enterprise Edition.

The web server can be installed on the same system as the hardware-based license server or as the software-based license server.

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](#).