IBM Virtual Dev and Test for z/OS 1.4.x

Tables of Contents

IBM® Virtual Dev and Test for z/OS®

This is the welcome page of the documentation.

What's new in ZVDT

Learn the new features and enhancements in ZVDT version 1.4.0 and all subsequent 1.4.x fix packs.

1.4.2

- This version is a fix pack that contains security fixes and documentation improvements. To download the fix pack, go to Fix Central.

Fixed APARs

APAR	Description
PH59689	ZD&T 14.x: Provision fails with error code 11102 when extracted image contains more than 256 communication devices
PH59976	ZD&T 14.X: LICENSE SERVER TAKES TOO LONG TO RELEASE LICENSES AFTER AN INSTANCE IS STOPPED VIA AWSSTOP
PH57773	ZD&T V14 LIMITED TO 2047 DEVICES
PH60103	ZD&T 14.X NEED TO CONTAIN UP TO DATE REST API CALLS DESCRIPTION

1.4.1

- This version is a fix pack that contains security fixes and documentation improvements. To download the fix pack, go to Fix Central.

Fixed APARs

APAR	Description
PH58138	ZD&T 13.3+: Documentation needs update to include newly included ADCD IPL script (zdt_config_adcd_ipl.sh).

1.4.0

- You can now install the web server and the license server on the Ubuntu distribution of Linux® on IBM Z®. For instructions on how to download the installers, see Downloading installation packages.
- Free z/OS® Integrated Information Processor (zIIP) is now supported. For more information about the license consumption of processors, see Product overview.
- 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 ZVDT.

Fixed APARs

APAR	Description
PH55992	ZVDT 1.x: LPARNAME (XX) filter used in IEASYMXX members will not resolve correctly to the specified value of SYSTEM_NAME.

Product overview

IBM® Virtual Dev and Test for z/OS® (ZVDT) enables enterprises to run current, genuine z/OS software and middleware on Linux® on IBM Z®. It is developed in lock step with IBM Z hardware, offering unmatched application portability and compatibility. ZVDT can eliminate typical development bottlenecks by enabling flexible, horizontal scaling of early development, test, and education activities.

ZVDT features

z/OS early development, test, and education on Linux on IBM Z, with full IBM Z instruction sets and virtualized I/O
: ZVDT provides support to full IBM Z instruction sets and uses virtualized I/O and other devices, enabling z/OS operating system, middleware and other z/OS software to run unaltered on Linux on on-premise IBM Z environments, for early development, test, and education purposes.

Web-based interface for self service build, deployment, and provision
: ZVDT offers a web-based interface to build and automate the deployment and provision of additional z/OS development and test environments on Linux on IBM Z. It is a role-based offering that empowers system programmers to create z/OS components and images more quickly, with the security and control they demand. Developers and testers can auto-deploy and auto-provision these official images for development and test activities with just a few clicks, giving them the autonomy and access that they need, using hardware that can easily be virtualized, cloned and reset.

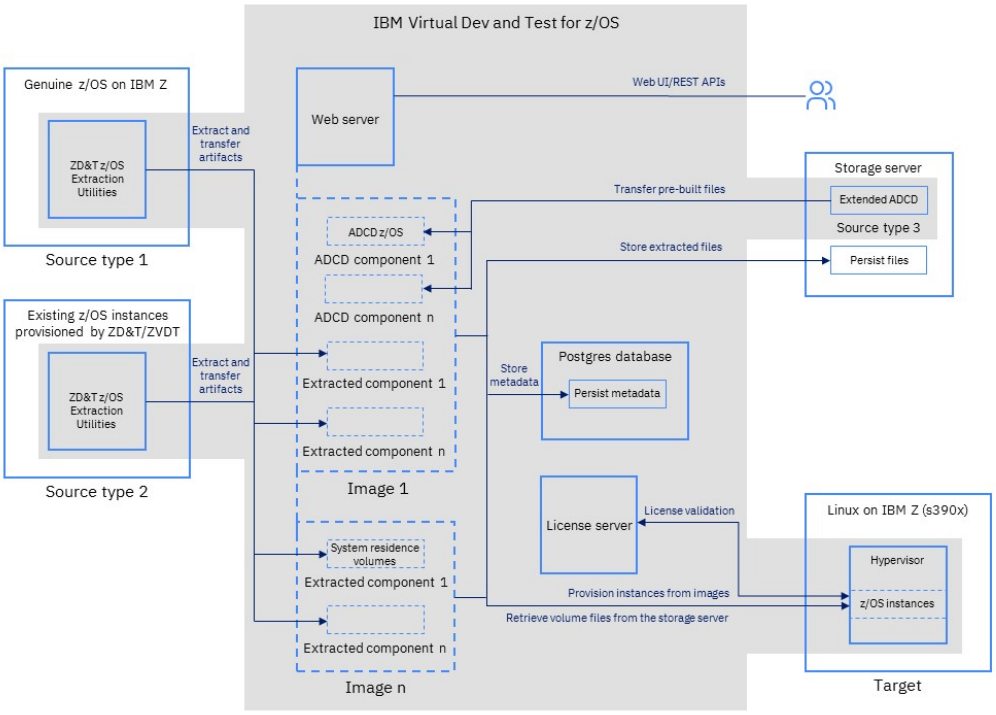
RESTful APIs for DevOps pipeline integration
: ZVDT also comes with powerful RESTful APIs, providing automation to enable the process of creation, deployment, and provision of z/OS environments. With RESTful APIs, enterprises can integrate ZVDT functionality into DevOps pipelines.

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 ZVDT hypervisor must have equal or more general purpose processors than zIIPs.

- **Architecture diagram**
- **End-to-end workflow**
- **Use cases**
  Learn about three typical use cases of ZVDT.
- **Understanding and determining when the ZVDT hypervisor is the z/OS host**

Architecture diagram



Source environments

ZVDT 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)
: ZVDT 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 ZVDT to use the IBM Z platforms as source environments to extract z/OS, CICS, and Db2 resources. For more information, see Optional: Installing z/OS Extraction Utilities.

Target environments

A target environment is a Linux on IBM Z (s390x) environment that is capable of running the ZVDT hypervisor, which can run multiple z/OS instances.

Hypervisor

The ZVDT hypervisor provides the ability to run unaltered z/OS software on Linux on IBM Z. Manual system configurations are required on the target environment. You need to install the hypervisor and configure how each instance will handle the networking. For installation instructions, see Installing and configuring the hypervisor.

ZVDT enhances the Linux on IBM Z Kernel Virtual Machine hypervisor to allow it to host virtual machine runtime environments for multiple z/OS guest operating systems at the same time. The ZVDT hypervisor identifies itself as the ZVDT host control program to the guest z/OS.

IBM® z/Architecture®, as described in *z/Architecture Principles of Operation*, provides a means for the guest operating system to determine the available hardware facilities, including both actual and virtualized hardware. The hardware information can be obtained through the Store System Information (STSI), Store Facility List (STFL), Store Facility List Extended (STFLE), Store CPU ID (STIDP) instructions, or any related interfaces of the hypervisor and the guest operating system. However, some hardware facilities or features might not be available to the guest operating system because the host control program chose not to support them. In addition, the host might hide or alter certain aspects of the underlying actual hardware. For more information about the hardware environments and related commands that z/OS supports, see the z/OS documentation .

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.

For more information, see Setting up the web server.

Storage server

ZVDT 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 ZVDT.

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 ZVDT 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 ZVDT, license activation is not required.

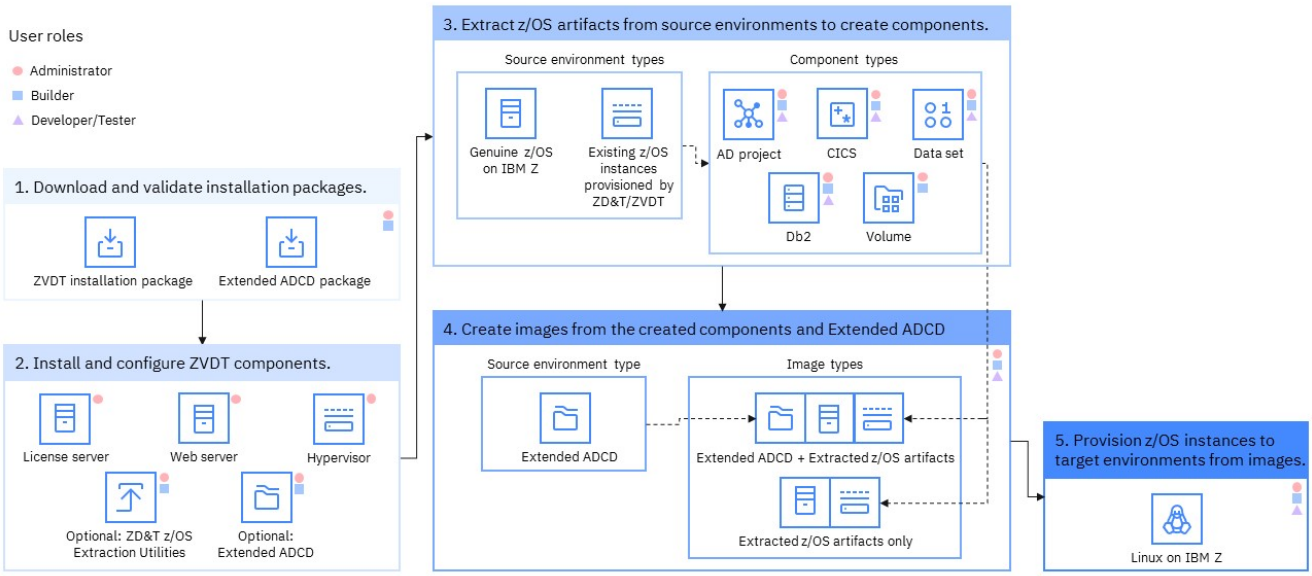
Each license key file is generated and is entitled to a number of emulated central processors. Central processors are also known as CPs. A single instance of ZVDT requires at least 1 CP, and can run with a maximum of eight CPs. The number of CPs needed depends on the number of users and the types and amount of processing required.

ZVDT also comes with a multi-user license that is known as the Resource Value Unit (RVU).

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.

End-to-end workflow



User roles

The following three typical types of user roles are covered. You can set up or use ZVDT according to your role.

Administrator
: Someone that is knowledgeable in setting up servers, and would only set up ZVDT server components without the need of much knowledge of IBM Z.

Builder
: Someone that specializes in running IBM Z and z/OS (sysadmin), or in running z/OS applications (sysprog). Expertise in these areas helps ensure good quality of the created z/OS images.

Developer/Tester
: Someone that conducts z/OS application development or testing activities on target environments. z/OS instances can be provisioned from images or deprovisioned according to actual needs.

Workflow

1. Download and validate installation packages. (Administrator, Builder)
   The ZVDT installation package contains the installers. In addition to the ZVDT installation package, the Extended ADCD package is offered by ZVDT as an optional choice of z/OS artifacts, which needs to be downloaded separately. For more information, see Downloading installation packages.

2. Install and configure ZVDT components.
   Five product components are included in ZVDT, and they need to be set up on different machines accordingly:
   - License server (Administrator)
   - Web server (Administrator)
   - Hypervisor (Administrator)
   - Optional: ZD&T z/OS Extraction Utilities (Administrator, Builder)
   - Optional: Extended ADCD (Administrator, Builder)

   For more information about each component and how they interact with each other, see Architecture diagram. For more information about how to set up the components, see Installing and configuring.
3. Extract z/OS artifacts from source environments to create components.
   z/OS artifacts can be extracted from either genuine z/OS on IBM Z or existing z/OS instances provisioned by ZD&T or ZVDT, where ZD&T z/OS Extraction Utilities must be installed. Extracted z/OS artifacts are reusable as components. One component can be combined with different components to create z/OS images.
   Creations of the following types of components are supported:
   - AD project (Administrator, Builder, Developer/Tester)
     ZVDT 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.

   - CICS (Administrator, Builder, Developer/Tester)
   - Data set (Administrator, Builder, Developer/Tester)
   - Db2 (Administrator, Builder, Developer/Tester)
   - Volume (Administrator, Builder)

   For more information about how to create components through the web UI, see Creating components.
4. Create images from the created components and Extended ADCD. (Administrator, Builder, Developer/Tester)
   An image is a collection of z/OS components that are packaged together from source environments. In addition to the extracted components mentioned in the previous step, the Extended ADCD package is offered with per-built z/OS components as an optional choice. To use the Extended ADCD components, Extended ADCD must be configured on the storage server, which is not part of ZVDT and needs to be prepared set up by yourself. 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 or application data from source environments. Creations of the following types of images are supported:
   - Combination of extracted z/OS components and Extended ADCD components
   - Extracted z/OS components only

   For more information about creating images through the web UI, see Creating images.
5. Provision z/OS instances to target environments from images. (Administrator, Builder, Developer/Tester)
   With the created images, z/OS instances can be automatically deployed and provisioned to Linux on IBM Z target environments, where the hypervisor is installed. Multiple z/OS instances are supported on a single target environment. For more information about provisioning through the web UI, see Provisioning instances to target environments.

Use cases

Learn about three typical use cases of ZVDT.

Use case 1: Isolated and on-premises development environments

z/OS system administrators or programmers: Define multiple components based on departments or projects, and keep them ready for developers to use on-demand, without the need of refreshing volumes. Create images from components for developers, and the latest changes of volumes are automatically picked up.

z/OS developers: Pick the predefined images to provision a z/OS instance on a Linux on IBM Z environment. Use the provisioned instance, which is an isolated and on-premises z/OS environment, where the z/OS operating system, middleware, and z/OS software can run unaltered with the full IBM Z instruction sets, to conduct early development activities. For example, this private environment can be used as a sandbox for prototyping and version-to-version migration.

Use case 2: Isolated and on-premises test environments

z/OS system administrators or programmers: Define multiple components based on departments or projects, and keep them ready for testers to use on-demand, without the need of refreshing volumes.

z/OS testers: Picks the predefined components to create images, and then provision a z/OS instance on a Linux on IBM Z environment. Use the provisioned instance, which is an isolated and on-premises z/OS environment with the same high performance, to conduct testing activities, for examples, z/OS software beta or version upgrade testing. After the testing is completed, deprovision the private instance.

Use case 3: Education environments for new z/OS developers

z/OS system administrators or programmers: Set up z/OS instances on a Linux on IBM Z environment as education environments for z/OS learners.

New z/OS developers: Use the education environments as a sandbox to learn z/OS, middleware, and DevOps tools without barriers.

Understanding and determining when the ZVDT hypervisor is the z/OS host

The ZVDT hypervisor presents a Store CPU ID (STIDP) environment code that indicates an alternate level 3 configuration and uses the Store System Information (STSI) instruction to provide three levels of the machine configuration information, including the central processor complex (CPC), the logical partition (LPAR), and the virtual machine (VM) in which the guest operating system is running. Under the virtual machine environment, the STSI instruction returns a System-Information Block (SYSIB) 3.2.2 with the VM Control-Program Identifier, which can be used to identify that the ZVDT hypervisor is the host control program. Both the STIDP environment code and the STSI instruction must be used together to determine that the ZVDT hypervisor is the host.

STIDP instruction

The ZVDT hypervisor returns STIDP information with the following qualifications:

- The environmental code is set to `x'FD'` to indicate an alternate level 3 configuration.

- The Configuration Identification is not altered from that presented by a level-2 configuration.

STSI instruction

The ZVDT hypervisor returns STSI information with the following qualifications:

- As the ZVDT hypervisor supports running on Integrated Facility for Linux (IFL) processors, the processors presented to the guest z/OS are virtualized General Processors and not IFL processors.
- STSI VM SYSIB 3.2.2
  - The first token of the Control-Program Identifier is set to Extended Binary-Coded Decimal Interchange Code (EBCDIC) `ZVDT`, and it ends at the first blank. Extra identification information might or might not be provided after the first token. The first token is sufficient to determine whether the ZVDT hypervisor is the host or not.
  - The EBCDIC Virtual-Machine Name is set to non-unique value "KVMguest". The EBCDIC VM UUID and Extended VM Virtual-Machine name are not set and thus not provided.

For more information about STIDP or STSI, see *z/Architecture Principles of Operation* and the z/OS documentation .

Getting started

To get started rapidly, refer to the following recommended role-based roadmap.

For more information about user roles and workflow, see End-to-end workflow.

Server administrator	z/OS system administrator or programmer	z/OS developer or tester
1. Downloading installation packages 2. Validating the signed installers 3. Setting up the web server 4. Installing and configuring the hypervisor 5. Setting up the software-based license server 6. Configuring on the web server	1. Optional: Configuring Extended ADCD 2. Optional: Installing z/OS Extraction Utilities 3. Adding source environments 4. Adding Linux on IBM Z target environments 5. Creating components 6. Creating images	1. Provisioning instances to target environments 2. Monitoring and managing the provisioned instances

Prerequisites

Learn about the ZVDT prerequisites.

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

Hardware and software requirements

Learn about hardware and software requirements for ZVDT.

For a complete list of hardware and software requirements, you can generate the report from Software Product Compatibility Reports.

For other prerequisites, see the following details.

- Storage server requirements
- Source environments
  - z/OS system requirements
  - Db2 extraction requirements
- Target environments: Linux on IBM Z requirements

Storage server requirements

To install and run ZVDT, a storage server to host the ZVDT artifacts, such as z system volumes, data sets, and ZVDT 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 ZVDT.
  - The SFTP client must be able to connect to the ZVDT 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 ZVDT.
  - 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 on IBM Z requirements

- ZVDT is supported on IBM Linux on Z, only when it is installed directly onto an LPAR on IBM z16™, IBM z15®, or IBM LinuxONE III hardware. Currently supported Linux distributions are Ubuntu 22.04, Red Hat® Enterprise Linux (RHEL) 9.1, and Red Hat Enterprise Linux 8.7.
- The minimum kernel level on a target Ubuntu 22.04 system is 5.4.0-109-generic.
- libnbd, perl-FindBin, and perl-lib must be installed on a Red Hat Enterprise Linux 9.1 target environment with the following commands for the hypervisor to function properly.

  ```
  yum install libnbd
  yum install perl-FindBin
  yum install perl-lib
  ```

- sha256sum must be installed. It can be installed with package coreutils.
- It is recommended to not over-commit memory on the hosting Linux environment, as out of memory conditions on Linux can result in all virtualized z/OS instances being forcibly terminated.
- The hypervisor must be installed on a different machine than the web server, storage server, and license server.
- Users running a z/OS instance via ZVDT must be a part of the kvm Linux group. For example, if a z/OS instance is to be run under the ibmsys1 Linux user account, user ibmsys1 must be added to group kvm before starting the instance. To add a user to group kvm and leave it in the supplementary group that it is already a member of, the following Linux command can be used:

  ```
  usermod -a -G kvm <user_name>
  ```

- Network configuration
    - To use MacVTap interfaces as the networking topology, a set of IP addresses must be reserved and referenced when you add a Linux on IBM Z target environment on the web server. In addition, a set of MacVTap adapters must be manually created against a physical adapter on the Linux on IBM Z system, and referenced when you add a target environment on the web server. Each z/OS instance requires one MacVTap adapter, which allows the instance to have a dedicated IP address and the full set of ports for use. Before you create MacVTap adapters, check MacVTap driver considerations. For more information about creating a MacVTap adapter, go to https://manpages.ubuntu.com/manpages/jammy/man8/ip-link.8.html.

Example

```
ip link add link physical_adapter_name macvtap_name type macvtap mode
bridge
ip link set macvtap_name up
```

To avoid any potential Address Resolution Protocol (ARP) flux issues that might randomly occur, it is recommended that at least two dedicated Open Systems Adapters (OSA) are available to the Linux environment. The first one can remain dedicated to Linux, while the others are for linking the MacVTap adapters that are used by the virtualized z/OS instances.

- To use iptables as the networking topology, you must route Linux port 2022 to port 22 on the provisioned z/OS to ensure that other systems can communicate with it. The port number to be routed is the one that you will specify when you configure the source system on the 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`.

- **Storage server requirements**
- **z/OS system requirements**
- **Db2 extraction requirements**
- **Linux on IBM Z requirements**

Required firewall ports for network flow

Before you install and use , 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

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

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 hypervisor. Make sure the right port number is specified on the web server UI.
2022	z/OS UNIX System Services user	SSH	TCP	For communication from a z/OS user to the z/OS system on the hypervisor 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 hypervisor.	No

Note: For other z/OS ports used for custom purposes, manual firewall rule configuration is required.

Limitations

Before you use ZVDT, make sure that you are aware of the following known limitations.

Creation support

- ZVDT 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.

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 Development and Test Environment.

- 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 ZVDT. To download the ZVDT V1.4.0 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. ZVDT V1.4.0

	Name	Part No.	Image Classification	File Name
1	IBM Virtual Dev and Test for z/OS 1.4 Installation Multilingual eAssembly	G0989ML	Required	N/A
2	IBM Virtual Dev and Test for z/OS 1.4 Installation Multilingual for x86	M0FDNML	Required	zvdt-install-x86.tgz
3	IBM Virtual Dev and Test for z/OS 1.4 Installer checksum Multilingual for x86	M0FDPML	Required	zvdt-x86.md5
4	IBM Virtual Dev and Test for z/OS 1.x Quick Start Guide	M083NML	Required	ZVDT_Quick_Start_Guide_1.x.pdf
5	IBM Virtual Dev and Test for z/OS 1.4 Installation Multilingual for s390x (Ubuntu only)	M0FLZML	Required	zvdt-install-s390x.tgz
6	IBM Virtual Dev and Test for z/OS 1.4 Installer checksum Multilingual for s390x (Ubuntu only)	M0FM0ML	Required	zvdt-s390x.md5

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

	Name	Part No.	Image Classification	File Name

	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

	Name	Part No.	Image Classification	File Name
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_RSU2309
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® Virtual Dev and Test for z/OS® 1.4 RSU 2303 Multilingual eAssembly	G098BML	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

	Name	Part No.	Image Classification	File Name
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

	Name	Part No.	Image Classification	File Name
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_RSU2303
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 ZVDT installation package contains the following signed installers and the corresponding signature files.

Installer	Signature file
Web server installer zvdt-install for Linux on x86 or zvdt-install-s390x for Linux on IBM Z (Ubuntu only)	zvdt-install.cosign.sig for Linux on x86 or zvdt-install-s390x.cosign.sig for Linux on IBM Z (Ubuntu only)
License server installer zdt-license-server for Linux on x86 or zdt-license-server-s390x for Linux on IBM Z (Ubuntu only)	zdt-license-server.cosign.sig for Linux on x86 or zdt-license-server-s390x.cosign.sig for Linux on IBM Z (Ubuntu only)

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 zvdt-install.cosign.sig
zvdt-install
```

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

Upgrading

Learn how to upgrade ZVDT.

If you have installed an earlier version of ZVDT, follow the steps the upgrade:

1. Upgrade the web server.
    a. Upgrading the web server
    b. Configuring the web server
    c. (Optional) Starting and stopping the web server
2. Upgrade the license server.
    a. Returning a software-based license key
    b. Upgrading the software-based license server
    c. Obtaining an update file
    d. Applying the update file to the software-based license server machine

Installing and configuring

Learn how to install and configure ZVDT components.

- **Setting up the web server**
  ZVDT 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.
- **Installing and configuring the hypervisor**
  Learn how to set up the hypervisor on a target environment.
- **Setting up the software-based license server**
  To operate ZVDT, 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

ZVDT 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 ZVDT web server, run the installer with the root user ID.
- **Installing a signed certificate on the web server**
  To access to the ZVDT 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 ZVDT web server, ensure that you configure the authentication for ZVDT.
- **Starting and stopping the web server**
  When you installed the ZVDT 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.

1. Open the directory where the installation package is stored.

   ```
   cd <directory>
   ```

   Package for Linux on x86
   zvdt-install-x86.tgz
   Package for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x.tgz

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 for Linux on x86
   zvdt-install
   Web server installer for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x

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 --zvdt
```

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

     ```
     rpm -qa | grep zvdtapp
     ```

   - Ubuntu operating system

     ```
     dpkg -l | grep zvdtapp
     ```

   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 ZVDT web server, run the installer with the root user ID.

1. Open the directory where the installation package is stored.

   ```
   cd <directory>
   ```

   Package for Linux on x86
   zvdt-install-x86.tgz
   Package for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x.tgz

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 for Linux on x86
   zvdt-install
   Web server installer for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x

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

   ```
   ./<installer_name>
   ```

   Standard installation
   This option will install ZVDT with the following default settings:

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

Default user ID for logging into the web server
zvdtadmin
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 --zvdt --zvdt_port=<port_number> --zvdt_user=
<username> --zvdt_path=<installation_path> --zvdt_password=<password>
```

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

     ```
     rpm -qa | grep zvdtapp
     ```

   - Ubuntu operating system

     ```
     dpkg -l | grep zvdtapp
     ```

   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 ZVDT 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 (zvdtkey.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/zvdt is the default installation directory, but you can specify your own installation directory during the installation process.
2. Run the following command to generate zvdtkey.p12 and put it in the *<installation_directory>*/zvdt-server/resources/security.

   ```
   openssl pkcs12 -export -out zvdtkey.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>*/zvdt-server/bootstrap.properties; for example, /opt/ibm/zvdt/zvdt-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/zvdt-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 ZVDT web server, ensure that you configure the authentication for ZVDT.

- **Authentication for the ZVDT application**
  Before you use the ZVDT web server, you can modify how authentication is accomplished by ZVDT.
- **Auditing the web server**
  Learn about the steps to enable audit log for ZVDT.

Authentication for the ZVDT application

Before you use the ZVDT web server, you can modify how authentication is accomplished by ZVDT.

- **Default authentication and authorization**
  ZVDT comes with a default user ID and password. You can also change the default password.
- **Basic authentication**
  By default, ZVDT uses a basic registry as the user registry for authentication and provides a default user ID 'zvdtadmin'. Also, you can add more users to this registry.
- **Lightweight Directory Access Protocol (LDAP)**
  Learn about the steps to enable LDAP authentication for ZVDT.

Default authentication and authorization

ZVDT comes with a default user ID and password. You can also change the default password.

The default user ID 'zvdtadmin' is defined in file *<installation_directory>*/Liberty/usr/servers/zvdt-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/zvdt-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/zvdt-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_SSTAXM7_1.4.0_com.ibm.zsys.rdt
   .tools.user.guide.doc_topics_default_ps_zvdt" realm="zvdt">
   <user name="zvdtadmin"
   password="*{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB*"/>
   </basicRegistry>
   ```

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

Basic authentication

By default, ZVDT uses a basic registry as the user registry for authentication and provides a default user ID 'zvdtadmin'. 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/zvdt-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/zvdt-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_SSTAXM7_1.4.0_com.ibm.zsys.rdt
   .tools.user.guide.doc_topics_new_user_zvdt" realm="zvdt">
           <user name="zvdtadmin" 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 ZVDT.

The Lightweight Directory Access Protocol (LDAP) settings for the server are defined in the *<installation_directory>*/Liberty/usr/servers/zvdt-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/zvdt-server/resources/security/zvdttrust.p12. The password for the zvdttrust.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 ZVDT.

The audit log for the web server is defined in the *<installation_directory>*/Liberty/usr/servers/zvdt-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 ZVDT 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 *ZVDT_USER* in the *<installation_directory>*/bin/zvdt_env.sh script file. You also need to change the file ownership of all the files in the directory *<installation_directory>*/Liberty/usr/servers/zvdt-server/ to the user ID, for example, **chmod -R newuser:root*<installation_directory>*/Liberty/usr/servers/zvdt-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_ZVDT_CMD* in the *<installation_directory>*/bin/zvdt_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 (zvdtadmin) 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
```

Installing and configuring the hypervisor

Learn how to set up the hypervisor on a target environment.

Make sure that the prerequisites are met. For more information, see Linux on IBM Z requirements.
The hypervisor must be installed and configured on the Linux on IBM Z system before this machine can be used as a target environment.

1. After the instillation of the web server is completed, you can find the hypervisor installer. Copy the installer to your target environment system.

   For Ubuntu
   <*installation_directory*>/zvdt/zSystem/z1091_2.2.M2.19_s390x.deb
   For RHEL
   <installation_directory>/zvdt/zSystem/z1091_2.2.M2.19_s390x.rpm

2. Install the hypervisor by running the following command. For dependencies, see Linux on IBM Z requirements.

   On Ubuntu

   ```
   dpkg -i <path_to_file>/z1091_2.2.M2.19_s390x.deb
   ```

   On RHEL

   ```
   rpm -i <path_to_file>/z1091_2.2.M2.19_s390x.rpm
   ```

3. Modify environment variables for root. Add the variables to ~/.bashrc.

   ```
   export PATH=/usr/z1090/bin:$PATH;
   export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH;
   ```

4. Configure the license server by running this command and following the prompt.

   ```
   /usr/z1090/bin/clientconfig
   ```

5. Create a non-root user ID, for example, ibmsys1, and then assign it to a KVM group.

   ```
   useradd -m ibmsys1 -g kvm -d /home/ibmsys1
   ```

6. Modify environment variables for the non-root user ID created in the previous step. Add the variables to ~/.bashrc.

   ```
   export PATH=/usr/z1090/bin:$PATH;
   export LD_LIBRARY_PATH=/usr/z1090/bin:$LD_LIBRARY_PATH;
   ```

Setting up the software-based license server

To operate ZVDT, 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 ZVDT.

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 ZVDT, 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 is stored.

   ```
   cd <directory>
   ```

   Package for Linux® on x86
   zvdt-install-x86.tgz
   Package for Linux on IBM Z® (Ubuntu only)
   zvdt-install-s390x.tgz

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 for Linux on x86
   zdt-license-server
   License server installer for Linux on IBM Z (Ubuntu only)
   zdt-license-server-s390x

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
```

Ubuntu Linux on IBM Z

```
dpkg -l | grep zdtlicense
```

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 is stored.

   ```
   cd <directory>
   ```

   Package for Linux on x86
   zvdt-install-x86.tgz
   Package for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x.tgz

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 for Linux on x86
   zdt-license-server
   License server installer for Linux on IBM Z (Ubuntu only)
   zdt-license-server-s390x

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
   ```

   Ubuntu Linux on IBM Z

   ```
   dpkg -l | grep zdtlicense
   ```

   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 ZVDT 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 the software-based license server**
  Learn about the steps to obtain an update file for the software-based license server.

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).

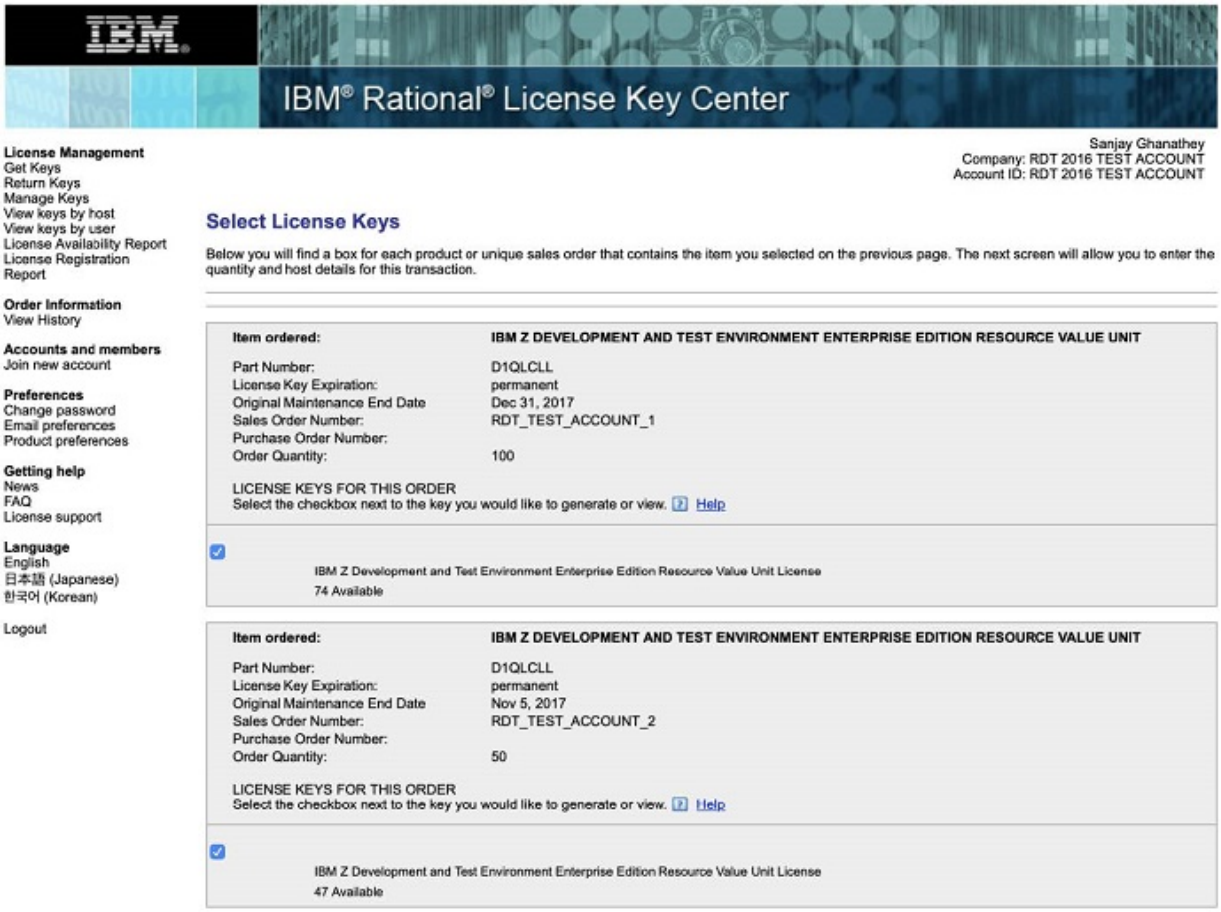
```
./uimserverstart
```

Combining all the available licenses on a single license server

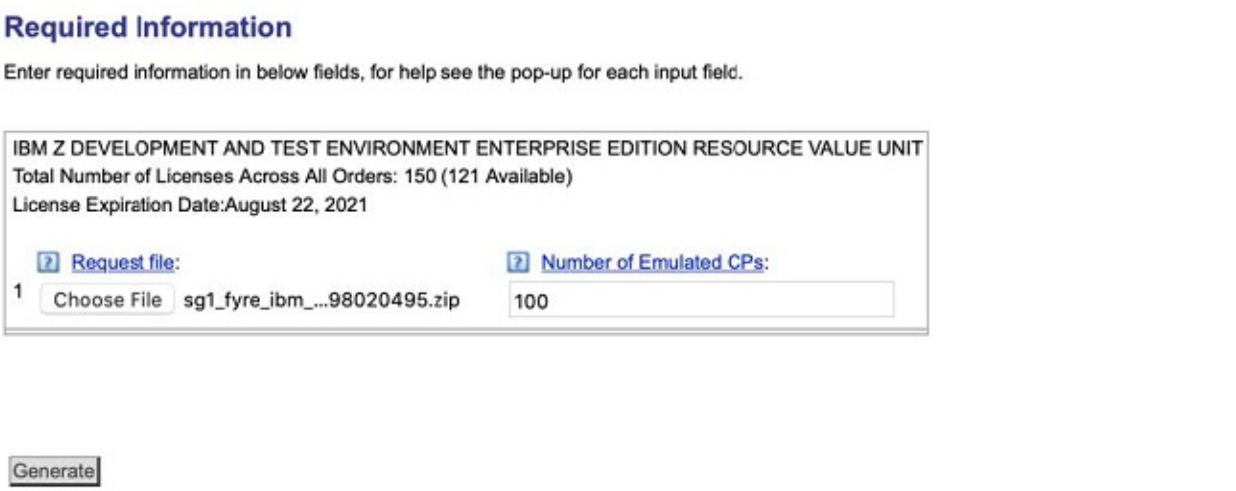
If you have multiple licenses available for ZVDT, you can combine all the available licenses on a single license server.

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.



3. Download the generated update file.

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: 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 ZVDT, 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 ZVDT.

   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 V2R5 May Edition of 2023, hints and tips, and the customized ADCD package (Extended ADCD) for ZVDT.

Extended ADCD reference

Learn about general release information of ADCD z/OS® V2R5 May Edition of 2023, hints and tips, and the customized ADCD package (Extended ADCD) for ZVDT.

- **ADCD z/OS V2R5 May Edition of 2023**
- **Extended ADCD for ZVDT**
  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 V2R5 May Edition of 2023

Learn what's new in ADCD z/OS V2R5 May 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 V2R5 May Edition of 2023

NOTE: As of ADCD z/OS V2R5 May 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 V2R5 May 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 V1.1.9
- IBM Z Multi-Factor Authentication V2.2.0
- IBM Application Delivery Foundation for z/OS V4.0.0
- 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 V7.3.0

The following products have been updated to PUT2303 / RSU2303

- z/OS 2.5 and all incorporated products that provided maintenance at the above stated level.

All of z/OS base, z/OS products and middleware volume size has been increased to mod-9 (10,017 cylinders) to provide more free space for expansion

Back to Table of Contents

Products contained in this release

Table_01 z/OS 2.5 May 2023 Edition

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-MQ9	IBM MQ FOR Z/OS CD V9.2.0	HMS9CD0	JMS9CD1	SYS1
5650-ZOS	Hardware Configuration Manager	HCM1I10	HCM1I10	SYS1
5650-ZOS	GDDM PGF	HGD3201	HGD3201	SYS1
5650-ZOS	XL C/C++ Base	HLB77C0	HLB77C0	SYS1
5650-ZOS	XL C/C++ Base	HLB77C1	HLB77C1	SYS1
5650-ZOS	IBM Open XL C/C++	HLB77C2	HLB77C2	SYS1
5650-ZOS	Infoprint Server - IP PrintWay Base	HMOS705	HMOS705	SYS1
5650-ZOS	Infoprint Server - NetSpool Base	HNET7D0	HNET7D0	SYS1
5650-ZOS	Infoprint Server - Print Interface Base	HOPI7D0	HOPI7D0	SYS1
5650-ZOS	SDSF Base	HQX77D0	HQX77D0	SYS1
5650-ZOS	Security Server - RACF Base	HRF77D0	HRF77D0	SYS1
5650-ZOS	RMF	HRM77D0	HRM77D0	SYS1
5650-ZOS	DFSORT Base	HSM1Q00	HSM1Q00	SYS1
5650-ZOS	C/C++ HOST PERFORMANCE ANALYZE	H24P111	H24P111	SYS1
5650-ZOS	BULK DATA TRANSFER FILE-TO-FIL	HBD6602	JBD6201	SYS1
5650-ZOS	BULK DATA TRANSFER SNA NJE	HBD6602	JBD6202	SYS1
5650-ZOS	High Level Assembler Toolkit	HMQ4160	JMQ416A	SYS1
5650-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
5650-ZOS	ICKDSF - Device Support Facilities, Base	EDU1H01	EDU1H01	SYS1
5650-ZOS	Environmental Record Editing and Printing	EER3500	EER3500	SYS1
5650-ZOS	MICR/OCR	EMI2220	EMI2220	SYS1
5650-ZOS	TIOC	ETI1106	ETI1106	SYS1
5650-ZOS	ICKDSF - Device Support Facilities, ISMF/MODS	EDU1H01	FDU1H07	SYS1
5650-ZOS	ICKDSF - Device Support Facilities, ISMF/ENU	EDU1H01	FDU1H08	SYS1
5650-ZOS	BCP Base	HBB77D0	HBB77D0	SYS1
5650-ZOS	BULK DATA TRANSFER BASE	HBD6602	HBD6602	SYS1
5650-ZOS	Cryptographic Services - System SSL Base	HCPT450	HCPT450	SYS1
5650-ZOS	Cryptographic Support - ICSF	HCR77D2	HCR77D2	SYS1
5650-ZOS	Hardware Configuration Definition Base	HCS77D0	HCS77D0	SYS1
5650-ZOS	Network File System Server and Client	HDZ225N	HDZ225N	SYS1

Product Number	Description	Fmid	Dependent FMID	HLQ
5650-ZOS	Data Facility System Managed Storage Base & ENU	HDZ2250	HDZ2250	SYS1
5650-ZOS	z/OS Font Collection	HFNT140	HFNT140	SYS1
5650-ZOS	FFST	HFST101	HFST101	SYS1
5650-ZOS	PC 3270 FILE TRANSFER	HFX1112	HFX1112	SYS1
5650-ZOS	GDDM BASE	HGD3200	HGD3200	SYS1
5650-ZOS	IBM HTTP Server	HHAP90P	HHAP90P	SYS1
5650-ZOS	ISPF Base	HIF7T02	HIF7T02	SYS1
5650-ZOS	IOCP	HIO1105	HIO1105	SYS1
5650-ZOS	Communications Server IP	HIP6250	HIP6250	SYS1
5650-ZOS	JES2 Base	HJE77D0	HJE77D0	SYS1
5650-ZOS	IBM KNOWLEDGE CENTER FOR Z/OS	HKCZ120	HKCZ120	SYS1
5650-ZOS	Cryptographic Services - PKI Services	HKY77D0	HKY77D0	SYS1
5650-ZOS	Language Environment Base	HLE77D0	HLE77D0	SYS1
5650-ZOS	SMP/E Base	HMP1K00	HMP1K00	SYS1
5650-ZOS	High Level Assembler Base	HMQ4160	HMQ4160	SYS1
5650-ZOS	OpenSSH for z/OS	HOS2240	HOS2240	SYS1
5650-ZOS	z/OS UNIX System Services Application Services Bas	HOT77C0	HOT77C0	SYS1
5650-ZOS	CIM - Common Interface Model	HPG77C0	HPG77C0	SYS1
5650-ZOS	BCP - Program Management Binder	HPM77D0	HPM77D0	SYS1
5650-ZOS	BCP - Capacity Provisioning	HPV77D0	HPV77D0	SYS1
5650-ZOS	zOS Data Gatherer	HRG77D0	HRG77D0	SYS1
5650-ZOS	IBM Tivoli Directory Server for z/OS Base	HRSL440	HRSL440	SYS1
5650-ZOS	Metal C Runtime Library	HSD7780	HSD7780	SYS1
5650-ZOS	Z/OSMF Network Configuration Assistant	HSMA25A	HSMA25A	SYS1
5650-ZOS	z/OSMF zERT Network Analyzer	HSMA25E	HSMA25E	SYS1
5650-ZOS	z/OSMF Core Functions	HSMA250	HSMA250	SYS1
5650-ZOS	z/OSMF ISPF	HSMA251	HSMA251	SYS1
5650-ZOS	IBM z/OS Management Facility - RM	HSMA252	HSMA252	SYS1
5650-ZOS	z/OSMF WLM	HSMA253	HSMA253	SYS1
5650-ZOS	z/OSMF Software Deployment	HSMA254	HSMA254	SYS1
5650-ZOS	z/OSMF Incident Log	HSMA255	HSMA255	SYS1
5650-ZOS	z/OSMF Capacity Provisioning	HSMA256	HSMA256	SYS1
5650-ZOS	z/OSMF Workflow	HSMA257	HSMA257	SYS1
5650-ZOS	ESCON DIRECTOR	HSWF100	HSWF100	SYS1
5650-ZOS	Integrated Security Services-Network Authenticatio	HSWK450	HSWK450	SYS1
5650-ZOS	TSO/E Base	HTE77D0	HTE77D0	SYS1
5650-ZOS	Runtime Library Extensions Base	HTV77C0	HTV77C0	SYS1
5650-ZOS	BCP - Support for Unicode Base	HUN77D0	HUN77D0	SYS1
5650-ZOS	Communications Server SNA	HVT6250	HVT6250	SYS1
5695-014	IBM Alternate Library for REXX on zSeries	HWJ9143	HWJ9143	SYS1

Product Number	Description	Fmid	Dependent FMID	HLQ
5650-ZOS	z/OS Liberty Embedded	HWLPEM0	HWLPEM0	SYS1
5650-ZOS	Web Toolkit	HWT0500	HWT0500	SYS1
5650-ZOS	HZAI250	HZAI250	HZAI250	SYS1
5650-ZOS	z/OS Container Extensions	HZDC7C0	HZDC7C0	SYS1
5650-ZOS	z/OS File System Base	HZFS450	HZFS450	SYS1
5650-ZOS	GDDM NLS ENU	HGD3200	JGD3219	SYS1
5650-ZOS	ISPF UpperCase English - ENP	HIF7T02	JIF7T16	SYS1
5650-ZOS	Communications Server X11R4 XWindows	HIP6250	JIP625X	SYS1
5650-ZOS	TSO/E ENU	HTE77D0	JTE77DE	SYS1
5650-ZOS	ICKDSF - DEVICE SUPPORT FACILITIES, ISMF/ENU JPN	EDU1H01	FDU1H09	SYS1
5650-ZOS	z/OS Font Collection - Chinese, Japanese, Korean	HFNT14J	HFNT14J	SYS1
5650-ZOS	Data Facility System Managed Storage JPN	HDZ2250	JDZ225K	SYS1
5650-ZOS	GDDM NLS JPN	HGD3200	JGD3227	SYS1
5650-ZOS	ISPF Japanese - JPN	HIF7T02	JIF7T14	SYS1
5650-ZOS	SMP/E JPN	HMP1K00	JMP1K11	SYS1
5695-014	IBM Alternate Library for REXX on zSeries -Japan	HWJ9143	JWJ9144	SYS1
5650-ZOS	IBM z/OS Change Tracker Base	HCYG100	HCYG100	SYS1
5650-ZOS	Communications Server Security Level 3	HIP6250	JIP625K	SYS1
5650-ZOS	JES3	HJS77D0	HJS77D0	SYS1
5650-ZOS	z/OS Security Level 3 - System SSL Security Level	HCPT450	JCPT451	SYS1
5650-ZOS	z/OS Security Level 3 - IBM TDS for z/OS Security	HRSL440	JRSL441	SYS1
5650-ZOS	z/OS Security Level 3 - Network Authentication Srv	HSWK450	JSWK451	SYS1

Table_02 z/OS 2.5 May 2023 Edition Products

Product Number	Description	Fmid	Dependent FMID	HLQ
5668-806	VS FORTRAN LIBRARY MVS	HFL2602	HFL2602	AFF260
5668-806	MVS IAD	HFR2602	HFR2602	AFF260
5668-806	COMPILER MVS	HFT2602	HFT2602	AFF260
5668-806	LIBRARY MVS DEP MODULES	HFL2602	JFL2611	AFF260
5668-806	MVS IAD (TSO)	HFR2602	JFR2611	AFF260
5668-806	MVS IAD (ISPF)	HFR2602	JFR2620	AFF260
5668-806	COMPILER MVS DEP MODULES	HFT2602	JFT2611	AFF260
5668-806	PUBS POST SCRIPT	HFT2602	JFT2612	AFF260
5668-806	PUBS BOOK MANAGER	HFT2602	JFT2613	AFF260
5648-B05	IBM COBOL and CICS Command Level Conversion Aid	H09F210	H09F210	ABJ210
5655-U59	HOURGLASS	HAD5710	HAD5710	AGG710
5655-M32	UPLOAD FOR z/OS	HPRF417	HPRF417	APS470

Product Number	Description	Fmid	Dependent FMID	HLQ
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	IBM Z Multi-Factor Authentication	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
5725-M54	UC DEPLOY FOR Z/OS	HRUC730	HRUC730	BUZ730
5655-TC1	IBM TOOLS CUSTOMIZER	HTCZ110	HTCZ110	CCQ110
5655-Y20	CICS Transaction Gateway for z/OS	HCTG930	HCTG930	CTG930
5655-GOZ	IBM Open Enterprise SDK for Go	HAMF1J0	HAMF1J0	CVG119
5655-PYT	IBM Open Enterprise SDK for Python	HAMB3B0	HAMB3B0	CYP3B0
5655-X11	IBM Sterling Connect:Direct for z/OS	HDGA620	HDGA620	DGA620
5655-X11	IBM STERLING CONNECT DIRECT FOR Z/OS STANDARD EDIT	HDGA620	JDGA620	DGA620
5655-103	DITTO/ESA base	H0GB310	H0GB310	DIT130
5655-Y24	CICS VSAM Recovery for z/OS - Base	HCCV520	HCCV520	DWW520
5655-Y24	CICS VSAM Recovery for z/OS - English	HCCV520	JCCV52E	DWW520
5698-AAR	IBM WORKLOAD SCHEDULER AGENT FOR Z/OS	HEEL940	HEEL940	EEL940
5655-R29	IBM Rational COBOL Runtime	HACZ601	HACZ601	ELA601
5755-A01	z/OS Debugger Base	HADRG00	HADRG00	ISM400
5755-A01	Application Performance Analyzer - Base Component	HAD0F10	HAD0F10	ISM400
5755-A01	IBM z/OS Source Code Analysis	HAKGG00	HAKGG00	ISM400
5755-A01	IBM Explorer for z/OS	HALG330	HALG330	ISM400
5755-A01	Rocket Git for z/OS	HAL2111	HAL2111	ISM400
5755-A01	Z Open Automation Utilities	HAL5120	HAL5120	ISM400
5755-A01	IBM z/OS Dynamic Test Runner	HAL6100	HAL6100	ISM400
5755-A01	IBM Remote System Explorer API (RSE API)	HAMA110	HAMA110	ISM400
5755-A01	IBM Dependency Based Build	HBGZ200	HBGZ200	ISM400
5755-A01	IBM z/OS Explorer Extensions	HHOPG00	HHOPG00	ISM400
5755-A01	Application Delivery Foundation Common Components	HVWR190	HVWR190	ISM400
5695-013	IBM Compiler for REXX on zSeries	HWK0140	HWK0140	FAN140
5695-014	IBM Library for REXX on zSeries	HWJ9140	HWJ9140	FAN140

Product Number	Description	Fmid	Dependent FMID	HLQ
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	REPORT MODERNIZATION UTILITY	HRMU110	HRMU110	FSY510
5655-WAS	IBM Installation Manager install kit	HGIN140	HGIN140	GIN143
5655-Y31	ODM FOR Z/OS	HBR8A00	HBR8A00	HBR8A0
5737-B16	IBM AD Connect for Mainframe	HALT611	HALT611	IAY611
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-J51	XML Toolkit for z/OS	HXML1B0	HXML1B0	IXM111
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
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_03 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_04 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

Product Number	Description	Fmid	Dependent FMID	HLQ
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_05 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_06 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

Product Number	Description	Fmid	Dependent FMID	HLQ
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
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
5655-CH1	IBM DB2 Change Management Solution Pack	HAQG120	HAQG120	ADBD10
5655-CH1	IBM Db2 Administration Tool for z/OS	H0IHD10	H0IHD10	ADBD10
5655-CH1	IBM Db2 Object Comparison Tool for z/OS	H25GD10	H25GD10	ADBD10
5655-CH1	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_07 IMS V15.1

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_08 IBM Z Monitoring Suite v1.3.0

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-B66	ITCAM for Application Diagnostics Common Services	HAAD71C	HAAD71C	ZMS130
5698-B66	ITCAM for Application Diagnostics	HAAD710	HAAD710	ZMS130
5698-B66	IBM z Monitoring Suite	HFZT121	HFZT121	ZMS130
5698-B66	IBM TIVOLI DISCOVERY LIBRARY ADAPTER FOR Z/OS	HIZD310	HIZD310	ZMS130
5698-B66	OMEGAMON for CICS on z/OS	HKC5550	HKC5550	ZMS130
5698-B66	OMEGAMON XE for DB2 PE on z/OS	HKDB54X	HKDB54X	ZMS130

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-B66	OMEGAMON XE for DB2 on z/OS	HKDB540	HKDB540	ZMS130
5698-B66	OMEGAMON for CICS TG on z/OS	HKGW550	HKGW550	ZMS130
5698-B66	OMEGAMON for IMS on z/OS	HKI5550	HKI5550	ZMS130
5698-B66	OMEGAMON for JVM MSU	HKJJ55U	HKJJ55U	ZMS130
5698-B66	OMEGAMON for JVM Base	HKJJ550	HKJJ550	ZMS130
5698-B66	OMEGAMON for IBM MQ Monitoring	HKMQ750	HKMQ750	ZMS130
5698-B66	OMEGAMON Data Provider	HKOA110	HKOA110	ZMS130
5698-B66	OMEGAMON Enhanced 3270 User Interface	HKOB750	HKOB750	ZMS130
5698-B66	OMEGAMON for IBM Integration Bus Monitoring	HKQI750	HKQI750	ZMS130
5698-B66	SHARED PROBES	HKSB750	HKSB750	ZMS130
5698-B66	OMEGAMON FOR STORAGE ON Z/OS	HKS3550	HKS3550	ZMS130
5698-B66	ITCAM for Application Diagnostics, TEMA	HKYN710	HKYN710	ZMS130
5698-B66	IBM InfoSphere Optim Client	HPMZ540	HPMZ540	ZMS130
5698-B66	OMEGAMON Integration Monitoring for DE	HRKD560	HRKD560	ZMS130
5698-B66	IBM Z OMEGAMON Network Monitor	HRKN560	HRKN560	ZMS130
5698-B66	IBM OMEGAMON Monitoring for z/OS	HRKZ560	HRKZ560	ZMS130
5698-A79	Installation/Configuration Assistance Tool	HKCI310	HKCI310	ZMS130
5698-A79	Tivoli Enterprise Monitoring Server	HKDS630	HKDS630	ZMS130
5698-A79	TMS:Engine	HKLV630	HKLV630	ZMS130

Table_09 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 BATCHPIPES FOR OS/390 2.1.0

Product Number	Description	Fmid	Dependent FMID	HLQ
5655-D45	BatchPipes for OS/390	HACH301	HACH301	SAS210

Table_12 IBM z Distribution ZOWE V2.0.0

Product Number	Description	Fmid	Dependent FMID	HLQ
5698-ZWG	IBM Z Distribution for Zowe	AZWE002	AZWE002	ZWE200

Reported Problems, Fixes, Maintenance and Observations

- zCX License Expiring awaiting fix see ADCD Download Update/Fixes on the ADCD Download Authentication page for additional information
- Issue installing 1090 z/PDT 1.7 GA Level - Install fails for the GA 1.7 level on a system running RH6.x. Please go to the z/PDT forum for additional information on the resolution.

Back to Table of Contents

Build Structure

Table_Base: - Minumnim Set of Volumes for Simple IPL. Use LoadParm: 0XyyzzM

Volume	Description
D5RES1, D5RES2	D5RES1 Base MVS system software target, or run-time libraries, and other system data sets required to use TSO/E and ISPF. D5RES2 An extension of the SYSRES volume **D5RES1 and D5RES2 both are required for IPL** D5RES1 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. D5RES2 is a 3390-9 volumes that are an extension of the SYSRES volume D5RES1. The volume is also required for IPL.
D5SYS1	System control data sets, such as SYS1.IPLPARM,IODF (I/O Definition File), and system's master catalog (CATALOG.Z25D.MASTER) reside on this volume. D5SYS1 contains the usercatalog USERCAT.Z25D.PRODS containing entries for z/OS relate products. **D5SYS1 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.Z25D.MASTER) reside on this volume. This volume is required to IPL. D5SYS1 contains the usercatalog USERCAT.Z25D.PRODS containing entries for z/OS related products
D5CFG1	The 'USER.**' data sets reside on this volume, they are cataloged in the master catalog. The RACF database resides on this volume. **D5CFG1 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.
D5USS1, D5USS2, D5USS3	D5USS1 Contains the Fonts, Version and zOSMF zFS files for UNIX System Services of z/OS. D5USS2 Contains root zFS and zFS files for UNIX System Services of z/OS D5USS3 Contains root zFS and zFS files for UNIX System Services of z/OS **D5USS1, D5USS2, and D5USS3 are all required for IPL** D5USS1 is a 3390-9 volume that contains the Fonts, Version and zOSMF zFS files for UNIX System Services of z/OS.

	D5USS2 is a 3390-9 volume that contains root zFS and zFS files for UNIX System Services of z/OS. D5USS3 is a 3390-9 volume that contains root zFS and zFS files for UNIX System Services of z/OS.
D5PAGA, D5PAGB, D5PAGC	D5PAGA contains the PLPA, common and one Local page data set. D5PAGB contains Contain one Local page data set D5PAGC contains Contain one Local page data set **D5PAGA is required for IPL** **Note: The loadparms that bring up DB2 or WAS have been defined to use D5PAGA D5PAGB and D5PAGC , 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 D5PAGA 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.
D5PRD1, D5PRD2, D5PRD3, D5PRD4, D5PRD5	D5PRD1 to D5PRD5 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_Product: - Volumes for Supported Products

Volume	Description
D5DIS1, D5DIS2, D5DIS3	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.
D5DBD1, D5DBD2,	D5DBD1 is a 3390-9 volume that contains DB2 V13 related data. D5DBD1 contains the DB2 usercatlog, USERCAT.Z25D.DB2VD. Also D5DBD1 contains all DB2 target libraries and DB2 V13

D5DBAR	utility products. This volume is not required to IPL but is required to bring up DB2 V13. D5DBD2 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. D5DBAR 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.
D5DBC1, D5DBC2, D5DBAR	D5DBC1 is a 3390-9 volume that contains DB2 V12 related data. D5DBC1 contains the DB2 usercatlog, USERCAT.Z25D.DB2VC. Also D5DBC1 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. D5DBC2 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. D5DBAR 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.
D5C560	D5C560 is a 3390-9 volume that contains CICS 5.6 Target, DLIB and all related data for CICS 5.6. USERCAT.Z25D.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.
D5C610	D5C610 is a 3390-9 volume that contains CICS 6.1 Target, DLIB and all related data for CICS 6.1. USERCAT.Z25D.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.
D5W901, D5W902	D5W901 and D5W902 are 3390-9 volumes that contains target and distribution libraries for Websphere Application Services.
D5IMF1	D5IMF1 is a 3390-9 volume that contains IMS Target, DLIB and all related data for IMS 15. USERCAT.Z25D.IMS15 resides on this volume. This volume is not required to IPL but is required to bring up IMS and perform maintenance on IMS.
D5KAN1	D5KAN1 is a 3390-9 volume that contains Target and DLIB libraries for IBM Z Monitoring Suite 1.3.0.
D5BLZ1	D5BLZ1 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.
D5INM1	D5INM1 is a 3390-9 volume that contains Target and DLIB libraries for Installation Manager 1.8.8.

Table_SMS: - Volumes for SMS Support

Volume	Description
D5USR1	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 D5USR1. 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_SMSI: - SMS Information

Item	Description
DCEXTEAV	Data Class
SGEXTEAV	Storage Group
SCEXTEAV	Storage class
D5USR1	Volume defined to SGEXTEAV (Storage Group)
SYS1.SMS.CNTL	Location of the SMS routines

D5ZCX1	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 D5ZCX1. 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_SMSI: - SMS Information

Item	Description
CXDC	Data Class
CXROOTSG	Storage Group
CXROOTSC	Storage class
D5ZCX1	Volume defined to CXROOTSG (Storage Group)
SYS1.SMS.CNTL	Location of the SMS routines

Table_zOWE: - Volume for zOWE Support

Volume	Description
D5ZWE1	D5ZWE1 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 .

Back to Table of Contents

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 D5xxxx volumes except for D5SYS1. D5CFG1 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.Z25D.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.Z25D.xxxxxx libraries be backed up regularly. When a new ADCD release is installed the USER.Z25D.xxxxxx libraries can be copied from backups to the new USER.Z25D.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.Z25D.LINKLIB
- ADCD.Z25D.LINKLIB
- SYS1.LINKLIB

USER.Z25D.xxxxxx libraries have all been built on D5CFG1 which is the only volume that should contain user data.

USER.Z25D.xxxxxx will not be changed by ADCD or System processes; thus, user updates in USER.Z25D.xxxxxx will be retained between release levels. ADCD.Z25D.xxxxxx libraries are allocated on D5SYS1 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.Z25D.xxxxxx
- Edit, compile, or run user program against the USER.Z25D.xxxxxx item
- Re-ipl, recycle system service, or re-logon to TSO

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

Back to Table of Contents

LOADPARMS Options

NOTE: JES2 should be COLD started the FIRST time you bring up the system.

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 5.5 libraries. up Starts up CICS 5.6 and RDz.
DB	CLPA and Warm start of JES2. Loads DB2 V12 libraries. Starts up DB2 V12 and RDz.
IM	CLPA and Warm start of JES2. Loads IMS 15 libraries. up Starts up IMS 15 and RDz.
IZ	CLPA and Warm start of JES2. Starts up z/OSMF and RDz.
WA	CLPA and Warm start of JES2. Loads WAS 9.0 libraries.Starts up RDz. WAS needs to be manually started.
AL	CLPA and Warm start of JES2. Loads all middleware libraries. Starts up CICS 5.5, DB2 V12, IMS 15, z/OSMF and RDz.
DC	CLPA and Cold start of JES2. Loads CICS, DB2 libraries. Starts up CICS 5.6, DB2 V12, UCD 7.2.0, RTC 7.0.2 and RDz.
DW	CLPA and Warm start of JES2. Loads CICS, DB2 libraries. Starts up CICS 5.6, DB2 V12, UCD 7.2.0, RTC 7.0.2 and RDz.

Back to Table of Contents

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

Action/Command	PFKey
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

Back to Table of Contents

USERIDS

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
ADCDA - ADCDZ	TEST
WEBADM	WEBADM
OPEN1 thru OPEN3	SYS1

Back to Table of Contents

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 V2R5 May 2023 Edition are at PUT2303 / RSU2303
- All functions of the CICS TS 5.6 are at PUT2303 / RSU2303.
- All functions of the CICS TS 6.1 are at PUT2303 / RSU2303.
- The functions of the DB2 12 base are at PUT2303 / RSU2303.
- The functions of the DB2 13 base are at PUT2303 / RSU2303.
- All functions of the IMS 15.1.0 are at PUT2303 / RSU2303.
- IBM Z Monitoring Suite V1.3.0 is at PUT2303 / RSU2303.

Back to Table of Contents

Extended ADCD for ZVDT

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 16.0.2 (HLQ - ISM332)
- z/OS Explorer API Version 1.1 (HLQ - ISM332)
- IBM Z® Open Automation Utilities Version 1.2.4 (HLQ - ISM332)
- Rocket Git 2.26.2-87 (HLQ - ISM332)
- Rocket Bash 5.1.0 (HLQ - ISM332)
- Rocket Perl 5.32.1 (HLQ - ISM332)
- IBM® Python SDK 3.9.11 (HLQ - CYP390)

Software packages that are updated:

- IBM Dependency Based Build is updated to 2.0 (HLQ - ISM333)
- IBM Debugger is updated to 16.0.2 (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 ZVDT.
- **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 ZVDT 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 ZVDT.

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 ZVDT, the member IEASYMAU 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 ZVDT 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 ZVDT 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 ZVDT 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 ZVDT 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 ZVDT.

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/zvdt/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 ZVDT system with Extended ADCD and decide to port a specific application to the ZVDT 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 ZVDT. 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 ZVDT z/OS image. The system will attempt to mount all ADCD ZFS files regardless of what is contained in the ZVDT 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 ZVDT 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 ZVDT 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 ZVDT. However, when the manual IPL process of z/OS is done, ZVDT 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.

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 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 ZVDT-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.

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 zvdt-install-x86.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 ZVDT 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 ZVDT web server UI to connect to the storage server.
- **Adding a software-based license server**
  The license servers that are used by each target ZVDT 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 ZVDT.
- **Optional: Configuring IBM AD integrations**
  To make ZVDT 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 Linux on IBM Z target environments**
  Learn how to add, configure, or delete a Linux on IBM Z target environment on the web server.

Connecting to your storage server

After you enabled an SFTP protocol, you can go to the ZVDT 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 ZVDT 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 ZVDT.

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 ZVDT 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 ZVDT 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 ZVDT
: Supports faster volume component creation by compressing and transferring an existing z/OS instance that ZVDT 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 ZVDT**
  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 ZVDT

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 or ZVDT instance 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 or the ZVDT hypervisor 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/zvdt.
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.

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 Optional: Configuring Extended ADCD.

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:

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 Optional: 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 Linux on IBM Z target environments

Learn how to add, configure, or delete a Linux on IBM Z target environment on the web server.

To add a Linux on IBM Z 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. Select ZVDT as the target environment type.
4. Enter a unique name that can be used to identify the Linux on IBM Z target environment.
5. Select the system configuration method. In this version, manual system configurations are required, and the Manual Configuration option is selected by default. You need to manually install the ZVDT hypervisor and configure how each instance handles the networking on the target environment.
6. Optional: Enter notes that can help you identify the target environment.
7. Enter the qualified hostname and the Secure Socket Shell (SSH) port number of the target environment.
8. Select the authentication type, Credential or Certificate, enter your authentication information, and then click Establish connection.
9. Select the networking topology. You can use either MacVTap interfaces or iptable rules on existing Linux network interfaces.
   If you select Use MacVTap adapters, complete the following steps:
   a. Specify an IP address range for instances to use. Each instance acquires one IP address from the specified range. The last octet of the IP address must not exceed 254.
   b. Select virtual tap adapters for instances to use. Each instance requires one virtual tap adapter, so the number of the selected adapters must equal the maximum number of instances that is specified in the previous step.
   If you select Use iptable rules on existing Linux network interfaces, enter the z/OS IP address and the Linux port that routes to the z/OS SSH.
10. Click Add a target environment.

Enabling z/OS software to run on Linux on IBM Z

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.

- **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 to Linux on IBM Z 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 hypervisor.

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.

ZVDT 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 or ZVDT 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 or ZVDT runs on Linux**
  After you configure an existing z/OS instance that ZD&T or ZVDT 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 ZVDT 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 ZVDT.

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 ZVDT, or grant read access to the data sets to be extracted to the user ID that runs ZVDT.

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>*/zvdt-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 ZVDT. 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 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 ZVDT 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 ZVDT machine. Alternatively, you can search the terms that you want.
   Note: Currently, ZVDT 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 or ZVDT runs on Linux

After you configure an existing z/OS instance that ZD&T or ZVDT 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 z/OS Extraction Utilities.

To create a volume component, complete the following steps.

1. Select a Linux running a ZD&T or ZVDT 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 or the ZVDT hypervisor.

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 to Linux on IBM Z 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.

- **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.

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 hypervisor 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.
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.

ZVDT 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 a hypervisor 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.
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.

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

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 ZVDT hypervisor 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 ZVDT 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.

2. 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.

3. Hypervisor status
   The hypervisor cannot be running.

4. 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.
5. Storage server connectivity
   - If you use SFTP as the storage server, the port number must be 22.
6. User and group configurations
   A non-root user ID, for example, ibmsys1, must be created and assigned to a KVM group. For instructions, see Installing and configuring the hypervisor.

7. Dependent libraries
   All dependencies must be installed on the target environment. For more information about the dependencies and command examples, see Linux on IBM Z requirements.

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

9. 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.

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/zvdt/volumes.

   a. **`awsstop`**

      Wait for few minutes for a running ZVDT to stop.

   b. **`awsstart aprof1`**

      Wait for few minutes for ZVDT 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 Linux on IBM Z environment**

Provisioning instances to a Linux on IBM Z environment

After images are created, you can provision z/OS instances from the images to an Linux on IBM Z target environment on the Images page.

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

1. From the drop-down menu in the TARGET ENVIRONMENT pane, select an Linux on IBM Z environment.
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. 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.
   - 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 the directory where ZVDT stores installation and volume files during the provisioning process. The user must have read and write permissions to perform the provision action, and the directory must contain adequate space to hold all of the image files.
   Note: ZVDT will create a subdirectory that is named 'zvdt' inside the deployment directory, and a couple of subdirectories inside 'zvdt'. 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. ZVDT 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 provisioned z/OS instance.

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 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. Then, click Provision instance.

    After you click Provision instance, the pre-provisioning validation will automatically check permission and the capability of the deployment directory. For more information, see items 9 - 11 in Pre-provisioning validation.

You can also monitor and manage the provisioning or provisioned instances on the Instances page. For more information, see Monitoring and managing the provisioned instances.

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 hypervisor.

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 Linux on IBM Z environments

To remove provisioned instances from Linux® on IBM Z 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 and revert network configurations.
   - Remove the image.

   The options are selected by default. You can select to remove ZVDT instances and revert network configuration, but keep the images in the deployment directory. Or, you can clear the option to remove ZVDT instances and revert network configuration, and the images also cannot be removed.
3. Enter the required credentials, and click Submit.

Note: An instance that is being provisioning cannot be removed or resumed. Also, you cannot start or stop the z/OS hypervisor currently.

Using REST APIs

ZVDT supports REST APIs usage. You can access all functions of ZVDT 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 ZVDT, and know the restrictions of REST APIs usage.

To find the complete specification of all the available REST APIs provided by ZVDT 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 'zvdtadmin' 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/zvdt/Liberty/usr/servers/zvdt-Server/bootstrap.properties.
2. Run the following command where you install the web server.

   ```
   /opt/ibm/zvdt/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/zvdt/Liberty/usr/servers/zvdt-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_SSTAXM7_1.4.0_com.ibm.zsy
   s.rdt.tools.user.guide.doc_topics_api_usage_zvdt" realm="zvdt">
   <user name="zvdtadmin"
   password="{aes}AM1LZsnwLRNsVtYAiwqhVDO9/RL+NgYthDZXZhQgARtB"/>
   </basicRegistry>
   ```

- **Adding REST APIs through Zowe API Gateway**
  The ZVDT installer contains a YAML file that can be used to add ZVDT 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 Linux on IBM Z target environments with REST APIs**
  You can call REST APIs to create and manage Linux on IBM Z 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 a Linux on IBM Z environment with REST APIs**
  You can provision a z/OS instance to a specific Linux on IBM Z target environment by using the UUID.
- **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 ZVDT installer contains a YAML file that can be used to add ZVDT REST APIs through Zowe™ API Gateway.

To add ZVDT REST APIs through Zowe API Gateway, complete the following steps.

1. Run installer zvdt-install-x86.tgz to install ZVDT, and download the YAML file that is located at the directory /opt/ibm/zvdt/zowe.
2. Obtain the IP and port of your ZVDT web server.
3. Update the YAML file to replace the pattern *{zvdt tools machine IP}:{zvdt 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.

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/

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

```
curl -k -u zvdtadmin:password -X POST --header 'Content-Type: application/json' --
header 'Accept: application/json' -d '{ "hostname": "your_target_system_host_name",
"sshPort": 22,  "label":"zvdtForTesting", "installOSPackages": true,
"concurrentVolumeTransferNumber": 3, "configNetwork": true,  "configUser": true,
"syntax": "sudo ${command}",  "syntaxForValidation": "sudo -n -l",
"networkInterface": "eth1", "tcpPortRules": "0:21 \n 23:2021 \n 2022>22 \n 2023:3269
\n 3271:9449 \n 9452:65535", "udpPortRules": "111 \n 514 \n 1023 \n 1044:1049 \n
2049", "zvdtManaged": "true", "hasRootAccess": "true"}'
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/targetEnvServices/targetSys
tem'
```

Creating and managing Linux on IBM Z target environments with REST APIs

You can call REST APIs to create and manage Linux on IBM Z target environments.

Make sure that your Linux on IBM Z target environment meets all the prerequisites and is configured as instructed in topic Installing and configuring the hypervisor.
You can create a Linux on IBM Z target environment with a UUID contained in the response by using the REST API. This UUID represents the Linux on IBM Z target environment that you create. You can manage the Linux on IBM Z target environment by using the UUID.
To create or manage Linux on IBM Z 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/

1. Use the following REST API to create a Linux on IBM Z target environment. After a Linux on IBM Z target environment is created, the response contains a UUID that represents the Linux on IBM Z target environment.

   ```
   curl -X 'POST' -k -u zvdtadmin:password \
     'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/target-environment-
   services/v1/linux-s390x' \
     -H 'accept: application/json;charset=utf-8' \
     -H 'Content-Type: multipart/form-data' \
   ```

```
  -F 'linux-target-request={
  "resume": false,
  "label": "string",
  "description": "string",
  "hostname": "mytarget.domain.com",
  "ssh-port": 22,
  "zvdt-managed": false,
  "ip-assignment-type": "CLIENT_MANAGED",
  "ip-start": "172.10.9.1",
  "ip-end": "172.10.9.10",
  "dhcp-port": 68,
  "macvtaps": [
    "my_macvtap"
  ],
  "max-instances": 1,
  "default-route": "172.10.9.1",
  "linux-creds": {
    "username": "myuser",
    "password": "mypassword"
  }
}' \
  -F 'private-key='
```

2. After a Linux on IBM Z target environment is created, you can use the following REST APIs to get, update, or delete the target environment.

   Getting a Linux on IBM Z target environment

   UUID is required to get a Linux on IBM Z target environment. To get a Linux on IBM Z target environment, use the following REST API.

   ```
   curl -X 'GET' -k -u zvdtadmin:password \
     'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/target-
   environment-services/v1/linux-s390x/your-linux-s390x-target-system-uuid ' \
     -H 'accept: application/json;charset=utf-8'
   ```

   Updating a Linux on IBM Z target environment

   UUID is required to update a Linux on IBM Z target environment. To update a Linux on IBM Z target environment, use the following REST API.

   ```
   curl -X 'PUT' -k -u zvdtadmin:password \
     'https://your_tools_server_host_name/ZDTREST/zdtrs/target-environment-
   services/v1/linux-s390x' \
     -H 'accept: application/json;charset=utf-8' \
     -H 'Content-Type: multipart/form-data' \
     -F 'linux-target-request={
     "uuid": "your-linux-s390x-target-system-uuid",
     "label": "new label",
     "description": "new description",
   }' \
     -F 'private-key='
   ```

   Deleting a Linux on IBM Z target environment

   UUID is required to delete a Linux on IBM Z target environment. To delete a Linux on IBM Z target environment, use the following REST API. After a Linux on IBM Z target environment is deleted, the response displays 200 `(OK)` status code.

   ```
   curl -X 'DELETE' -k -u zvdtadmin:password \
     'https://localhost:9443/ZDTREST/zdtrs/target-environment-
   services/v1/linux-s390x/<UUID>?force=false&resume=false' \
     -H 'accept: application/json;charset=utf-8'
   ```

   The following two query parameters are optional:

   force

Specifies whether the deletion is forced to complete. This does not guarantee that the artifacts on the target environments are properly removed. The default value is false.

resume
Specifies whether you are resuming a failed deletion. The default value is false.

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 ZVDT instances**
  Learn about the steps to create volume components from existing ZD&T or ZVDT 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.

1. Call the REST API to authenticate the SSH connection to the z/OS® system.

   ```
   curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin: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/ZVDTREST/zvdtrs/componentServices/v2/z
osVolumeComponent'
```

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.

1. Call the REST API to authenticate the SSH connection to the z/OS® system.

   ```
   curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/auth-
   services/v3/authenticate-ssh'
   ```

   The following response is displayed.

   ```
   {"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVN
   FUjM1NTE4ODczMjY3NjU=","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 zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin:password -X POST --header 'Content-Type: application/json'
   --header 'Accept: application/json' --header 'SSHAuthorization:
   {"token":"dXN1cjM1NTg5OTg4OTgwNA==","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/ZVDTREST/zvdtrs/componentServices/zosDa
tasetComponent
```

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.

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 zvdtadmin: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/ZVDTREST/zvdtrs/auth-
   services/v3/authenticate-ssh'
   ```

   The following response is displayed.

   ```
   {"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVN
   FUjM1NTE4ODczMjY3NjU=","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 zvdtadmin:password -X POST --header 'Content-Type: application/json'
   --header 'Accept: application/json' --header 'SSHAuthorization:
   {"token":"dXN1cjM1NTg5OTg4OTgwNA==","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/ZVDTREST/zvdtrs/componentServices/cics
   Component
   ```

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.

1. Call the REST API to authenticate the SSH connection to the z/OS system.

   ```
   curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin:password -X POST --header 'Content-Type: application/json'
--header 'Accept: application/json'
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/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 zvdtadmin: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/ZVDTREST/zvdtrs/db2/v1/schemas?
hostName=your_zsystem_host_name&sshPort=22&subsystemName=DC1A
```

After the request is complete, the list of schema names is displayed.

```
["ADB","CLARKG","DSN8110","DSN81110","DSN811SA","DSN8BQRY","DSNRGCOL","IBMUSER",
"Q","RUNSTATS","SYSIBM","SYSIBMTS","SYSTOOLS"]
```

4. Obtain the list of artifacts by passing the schema name with the hostname, port, and subsystem ID of the z/OS system, and providing the authentication to the z/OS system.

```
curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/db2/v1/schemas/your_sc
```

```
hema_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,"uns
upportReason":null,"tableSpaceName":"PLANRTAB","creationTime":null,"subSystem":n
ull,"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 zvdtadmin: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","ftpSyste
m":
{"hostname":"your_ftp_system_host_name"},"concurrentExtractionNumber":"1","subsy
stemName":"your_subsystem_id","selectedTables":
[{"name":"PLAN_TABLE","tableType":"TABLE","schemaName":"your_db2_schema_name","d
atabaseName":"your_db2_database_name"}],"integrity":true,"db2DDL":""}'
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/componentServices/v1/d
b2'
```

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.

1. Call the REST API to authenticate the SSH connection to the z/OS® system.

```
curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin:password -X GET --header 'Accept: application/json'
https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/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 zvdtadmin:password -X GET --header 'Accept: application/json'
https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/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 zvdtadmin:password -X GET --header 'Accept: application/json'
https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/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 zvdtadmin: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/ZVDTREST/zvdtrs/componentServices/zosDa
tasetComponent
```

Creating volume components from existing ZD&T or ZVDT instances

Learn about the steps to create volume components from existing ZD&T or ZVDT instances.

1. Call the REST API to authenticate the SSH connection to the z/OS® system.

```
 curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin: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/ZVDTREST/zvdtrs/zvdt-
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 ZVDT instance.

```
curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/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 zvdtadmin: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_ZVDT_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/ZVDTREST/zvdtrs/componentServices/zvdt
-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.

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

```
curl -k -u zvdtadmin:password -X GET --header 'Accept: application/json'
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/componentServices/component
s/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.

1. Call the REST API to authenticate the ssh connection to the z/OS system.

```
curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/auth-
services/v3/authenticate-ssh'
```

The following response is displayed.

```
{"expiration":1516202935577,"username":"ZOSUSER","password":"ZOSPW","token":"VVN
FUjM1NTE4ODczMjY3NjU=","port":22}
```

2. Run the following command to submit the request.

```
curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/componentServices/comp
onent'
```

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.

1. Create the images by including some of the components that you have created earlier.

```
curl -k -u zvdtadmin: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/ZVDTREST/zvdtrs/appServices/v2/zosAppl
icationImage'
```

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 zvdtadmin: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/ZVDTREST/zvdtrs/appServices/zosApplica
tionImage'
```

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.

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 zvdtadmin:password -X GET
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/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,
      "zvdtVolumeSms": 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 zvdtadmin:password -X GET
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/componentServices/comp
onents?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 zvdtadmin: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/ZVDTREST/zvdtrs/appServices/ADCDA
     pplicationImage'
     ```

     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 zvdtadmin: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/ZVDTREST/zvdtrs/appServices/ADCDA
     pplicationImage'
     ```

   - **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 zvdtadmin: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/ZVDTREST/zvdtrs/appServices/ADCDA
pplicationImage'
```

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.

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

```
curl -k -u zvdtadmin:password -X GET --header 'Accept: application/json'
'https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/appServices/application/YOU
R_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.

1. To edit an image that is not provisioned, run the following command.

```
curl -k -u zvdtadmin:password -X PUT --header 'Content-Type: application/json' -
-header 'Accept: application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zvdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":
{"domain":0,"ccp":0},"newInfo":{"name":"zvdt-
dev1.wholeimage.without.LOAD$1"},"addItems":[{"name":"zvdt-
dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":
[{"name":"zvdt-dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":false,"new
Version":false,"sysResComponent":{"name":"zvdt-
dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}'
https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/appServices/application
```

2. To create a new version of an image, run the following command.

```
curl -k -u zvdtadmin:password -X PUT --header 'Content-Type: application/json' -
-header 'Accept: application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zvdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":
{"domain":0,"ccp":0},"newInfo":{},"addItems":[{"name":"zvdt-
dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":
[{"name":"zvdt-dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":false,"new
```

```
Version":true,"sysResComponent":{"name":"zvdt-
dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}'
https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/appServices/application
```

3. To duplicate an image to a new image, run the following command.

```
curl -k -u zvdtadmin:password -X PUT --header 'Content-Type: application/json' -
-header 'Accept: application/json' -d '{"version":3,"zSystem":
{"hostname":"myzos.domain.com","port":1022},"name":"zvdt-
dev1.wholeimage.with.LOAD$1","zsystemInformationUpdateRequest":
{"domain":0,"ccp":0},"newInfo":{"name":"zvdt-
dev1.wholeimage.with.LOAD$1.duplicated"},"addItems":[{"name":"zvdt-
dev1.1vol.s25","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"removeItems":
[{"name":"zvdt-dev1.1vol.LOAD$1","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":1}],"duplicate":true,"newV
ersion":false,"sysResComponent":{"name":"zvdt-
dev1.wholeimage.with.volinit","zSystem":
{"hostname":"myzos.domain.com","port":1022},"version":3}}'
https://your_tools_server_host_name:9443/ZVDTREST/zvdtrs/appServices/application
```

Provisioning instances to a Linux on IBM Z environment with REST APIs

You can provision a z/OS® instance to a specific Linux on IBM Z target environment by using the UUID.

After a Linux on IBM Z target environment is created by using REST APIs, the response contains a UUID that represents the Linux on IBM Z target environment. You can use this UUID to provision instances to the target environment. For more information, see Creating and managing Linux on IBM Z target environments with REST APIs.
To provision instances to a Linux on IBM Z target environment by using the UUID, follow the steps:

1. Create a provisioning to a Linux on IBM Z 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 zvdtadmin:password \
  'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-
services/v1/linux-s390x' \
  -H 'accept: application/json;charset=utf-8' \
  -H 'Content-Type: multipart/form-data' \
  -F 'linux-provision-request={
  "emulator": {
    "cp": 3,
    "ram": "3221225472",
    "ziip": 0
  },
  "ipl": {
    "device-address": "a80",
    "iodf-address": "0a82",
    "load-suffix": "au"
  },
  "general": {
    "label": "label",
    "description": "description",
    "target-uuid": "your-linux-s390x-target-system-uuid",
    "resume": false,
    "update": false,
    "application-uuid": "your-image-uuid"
```

```
  },
  "zos-creds": {
    "username": "zos_user",
    "password": "zos_pwd"
  },
  "deployment-directory": "/deployment/",
  "linux-creds": {
    "username": "user",
    "password": "pwd"
  },
  "validate-linux": true
}' \
  -F 'private-key='
```

2. After you create a provisioning, you can use the following REST API to monitor the provisioning progress.

```
curl -X 'GET' -k -u zvdtadmin:password \
  'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/monitor-
services/v1/provision-status/<UUID>' \
  -H 'accept: text/event-stream'
```

If the response is `"status": "SUCCEEDED"`, the provisioning is completed. To resume a failed provisioning, use the following REST API:

```
curl -X 'POST' -k -u zvdtadmin:password \
  'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-
services/v1/linux-s390x' \
  -H 'accept: application/json;charset=utf-8' \
  -H 'Content-Type: multipart/form-data' \
  -F 'linux-provision-request={
  "emulator": {
    "cp": 3,
    "ram": "3221225472",
    "ziip": 0
  },
  "ipl": {
    "device-address": "a80",
    "iodf-address": "0a82",
    "load-suffix": "au"
  },
  "general": {
    "provision-uuid": "your-provision-uuid",
    "resume": true,
    "update": false
  },
  "zos-creds": {
    "username": "zos_user",
    "password": "zos_pwd"
  }
}' \
```

3. If you need to deprovision from a Linux on IBM Z target environment, use the following REST API. After the deprovisioning is completed, the response displays 200 `(OK)` status code.

```
curl -X 'DELETE' -k -u zvdtadmin:password \
  'https://your_tools_server_host_name:9443/ZDTREST/zdtrs/provision-
services/v1/linux-s390x/<UUID>?force=false&resume=false&preserve=false' \
  -H 'accept: application/json;charset=utf-8'
```

The following three query parameters are optional:

force
: Specifies whether the deprovisioning is forced to complete. This does not guarantee that the artifacts on the target environments are properly removed. The default value is false.

preserve

Specifies whether the volume files are preserved. The default value is false.

resume

Specifies whether you are resuming a failed deprovisioning. The default value is false.

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.

Error ID	Description
10206	Failed to set the file transfer type.
10207	A failure occurred when storing content on the storage server.
10208	A failure occurred when retrieving content from the storage server.
10209	Failed to create directory *{0}* on the storage server.
10210	Unable to access the directory *{0}* on the storage server.
10211	Failed to delete the file *{0}* from the image storage server.
10213	Internal Error.
10214	The old base directory is needed.
10215	This feature is supported only for SFTP storage system.
10216	This source system cannot be deleted because there exists components or applications related to it.
10300	Failed to connect to the IBM Application Discovery server *{0}* on port *{1}*.
11001	Installation files download failed.
11003	Deploy request parameter not passed.
11007	Invalid Linux® platform of target system.
11011	zPDT® installation script error: Parameters invalid.
11012	zPDT installation script error: Primary licensing server is required.
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 ZVDT 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 ZVDT with missing path.
11036	Patch installation failed when installing ZVDT.
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.

Error ID	Description
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 ZVDT image was not provided.
12008	The compression directory does not exist in the specified location: *{0}*.
12009	The user *{0}* does not have write access to the compression directory *{1}*.
12010	Should not specify the compression directory *{0}* because the compression option is off.
20001	Authorization token has expired.
20002	Authorization token was not supplied.
20003	User already exists.
20004	User does not exist.
21001	The machine running the ZVDT 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.

Error ID	Description
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.

Error ID	Description
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.
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 ZVDT image failed due to missing required input.
30061	The request to validation an existing ZVDT 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.

Error ID	Description
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.
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.

Error ID	Description
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.

Error ID	Description
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.
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 ZVDT hypervisor 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 ZVDT hypervisor 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}*.

Error ID	Description
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.
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}*.

Error ID	Description
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 ZVDT 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.

Error ID	Description
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.
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.

Error ID	Description
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 ZVDT instance controller service listening on port *{1}* of host *{0}*.
50941	A failure occurred while executing an Instance Controller API.
50942	An unsupported Instance Controller HTTP method was called.
50943	An attempt was made to call an Instance Controller API with invalid credentials (HTTP status code 401).
50944	An attempt was made to call an Instance Controller API for which the user is not authorized to access (HTTP status code 403).

Error ID	Description
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 ZVDT extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS returned an unexpected result.
70991	Checking *{0}* version for the ZVDT 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 ZVDT extraction utility SFTP module (/usr/lpp/IBM/zdt/zDTMainframeSFTP.jar) on this z/OS is either not installed or corrupted.
70994	The ZVDT 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 ZVDT extraction utility (/usr/lpp/IBM/zdt/feuc) has not been installed.
70996	The ZVDT extraction utility version file could not be found.
70997	The ZVDT 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 ZVDT extraction utility (/usr/lpp/IBM/zdt/feuc) on this z/OS is older than the current version.
70999	Internal error acquiring information from z/OS.
79999	An error occurred when connecting to z/OS: *{0}*.
99998	Unexpected exception.
99999	Unknown error.

Troubleshooting and workarounds

Learn how to troubleshoot ZVDT 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 ZVDT.
- **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 ZVDT.

On the Linux system where the web server is installed

/var/log/zvdt*.log
: Records events and activities of the web server installer.

<installation_path>/zvdt/zvdt-server/logging.xml
: Specifies the logging level configurations of the web server.

<installation_path>/zvdt/zvdt-server/logs/*.log
: Records events and activities within the web server.

<installation_path>/zvdt/zvdt-server/postgresql/logfile
: Records database-related issues that might occur when attempting to start the web server.

<installation_path>/zvdt/zvdt-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>/zvdt/log/
: Records provisioning events and activities.

/home/ibmsys1/z1090/logs/
: For provisioning by the root user ID. Records events and activities of the hypervisor, 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 hypervisor, with the target z/OS console logs stored in files that are named log_console*.txt.

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

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 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.

Symptom	Problem determination
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 ZVDT 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.

Troubleshooting for installation

Table 1. Troubleshooting the installation of the ZVDT

Symptom	Problem determination
When you are attempting to run launchpad.sh or launchpad-console.sh,	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 get the following error: `./InstallerImage_linux.gtk.x86_64/ tools/imcl: Permission denied`	• 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.

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 defining the ZVDT machine characteristics (DEVMAP)

Table 1. Defining the ZVDT machine characteristics (DEVMAP). Troubleshooting defining the ZVDT 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 ZVDT. This updates the path to the needed files.

Troubleshooting for the starting and stopping of ZVDT

Table 1. Starting and stopping ZVDT. Troubleshooting the starting and stopping of ZVDT

Symptom	Problem Determination
`AWSSTA140E – Initialization failure`	Ensure that you are starting ZVDT from the home directory of the user who is created for running ZVDT.

`after license obtained`	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`**.

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/OutOfMemory Error: 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 ZVDT 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 ZVDT 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.
	  ○ 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

Symptom	Cannot SFTP or Telnet to a z/OS defined IP address on the ZVDT instance
Problem Determination	• Verify that you can ping to the address from the remote location. If you cannot, follow the steps defined here. • If you can ping, check your firewall configuration with the command `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 ZVDT 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:

	# 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)** You might need to experiment with these commands. These commands must be entered after each Linux boot. If a script is used to start ZVDT, such as the **runzpdt** script shipped with the product, that script can be modified to enter one or more of these commands each time ZVDT 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**
Symptom	ZVDT performs slowly. CPUs reach 100% utilization.
Problem Determination	Verify that you are meeting the minimum requirements for both the hardware and the software. See Hardware and software requirements.

Troubleshooting tips

Table 1. Troubleshooting tips

Symptom	Problem Determination
For any problem after installation of ZVDT (creating a devmap, installing z/OS®, starting or stopping ZVDT, 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 hypervisor is terminating abnormally.
AWScccnnns message that is issued from the Linux® terminal that is running ZVDT.	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.

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.

Uninstalling

Learn how to uninstall ZVDT.

- **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. Open the directory where the installer is stored.

   ```
   cd <directory>
   ```

   License server installer for Linux on x86
   zdt-license-server
   License server installer for Linux on IBM Z (Ubuntu only)
   zdt-license-server-s390x

3. 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
   ```

4. 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
```

Ubuntu Linux on IBM Z

```
dpkg -l | grep zdtlicense
```

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.

To uninstall the web server, complete the following steps:

1. Open the directory where the installation package is stored.

   ```
   cd <directory>
   ```

   Package for Linux on x86
   zvdt-install-x86.tgz
   Package for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x.tgz

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 for Linux on x86
   zvdt-install
   Web server installer for Linux on IBM Z (Ubuntu only)
   zvdt-install-s390x

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 --zvdt
   ```

Terminology

Find terms that are used in this documentation.

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.

Extended Application Developers Controlled Distribution (ADCD)
: ZVDT 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.

hypervisor
: The ZVDT hypervisor provides the ability to run unaltered z/OS software on Linux® on IBM Z®. Manual system configurations are required on the target environment. You need to install the hypervisor and configure how each instance will handle the networking. For installation instructions, see Installing and configuring the hypervisor.

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.
: In the IBM® License Key Center, a quantity of product entitlement parts purchased.

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 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 ZVDT, a network-accessible service that manages and dispenses ZVDT 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” ZVDT installation internally accesses the license server through internal TCP/IP. Remote license servers are accessed through network TCP/IP.

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.

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 ZVDT to provide the z Systems serial number.

software-based license server
: The remote ZVDT 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 ZVDT 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
: ZVDT 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 Linux on IBM Z environment that is capable of running one or more z/OS instances.

virtual tap adapter
: If you choose to use MacVTap interfaces as the networking topology, each z/OS instance of ZVDT requires a MacVTap adapter, which allows the instance to have a dedicated IP address and the full set of ports for use.
  For more information about the MacVTap adapter, see Using the MacVTap driver.

web server
: A server that enables ZVDT provisioning tools. It is a component of ZVDT.
  The web server can be installed on the same system as the software-based license server.

ZD&T z/OS Extraction Utilities
: A component of ZVDT to be installed on one or more IBM Z platforms to allow ZVDT 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.