Installing IBM Unified Management Server for z/OS for IBM IMS Administration Foundation for z/OS

Document Version 2

Contents

Overview

This information is intended for those who plan to install and configure only IBM® IMS Administration Foundation for z/OS® (IMS Administration Foundation) of IBM® IMS Tools Base for z/OS® (IMS Tools Base) 1.7 on IBM® Unified Management Server for z/OS® (UMS).

- If you plan to install and configure IMS Administration Foundation of IMS Tools Base 1.8, refer to the IMS Administration Foundation Installation and Configuration Guide.
- If you also plan to install other UMS extension products, such as IBM Db2® Administration Foundation for z/OS or IBM Db2 DevOps Experience for z/OS, refer to the IBM Unified Management Server for z/OS 1.2 User Guide (UMS User Guide) instead of this information.

This information covers the prerequisites, pre-installation tasks, installation, and configuration instructions for UMS and IMS Administration Foundation to help you enable IMS Administration Foundation features on UMS.

This information also explains the UMS installation script, which generates required JCL members for each step of the installation process. Each JCL member contains detailed instructions to help you execute the script.

What you need to know before using this information

This information is designed to help you set up UMS and IMS Administration Foundation. It includes references to topics in the *UMS User Guide*, which provides installation and configuration instructions not only for IMS Administration Foundation but for all UMS extension products. Therefore, some topics in the *UMS User Guide* may not be relevant if you are installing and configuring only IMS Administration Foundation with UMS. If you are installing and configuring only IMS Administration Foundation on UMS, follow the instructions in this guide and refer to the *UMS User Guide* as directed.

The security configuration instructions in this information are specific to RACF. If you are using an external security manager (ESM) other than RACF, you might need to adjust these instructions to suit your specific ESM environment.

Installation and configuration flow

Installation and configuration of UMS and IMS Administration Foundation consist of the following steps.

Table 1: Installation and configuration flow

Step	Major role for the task
“Installation planning” on page 4	Infrastructure architect or engineer
“Installing UMS and its prerequisite software” on page 6	z/OS system programmer
“Preparing data sets and USS directories for UMS” on page 8	z/OS system programmer
“Configuring security for UMS and IMS Administration Foundation” on page 9	z/OS security administrator
“Configuring UMS and IMS Administration Foundation” on page 18	z/OS system programmer, UMS server administrator

Installation planning

Before you start the installation, make sure that you understand the UMS architecture and resource dependencies. Then, confirm that all prerequisites for installing UMS and IMS Administration Foundation are met.

Use the following information to plan the installation:

UMS architecture

UMS is installed and operates as a Zowe server extension component. Before planning the installation of UMS and IMS Administration Foundation, you must understand the relationship between Zowe and UMS.

The figure in "Architectural overview" in the *UMS User Guide* illustrates a high-level architectural overview of UMS, supported offerings, and the relationship between Zowe, UMS, and IMS Administration Foundation. You use UMS features through the web application interface called *IBM Unified Experience for z/OS*, which is built on top of Zowe Desktop. UMS also provides its own REST API services. IMS Administration Foundation leverages the UMS architecture and extends UMS user interface and REST API services for IMS system administration.

From an installation perspective, it is important to understand how UMS extends Zowe:

- The UMS server is a Java server that is installed as a Zowe server extension component. To learn about the Zowe server extension mechanism and how an extension component can be added to a Zowe server, see "Zowe z/OS extensions" in Zowe Docs.
- UMS extends Zowe Desktop by using the Zowe App Server application plug-in capability to provide IBM Unified Experience application on Desktop. To learn about the Zowe App Server application plug-in and the structure of the application plug-in, see "Plug-ins definition and structure" in Zowe Docs.
- UMS leverages Zowe by using plug-ins that reside in Zowe System Services (ZSS) server, Zowe cross memory server, and its auxiliary address space. ZSS is a Zowe component that is utilized by Zowe and interfaces with the Zowe cross memory server. UMS extends this capability by adding plug-ins to both ZSS and cross memory servers, and by installing various service routines that run in the auxiliary address space of the cross memory server. To learn about Zowe architecture including ZSS and cross memory servers, see "Zowe architecture" and "Zowe Auxiliary Address space" in Zowe Docs.

Capacity planning

The UMS (Java) server runs as a Zowe server extension component. Therefore, the UMS server must run on the same z/OS system as the Zowe launcher and Zowe core component servers such as Zowe App Server (Node.js), API Gateway, and ZSS server.

The UMS server can manage all IMS subsystems running on any z/OS system that is part of the same sysplex. To discover and manage IMS subsystems on other z/OS systems in the same sysplex, a Zowe cross memory server and its associated auxiliary address space must be running on each of the z/OS systems. For information about supported IMS releases and configurations, see "Software requirements for IMS Administration Foundation" on page 5.

For UMS server resource requirements when IMS Administration Foundation features are activated, see "Capacity planning" in the *UMS User Guide*.

Software requirements for UMS

Ensure that the software requirements for UMS are met.

Because a UMS server instance operates as an extension component of Zowe, Zowe is a key prerequisite for UMS. However, UMS does not require all Zowe server components. A UMS server shares some prerequisites with the Zowe core server components it uses.

For information about software requirements for UMS, compatibility between Zowe releases and UMS release levels, and Zowe server components required by UMS, see "Software Requirements for Unified Management Server" in the *UMS User Guide*. If any required software is missing, you must obtain it, then install and configure it before proceeding with the UMS installation.

Major dependencies and release specific considerations:

- Zowe 2.18.0 is the minimum recommended release for use with the UMS server.
- UMS 1.2.0.8 is the minimum UMS release that supports Zowe 2.18.0. Recommended UMS release is UMS 1.2.0.10 or later.
- IMS Administration Foundation 1.7.0.3 is compatible with UMS 1.2.0.9 or later. For details about the compatibility between UMS releases and IMS Administration Foundation, see "UMS PTF compatibility" in the *UMS User Guide*.
- If you are using z/OS 3.1 or later, see "Additional requirements for z/OS 3.1" in the *UMS User Guide*.
- If you plan to use Semeru 17 as the Java runtime for Zowe and UMS servers, see "Additional requirements for IBM Semeru Runtime Certified Edition for z/OS, version 17" in the *UMS User Guide*.
- If you plan to use Zowe 3.0.0 or later, UMS 1.2.0.9 or later is required. Also, Semeru 17 is required as the Java runtime for both Zowe and UMS servers. See "Additional requirements for Zowe V3 upgrade" in the *UMS User Guide*.

Unless there are special circumstances, install each product using the recommended release or a later release. Earlier releases have different configuration steps.

This information assumes that the following releases and maintenance levels of Zowe, UMS, and IMS Administration Foundation are used:

- Zowe 2.18.0 or later for Zowe Version 2, or Zowe Version 3.

  > **Note:** For supported releases of Zowe Version 3, see the *UMS User Guide*.

- UMS 1.2.0.10 or later
- IMS Administration Foundation 1.7.0.3 or later

Software requirements for IMS Administration Foundation

Ensure that your IMS configuration and environment meet the minimum software requirements for IMS Administration Foundation.

Minimum software requirements are described in "Software requirements for IMS Administration Foundation" in the *UMS User Guide*.

Some requirements are conditional, depending on IMS Administration Foundation features that you plan to use.

Installing UMS and its prerequisite software

Installing UMS and its prerequisite software involves creating and configuring a Zowe server instance, installing UMS with SMP/E, and installing IMS Tools Base and other IMS Tools products with SMP/E.

This step is typically performed by a z/OS system programmer, except for the security-related configuration tasks required to create and start a Zowe server instance.

This step consists of the following tasks:

1. “Creating and configuring a Zowe server instance for UMS” on page 6
2. “Installing UMS with SMP/E” on page 6
3. “Installing IMS Tools Base and other IMS Tools products with SMP/E” on page 7

Creating and configuring a Zowe server instance for UMS

Zowe is the primary prerequisite software for UMS. Set up a Zowe server instance that provides the environment required to install and run UMS.

About this task

Creating and validating a minimal Zowe instance before integrating UMS helps reduce configuration issues. This task guides you through creating a Zowe instance with a minimum set of Zowe core components required by UMS, verifying it, and setting up security for the Zowe instance.

Procedure

1. Set up the security for the Zowe instance.
   Follow the instructions in "Configuring Security" in Zowe Docs.

   After the security setup is complete and the Zowe z/OS runtime is initialized, configure the required certificates. Certificate configuration must be performed by a user with security administrator authority.
2. Create a Zowe instance with the minimum required components.
   UMS requires only the server components of Zowe. Create a Zowe instance using only the core server components that are required by UMS.

   Follow the instructions for installing Zowe server components in "Zowe server-side installation overview" and "Preparing for installation" in Zowe Docs.

   > **Note:** When viewing Zowe Docs, select your Zowe release from the drop-down menu at the upper-right of the Zowe Docs site to ensure you are viewing the correct information for your release.

3. Start the Zowe instance and verify that the Zowe instance is functioning as expected.
   Ensure that the instance starts without errors and that all necessary Zowe services are running.

Installing UMS with SMP/E

Install UMS by using SMP/E, and then apply required maintenance to UMS.

Before you begin

Ensure that a Zowe instance is running without issues. A UMS server instance runs as an extension component of Zowe and it shares common prerequisites with Zowe core server components that UMS uses. A functional Zowe instance confirms that all UMS prerequisite related to Zowe are satisfied.

Procedure

1. Install UMS.

Follow the instructions in "Performing SMP/E installation tasks for Unified Management Server" in the *UMS User Guide*.

You can disregard any information related to Db2 products.

2. Apply all available PTFs to UMS.
   Follow the instructions in "Installing a program temporary fix (PTF)" in the *UMS User Guide*.

Installing IMS Tools Base and other IMS Tools products with SMP/E

Install IBM IMS Tools Base for z/OS 1.7 (IMS Tools Base) and any other IMS Tools products by using SMP/E.

Procedure

1. Install IMS Tools Base.
   IMS Tools Base is an IMS Tools product that is required to use IMS Administration Foundation features on a UMS server instance.

   IMS Tools Base 1.7 provides two features:

   IMS Administration Foundation for z/OS (FMID: HAFN170)

   This feature is required for activating IMS Administration Foundation base features on a UMS server instance.

   IMS Tools Base (FMID: HAHN170)

   This feature is optional. Install HAHN170 if you plan to use any of the features that are listed in "IMS Administration Foundation and IMS Tools" in the *UMS User Guide*.

   Follow the instructions in the Program Directory for IBM IMS Tools Base for z/OS 1.7, and install HAFN170, and optionally HAHN170.

2. Install individual IMS Tools products.
   See "IMS Administration Foundation and IMS Tools" in the *UMS User Guide* and identify which IMS Tools products you will install. Then, follow the instructions in the Program Directory for each IMS Tools product. You can download Program Directories from IMS Tools Product Documentation.

3. Apply available PTFs to IMS Tools Base and individual IMS Tools products that you installed.

Preparing data sets and USS directories for UMS

Prepare the z/OS Unix System Services (USS) directories and MVS data sets required to configure and run a UMS server instance under the Zowe server instance.

About this task

UMS requires specific USS directories and MVS data sets. Follow the instructions in this topic to prepare the directories and data sets. You will set the permissions for the directories and data sets in a later step.

Procedure

1. Identify the UMS runtime directory.
   The UMS runtime directory is a USS directory where the UMS product deliverables are installed. This USS directory is created during SMP/E installation and should already exist. The default UMS runtime directory is `/usr/lpp/IBM/izp/v1r2m0/bin`.

   Record the directory path, as you will specify this path in the UMS configuration file later.

2. Create a USS directory for the UMS workspace.
   Create or mount a USS directory to be used as the UMS workspace directory. The UMS server instance will create files required for server operation and store individual users' persistent data in this directory.

   For more information about the UMS workspace directory, see "Item 2: Identify the UMS z/OS UNIX System Services directories" in the *UMS User Guide*.

   Record the directory path, as you will specify this path in the UMS configuration file later.

3. Create an MVS data set for UMS PARMLIB.
   Create an MVS partitioned data set (PDS) or PDSE with LRECL 80 to store configuration members for UMS and UMS extension products.

   Record the data set name, as you will specify the name of this data set in the UMS configuration member later.

4. Create an MVS data set for UMS JCLLIB.
   Create an MVS PDS or PDSE with LRECL 80 for storing JCL members. The JCL members will be created during the UMS configuration process from the templates installed by SMP/E. This data set must be separate from the UMS PARMLIB data set.

5. Determine a high-level qualifier for data sets that are dynamically created during UMS configuration.
   UMS will allocate MVS data sets during the configuration process. It is recommended to use IZP as the high-level qualifier for these data sets.

   Once you have determined the high-level qualifier, record the qualifier as you will later specify it in the UMS configuration member. The qualifier will then serve as the common high-level qualifier for dynamically created data sets.

6. Create an MVS data set and copy UMS sample and installation JCL members into it.
   Because you will modify sample members in the UMS sample data set (SIZPSAMP), create a copy of the members in a new data set.

   - Use the common high-level qualifier that you determined in Step "5" on page 8 (for example, IZP) for this data set.
   - Consider using INSTLIB as the lowest-level qualifier for the installation JCL data set to make it easily identifiable.

Configuring security for UMS and IMS Administration Foundation

Configuring security for a Zowe server instance is one of the most complex tasks of Zowe and UMS server configuration. However, this process is critical to ensure seamless communication between servers and clients, while protecting the system against potential threats.

Each Zowe component server, including the UMS server, must be configured to handle TLS-secured TCP communication channels. This setup involves setting up encryption protocols, managing digital certificates, and ensuring compliance with security policies. Additionally, the setup must account for various authentication mechanisms, access controls, and continuous monitoring to detect and respond to any security incidents.

Because of the complexity of these tasks, a deep understanding of z/OS security architecture and security best practices is essential.

The following STC IDs and user IDs are involved in the security configuration:

Table 2: STC IDs and user IDs used for security configuration

ID	Description
UMS STC ID	Typically the same as the Zowe STC ID.
Zowe Cross-Memory Server STC ID	Used by the Zowe cross-memory server component.
UMS installer user ID	The user ID of the user who configures UMS. **Note:** The UMS installer user ID is typically not the user ID that is used to install UMS with SMP/E or to apply PTFs to UMS. It is the user ID that is used to configure UMS after the SMP/E installation is complete.
UMS (default) DBA user ID	A machine user ID used by UMS for backend processing.
UMS login user IDs	General user IDs that are authorized to access UMS features.

The security setup for UMS and IMS Administration Foundation consists of various tasks. The following list covers the primary steps involved:

Setting up STC user IDs for UMS

In addition to the Zowe server started task (STC) user ID, the UMS server uses two types of user IDs; a UMS server STC user ID, and a Zowe cross memory server auxiliary address space user ID. You must set up these user IDs.

Procedure

1. Verify that the following required user IDs exist.
   - A UMS server STC user ID. This ID is the same as the Zowe server STC user ID.
   - A Zowe cross memory server auxiliary address space user ID. This ID can be the same as the Zowe cross memory server STC user ID.
2. Prepare RACF STARTED class profiles for the server STC user IDs.
   The user IDs for the server address space are determined by the RACF STARTED class profiles. Work with your z/OS security administrator to prepare the profiles.

Setting permissions for directories and data sets for Zowe and UMS

Set up the required privileges for the UMS installer user ID and the Zowe STC user ID so that the USS directories and MVS data sets that are used by Zowe and UMS can be accessed.

Procedure

1. Locate the Zowe configuration file, zowe.yaml.
   Follow the instructions in "Item 3: Confirm the location of Zowe configuration YAML file" in the *UMS User Guide*. You will use this file later when you configure UMS and IMS Administration Foundation.
2. Assign privileges for user IDs.
   The following tables list the required privileges for the UMS installer user ID and the Zowe STC user ID to access the USS directories and MVS data sets used by Zowe and UMS. Note that the UMS server STC user ID is the same as the Zowe STC user ID.

Table 3: Required privileges for data sets for Zowe and UMS

Data set	UMS installer user ID	STC user ID for Zowe and UMS	UMS (default) DBA user ID
Zowe runtime data sets (see Note a)	READ	READ	NONE
Zowe custom data sets (see Note b)	UPDATE (ALTER if the UMS installer allocates the data set.)	READ	NONE
UMS load module library SIZPLOAD	READ	NONE	NONE
UMS JCL templates library SIZPCUS*	READ	NONE	NONE
UMS parameters library SIZPPARM	READ	NONE	NONE
UMS samples library SIZPSAMP	READ	NONE	NONE
UMS load module library SIZPLOAD	READ	NONE	READ
UMS PARMLIB data set	UPDATE (ALTER if the UMS installer allocates the data set.)	UPDATE	NONE
UMS JCLLIB data set	UPDATE (ALTER if the UMS installer allocates the data set.)	NONE	NONE
UMS INSTLIB data set (UMS installation JCL data)	UPDATE (ALTER if the UMS installer allocates the data set.)	NONE	NONE
Data sets created during UMS installation	ALTER It is recommended that you create a generic profile for the HLQ that you chose for these data sets.	NONE	NONE

Data set	UMS installer user ID	STC user ID for Zowe and UMS	UMS (default) DBA user ID
IMS Administration Foundation EXEC data set SAFNEXEC (see Note c)	NONE	NONE	READ
IMS Administration Foundation temporary data sets (see Note c)	NONE	NONE	ALTER

Notes:

a. Zowe read-only runtime data sets are the data sets referred to as *runtime datasets* in "Sever Datasets Reference" in Zowe Docs.

b. Zowe custom data sets are the data sets that the `zowe.setup.dataset` parameter section of the zowe.yaml file specifies. For details, see "Initializing Zowe custom data sets" in Zowe Docs.

c. For details about the SAFNEXEC data set and temporary data sets for IMS Administration Foundation, see "Installing IMS Administration Foundation" in the *UMS User Guide*. This topic also provides information about additional permissions that are required for using specific IMS Tools integration functions with IMS Administration Foundation.

Table 4: Required bitmask (rwx) for USS directories for Zowe and UMS

Directory	UMS installer user ID	STC user ID for Zowe and UMS
Zowe USS file system path (default path: `/usr/lpp/zowe`)	Read and execute (r-x)	Read and execute (r-x)
Zowe instance workspace directory (default path: `/usr/lpp/IBM/izp/v1r2m0/bin`)	Read, write, and execute (rwx)	Read, write, and execute (rwx)
UMS runtime directory	Read and execute (r-x)	Read and execute (r-x)
UMS workspace directory	Read, write, and execute (rwx)	Read, write, and execute (rwx)

For information about the Zowe USS file system path, instance workspace, and custom data sets, see the following topics in Zowe Docs:

- For an overview of USS directories, MVS data sets, and address spaces that are used to run a Zowe instance, see "Preparing for installation" in "Installing Zowe server-side components".
- For details about SMP/E-based Zowe installation and associated library data set names (SZWEAUTH, SZWESAMP, and SZWEEXEC) and the USS file system path for Zowe, see "Installing Zowe SMP/E overview" and "Installing Zowe via SMP/E instructions". Zowe and UMS use the zwe command in the bin directory in the file system path at installation and during run time.
- For information about data sets that are used by Zowe, such as PARMLIB, JCLLIB, and ZWESAPL, see "Initializing Zowe custom data sets". The PARMLIB data set can be shared with other servers that are installed on the z/OS system. The parameter member for Zowe cross memory server will be placed in the PARMLIB data set and will be updated during the UMS configuration process.

For descriptions of UMS data sets and directories, see "Preparing data sets and USS directories for UMS" on page 8.

Setting up additional permissions for Zowe cross-memory server

UMS uses certain IMS-specific services that are provided by programs that run in the auxiliary address space of the Zowe cross-memory server. You must configure security permissions for the user ID associated with this auxiliary address space.

Procedure

- Follow the instructions in "Security setup for the auxiliary address space of Zowe cross-memory server" in the *UMS User Guide* to implement the required security configurations for the auxiliary address space user ID.

How UMS uses certificates

The UMS server uses digital certificates to support secure communication and authentication. Learn about the certificate requirements and how to specify certificates in Zowe and UMS configurations.

Digital certificates used by UMS

UMS uses digital certificates for two main purposes:

Secure Communication (TLS)

- HTTPS for UMS API calls: Encrypts communication between UMS API clients and the UMS server.
- Internal or backend server connections: Secures connections between the UMS server and other Zowe components or non-Zowe backend servers by using HTTPS or TLS connections.
- Mutual TLS (mTLS): Enables both the client and the server to authenticate each other using certificates. Mutual TLS can be used for internal communication between Zowe components and for connections from the UMS server to backend servers.

Certificate-based authentication

- UMS DBA users: UMS DBA users are machine users used for UMS backend processing. Each of these users can be authenticated by using a digital certificate instead of a password.

Default certificate configuration in UMS

By default, UMS uses the keystore and truststore defined in the zowe.yaml configuration file for secure communication with Zowe server components and UMS backend servers. A configuration example is available in the later step "Setting up network security for the UMS server" on page 15.

Servers that can communicate with UMS with mutual TLS

In IMS Administration Foundation, the UMS server acts as a client for the following backend servers:

- A Zowe ZSS server
- A Zowe API Gateway server (if UMS authType=MFA_JWT is used)
- A z/OSMF server
- One or more IMS Connect servers
- One or more IMS Tools Base TCP servers (if one or more IMS Tools integration features are used)

If any of the following backend servers use mutual TLS, it is highly recommended that you use a UMS keystore that can be commonly used as the client keystore for the mutual TLS connections with those servers:

- A Zowe ZSS server
- A Zowe API Gateway server
- A z/OSMF server

The client certificate in the keystore must be signed by a root CA that is trusted by all target servers. You must specify the keystore and certificate in the UMS configuration parameter section `components.izp.security.certificate.keystore.genericClientKeyStore` (supported in UMS 1.2.0.10 and later). If the Zowe certificate that you configured in the `zowe.certificate.keystore` section in the zowe.yaml file satisfies both of the following conditions, you can specify the same keystore and certificate in the `genericClientKeyStore` parameter section for the UMS configuration:

- The certificate in the keystore is signed by a root CA that is trusted by all target servers.
- The certificate in the keystore is not mapped to any specific user ID.

Considerations for genericClientKeyStore: The certificate that is specified in the `genericClientKeyStore` parameter section of the UMS configuration must be generic. A generic certificate is not mapped to a specific user ID and is not used to authenticate a UMS login user. The certificate is used only to establish secure connections.

- If mutual TLS is configured for the server port of the Zowe ZSS server, the z/OSMF server, or the IMS Tools Base TCP server, you must specify the `genericClientKeyStore` parameter section in the UMS configuration. UMS sends the specified certificate during the mutual TLS handshake, and the login user is authenticated by a password or a passticket.
- If mutual TLS is configured for the z/OSMF port and the UMS authentication type is set to MFA_JWT (authType=MFA_JWT), the UMS server first connects to the API Gateway by using the certificate specified in the `genericClientKeyStore` section. The API Gateway then uses the Zowe certificate to connect to the z/OSMF server and authenticates the login user ID with a passticket.

Restriction: If any IMS Connect servers used by IMS Administration Foundation are configured for mutual TLS, ensure that the certificate used by Zowe is signed by a root CA that is trusted by all those IMS Connect servers. The keystore that is specified in the parameter section `components.izp.security.certificate.keystore.genericClientKeyStore` is not used for mutual TLS connections to IMS Connect server ports.

Authentication methods for UMS DBA user ID

The UMS server uses a machine user ID called the UMS DBA user ID to connect to UMS backend servers for unattended processing. IMS Administration Foundation uses a single UMS DBA user ID (the default UMS DBA user ID) for all IMS subsystems and data sharing groups. The IMS resource access permissions required for the UMS DBA user ID are described in "Security setup for IMS Administration Foundation" and "Security setup for IMS Administration Tool" in the *UMS User Guide*. Details about the UMS default DBA user are described in "Setting up DBA user for Unified Management Server" in the *UMS User Guide*. The following two authentication methods are supported by UMS.

Certificate-based authentication (recommended)

Certificate-based authentication for the UMS DBA user ID is supported for connections from the UMS server to the following backend servers:

- A Zowe ZSS server
- A z/OSMF server
- One or more IMS Tools Base TCP servers

  Requirement: All IMS Tools Base TCP server ports must be configured with ClientAuthType=SAFCheck by using AT-TLS.

For instructions on setting up the UMS DBA user ID and its certificate, see "Setting up the default UMS DBA user ID" on page 14. For details about setting up the IMS Tools Base TCP server port for certificate-based authentication for the UMS DBA user, see "Setting up AT-TLS SAFCheck client authentication for DAI TCP server" in the *UMS User Guide*.

Password-based authentication

You can configure the UMS DBA user ID to authenticate with a password instead of a certificate. However, this method is not preferred due to password management concerns.
If you choose to use password-based authentication, the considerations listed in Considerations for genericClientKeyStore also apply to the UMS DBA user ID.

Important: Certificate mapping considerations

- Do not map the certificate specified in zowe.yaml or the `genericClientKeyStore` parameter section to any specific user.

- The certificate for the UMS DBA user must be mapped to a specific user. Therefore, the certificate must be different from the certificate specified in zowe.yaml or the UMS `genericClientKeyStore` parameter section.

Important: Certificates have expiration dates

Certificates are not permanent. Each certificate has an expiration date. After a certificate expires, it becomes invalid and cannot be used for TLS communication or user authentication.

When a certificate expires, users might receive error messages that do not clearly indicate the cause. For example, the message might simply state that authentication failed, without explaining that the certificate has expired.

To avoid unexpected authentication failures or service disruptions:

- Monitor certificate expiration dates regularly.
- Set up reminders or automation to renew certificates before they expire.
- Use certificate management tools that support automatic renewal and deployment.

Proactive certificate management helps ensure reliable and secure communication and authentication.

Setting up the default UMS DBA user ID

Set up the default DBA user ID for UMS.

About this task

The UMS server uses a special machine user ID called *UMS DBA user ID*. This user ID is used for UMS server backend processing for IMS and Db2 subsystems. While UMS extension products for Db2 support subsystem-level UMS DBA user ID, IMS Administration Foundation uses only the default UMS DBA user ID.

Procedure

- Set up the default UMS DBA user ID.
  For details, see the following topics in the *UMS User Guide*:
  - "Setting up DBA user for Unified Management Server"
  - "Configuring the DBA user by using certificate"

  You will later specify this setting in the UMS configuration file.

  > **Recommendation:** Use certificate-based authentication for the default UMS DBA user. By using certificate-based authentication, frequent password updates or non-expiring password for the default DBA user ID can be avoided.

Setting up resource profiles for ICSF services

Zowe and UMS require the IBM z/OS Integrated Cryptographic Service Facility (ICSF) server to be active and accessible to perform cryptographic operations. ICSF provides services such as encryption, decryption, key management, and other security-related functions, regardless of whether cryptographic hardware is used.

About this task

UMS uses the Zowe ZSS server regardless of whether Zowe desktop UI is used. The ZSS server always uses ICSF services, making ICSF a mandatory requirement for UMS.

UMS itself uses ICSF services conditionally. Some ICSF services are used when either or both of the following optional features are used:

- Using a UMS DBA user ID with password authentication (not certificate-based authentication)
- Using the Java keystore type JCECCARACFKS

The Zowe STC ID, which is also the UMS STC ID, requires permissions for certain resource profiles in the RACF CSFSERV class (when the class is active), either explicitly or through generic resource profiles.

Procedure

- Authorize the Zowe STC ID for required resources in the RACF CSFSERV class. For details about required configurations and associated permissions, see the following information:
  - For ICSF dependencies and permissions related to Zowe, see "Configure an ICSF cryptographic services environment" in Zowe Docs.
  - For ICSF dependencies related to UMS, see "Integrated Cryptographic Service Facility (ICSF) validation" in the *UMS User Guide*.
    - For required permissions for CSFSERV class resources and CRYPTOZ class resources when password authentication is used for the UMS DBA user ID, see "Failure in DBA configuration izp-encrypt-dba.sh" in the *UMS User Guide*.
    - Required permissions for CSFSERV class resources when using the Java keystore type JCECCARACFKS depend on the cryptographic algorithms used for key materials that are stored in the ICSF PKDS. Consult your security administrator about the cryptographic algorithms and set up the necessary CSFSERV resource permissions for the Zowe STC ID.
  - For the relationship among ICSF cryptographic functions, ICSF callable services, and required CSFSERV resources, see "RACF CSFSERV resource requirements" in *z/OS Cryptographic Services System Secure Sockets Layer Programming*.

Setting up network security for the UMS server

To secure communication, configure the keystore and truststore for the UMS server. The recommended approach is to use the configuration options described in this topic.

Procedure

1. Decide whether to use the same keystore and truststore for both Zowe and UMS.
   The recommendation is to use the same keystore and truststore for both Zowe and UMS.
   - UMS runs as a component of Zowe. If you use the same keystore and truststore for both Zowe server and the UMS server, you do not need to define the `components.izp.security.certificate` section of the UMS configuration file that is mentioned in "Setting up secure communication for UMS" in the *UMS User Guide*.
   - Ensure that the key ring and the certificate used for Zowe and UMS servers meet the required conditions. See the "Important" notice in "Setting up secure communication for UMS" in the *UMS User Guide*.

   If you choose to use different keystore and truststore for Zowe and UMS, follow the instructions in "Setting up secure communication for UMS" in the *UMS User Guide*.
2. Prepare a key ring for Zowe keystore and truststore.
   - To generate a key ring from existing certificates by using the Zowe zwe init certificate command, see "Certificate configuration scenarios" in Zowe Docs.
   - To use an existing key ring, specify the key ring type, name, and the certificate label in the configuration. See "YAML configurations – certificate" in Zowe Docs.

   The following figure shows an example `zowe.certificate` section of the Zowe configuration file zowe.yaml when a key ring is used for both keystore and truststore:

```
zowe:
  ...
  certificate:
    keystore:
      type: "JCERACFKS"
      file: "safkeyring:////ZWESTC/ZoweKeyring"
      password: "password"
      alias: "certificate label"
    truststore:
      type: "JCERACFKS"
```

```
      file: "safkeyring:////ZWESTC/ZoweKeyring"
      password: "password"
    pem:
      key: ""
      certificate: ""
      certificateAuthorities: "safkeyring:////ZWESTC/ZoweKeyring&rootca"
...
```

3. Set up a CA certificate in the key ring of the backend server.
   If TLS client authentication is configured by the AT-TLS policy for UMS backend server ports (IMS Connect server port or IMS Tools Base TCP server port), you must add the appropriate Certificate Authority (CA) certificate to the key ring used by each UMS backend server as its truststore.

   The certificate must be the root CA certificate in the signing chain for the UMS server certificate connected to the keystore that is used for the UMS server.

   The SAF privilege considerations for specific IRR.DIGICERT resources in the FACILITY class as described in "Setting up TLS client authentication for IMS Connect servers" in the *UMS User Guide* apply to both IMS Connect servers and IMS Tools Base TCP servers.

Preparing a RACF class for UMS resource profiles

Create a RACF class IZP for UMS resource profiles.

Procedure

- Create a RACF class IZP for UMS.
  During the UMS configuration process, a JCL member is generated to assist with creating the IZP class. You can use this JCL member to define the class. Alternatively, if you want your z/OS security administrator to create this class before you start the UMS configuration process, follow the instructions in "Defining a security class for UMS" in the *UMS User Guide*.

Setting up UMS login users with their roles and privileges

UMS login user IDs are used to log in to Zowe Desktop and use the Zowe app IBM Unified Experience for z/OS provided by UMS. These user IDs are also used to interact with the REST APIs provided by the UMS server. Many features of IBM Unified Experience app rely on these REST APIs.

About this task

UMS login user IDs are classified into two types:

UMS super administrator user IDs

These user IDs are authorized through a SAF profile defined in the IZP class. UMS super administrators have elevated privileges and can perform UMS server management tasks such as subsystem registration. A UMS super administrator must also be a valid Zowe user.

UMS regular user IDs

These user IDs are authorized through another SAF profile in the IZP class. UMS regular users can access the common core features of UMS. Additional permissions might be required for features specific to IMS or Db2. A UMS regular user must also be a valid Zowe user.

Procedure

- Create resource profiles and define users by using the profiles.
  UMS login user IDs can be existing user IDs, but they must be associated with specific resource profiles that are defined in the RACF IZP class for UMS.

  Follow the instructions in the following topics in the *UMS User Guide*:

  - "Setting up users and teams"
  - "UMS roles and responsibilities"

Important: It is recommended to manage users and teams using SAF-based management by specifying true for the `components.izp.security.useSAFOnly` parameter for UMS.

Configuring UMS and IMS Administration Foundation

Complete the following UMS and IMS Administration Foundation configuration steps to enable IMS Administration Foundation features.

Before you start, make sure that you have completed the following tasks:

- “Installation planning” on page 4
- “Installing UMS and its prerequisite software” on page 6
- “Preparing data sets and USS directories for UMS” on page 8
- “Configuring security for UMS and IMS Administration Foundation” on page 9

Steps:

1. “Preparing the UMS configuration member” on page 18
2. “Editing the UMS configuration member” on page 19
3. “Validating the UMS configuration member” on page 19
4. “Conditionally required steps for Zowe and UMS customization” on page 19
5. “Installing UMS plugins to Zowe server components” on page 20
6. “Customizing the Zowe server PROC for UMS” on page 21
7. “Validating the UMS configuration” on page 21
8. “Configuring IMS Administration Foundation” on page 22
9. “Validating IMS Administration Foundation configuration” on page 22

Preparing the UMS configuration member

Prepare the ZWEYAML configuration member for UMS.

Procedure

1. Shut down the Zowe main server and its all cross memory servers on all z/OS systems in the sysplex.

   > **Important:** All cross memory servers must be stopped because a runtime MVS load module library used by the cross memory servers is updated during the UMS configuration process.

2. If the UMS PARMLIB data set does not exist, allocate the UMS PARMLIB data set by editing and submitting IZPALOPL JCL in the INSTLIB data set.

   > **Note:** The INSTLIB data set is a copy of the UMS SMP/E data set SIZPSAMP that you created in “Preparing data sets and USS directories for UMS” on page 8.

   For information about which values to modify in the JCL, see Step 4 in "Step 2: Installing Unified Management Server" in the *UMS User Guide*.

3. Create the UMS configuration member, ZWEYAML, in the UMS PARMLIB data set by editing and submitting the IZPCPYML JCL in the INSTLIB data set.
   The IZPCPYML job copies member IZPYAML from the UMS SMP/E data set SIZPPARM to the UMS PARMLIB data set as member ZWEYAML.

   For information about which values to modify in the JCL, see Step 5 in "Step 2: Installing Unified Management Server" in the *UMS User Guide*.

4. Update the existing ZWEYAML member by editing and submitting the IZPSYNCY JCL.

The IZPSYNCY JCL updates the existing ZWEYAML member with new configuration parameters introduced by PTFs.

For information about which values to modify in the JCL, see Step 1 in "Installing a program temporary fix (PTF)" in the *UMS User Guide*.

Editing the UMS configuration member

Edit the ZWEYAML configuration member to specify the required configuration parameters for UMS.

Procedure

- Edit the UMS configuration member ZWEYAML and specify the required configuration parameters.
  Refer to the list of parameters in "Parameters in ZWEYAML" in the *UMS User Guide*. You can also refer to the member IZPYAML in the SIZPPARM data set for detailed descriptions of parameters.

  Also refer to the following information, which summarizes recommended UMS configuration parameters when only IMS Administration Foundation is used with UMS.

  `components.izp.security.dba.defaultAuthenticationMechanism` parameter

  Specify CERTIFICATE for this parameter. If you specify CERTIFICATE, you do not need to configure the `components.izp.security.pkcs11` section, except for the `components.izp.security.pkcs11.dbaUser` parameter, which you will later specify.
  For information about the `components.izp.security.dba` section, see "Setting up DBA user for Unified Management Server" in the *UMS User Guide*.

  `components.izp.server.javaArgs` parameter

  You do not need to specify any parameters explained in "Registering Db2 applications" in the *UMS User Guide*.

  `components.izp.dataset.loadlibary.db2` parameter

  Remove this parameter. This parameter is for Db2 features of UMS.

  `components.izp.toolsDiscovery.enabled` parameter

  Specify false for this parameter. The `components.izp.toolsDiscovery` section is for Db2 features of UMS.

Validating the UMS configuration member

Validate the UMS configuration parameters in member ZWEYAML by generating a set of configuration JCL members from the UMS configuration member.

Procedure

- Generate a set of configuration JCL members by editing and submitting the IZPGENER JCL in the INSTLIB data set.
  The IZPGENER job generates JCL members in the JCLLIB data set that is determined by the parameters specified in the UMS configuration member ZWEYAML.

  Follow the instructions for IZPGENER JCL in "Step 2: Installing Unified Management Server" in the *UMS User Guide*.

  A successful job completion indicates that the UMS configuration parameters in member ZWEYAML are customized correctly.

Conditionally required steps for Zowe and UMS customization

A set of configuration JCL members is created by running the IZPGENER JCL. Determine which JCL members you need to run, then submit them.

About this task

If both of the following conditions are met, you can skip the procedure steps in this section:

- Parameter `components.izp.security.useSAFOnly` is set to true.
- Parameter `components.izp.security.dba.defaultAuthenticationMechanism` is set to CERTIFICATE.

You only need to use the following two JCL members. However, do not submit them yet; you will be instructed to run these members in subsequent topics.

- IZPIPLUG: Installs Zowe plugins using the Zowe CLI command zwe.
- IZPEXPIN: Prepares or updates configuration members for IMS Administration Foundation in the UMS PARMLIB data set.

If either or both conditions are not met, complete the following procedure.

> **Note:** The instructions in this information apply to Zowe 2.18.0 or later and UMS 1.2.0.10 or later. If you are using an earlier release of Zowe or UMS, consult the *UMS User Guide* to determine which JCLs to use. Follow the instructions for the generated JCL members in "Step 2: Installing Unified Management Server" in the *UMS User Guide*.

Procedure

- `components.izp.security.useSAFOnly` is set to false
  Follow the instructions for IZPA1, IZPA2, and IZPUSRMD JCL members in this order in "Step 2: Installing Unified Management Server" in the *UMS User Guide*.
- `components.izp.security.dba.defaultAuthenticationMechanism` is set to PASSWORD
  Submit the IZPA3 JCL and then execute the USS shell script izp-encrypt-dba.sh by following the instructions in "Step 2: Installing Unified Management Server" in the *UMS User Guide*.

Installing UMS plugins to Zowe server components

UMS extends the functionality of Zowe by using several plug-ins. These plug-ins include plug-ins for the Zowe App server, the Zowe ZSS server, and the Zowe cross memory server. You must install these plug-ins.

About this task

Configure Zowe so that the UMS server runs as a component of Zowe and uses Zowe infrastructure to provide services that are specific to UMS or its extension products, such as IMS Administration Foundation.

For details about how Zowe can be extended using various plugins for its components and what the IZPIPLUG JCL member does, see "Plug-ins in ZSS, ZIS, or AUX" in the *UMS User Guide*.

Procedure

- Install the plug-ins to Zowe server components by submitting the IZPIPLUG JCL member in the JCLLIB data set.
  You do not need to modify the IZPIPLUG JCL member.

  > **Important:** The IZPIPLUG job updates the Zowe instance USS directory and the Zowe cross memory server load module libraries.

  Follow the instructions for the IZPIPLUG JCL member in "Step 2: Installing Unified Management Server" in the *UMS User Guide*. If you are setting up the recommended release or a later release of Zowe (2.18.0 or later) and UMS (1.2.0.10 or later), you do not need to specify the IZP_YAML parameter in the STDENV DD statement of the IZPIPLUG JCL. You can ignore all explanations for lower levels of Zowe and UMS.

  If you encounter any issues with the IZPIPLUG job, see "Issues with the IZPIPLUG" in the *UMS User Guide*.

Customizing the Zowe server PROC for UMS

To start the UMS server instance as an extended component of the Zowe server, update the Zowe started task PROC. The Zowe launcher program reads the UMS configuration member ZWEYAML in the UMS PARMLIB as an extension to the Zowe base configuration file (default file name: zowe.yaml).

Procedure

1. Update the Zowe started task, ZWESLSTC.
   If the existing CONFIG parameter in the Zowe started task PROC is:

   ```
   CONFIG=FILE(/path/to/zowe.yaml)
   ```

   Change it as follows:

   ```
   CONFIG=PARMLIB(ums_parmlib(ZWEYAML))\
   :FILE(/path/to/zowe.yaml)
   ```

   Where *ums_parmlib* is the data set name of the UMS PARMLIB.

   Ensure that the following environment variable is specified in the STDENV DD statement of the PROC:

   ```
   _CEE_ENVFILE_CONTINUATION=\
   ```

   For details, see the instructions related to ZWESLSTC in "Step 2: Installing Unified Management Server" in the *UMS User Guide*. If you are setting up the recommended release or a later release of Zowe (2.18.0 or later) and UMS (1.2.0.10 or later), you can ignore all explanations for lower levels of Zowe and UMS.
2. Restart all Zowe cross memory servers.
3. Restart the Zowe main server.

Validating the UMS configuration

If your UMS configuration is successful, you can access the Zowe URL, log in to Zowe, and view and start the UMS web interface.

About this task

This step confirms that your UMS server configuration is successful and that Zowe is running with an UMS server instance as a Zowe extension component.

Procedure

- Log in to UMS using one of the UMS super administrator IDs.
  Follow the instructions in "Step 3: Validating the Unified Management Server installation" in the *UMS User Guide*.

  The following additional information might help you while you follow the instructions:

  Message IZPPL0028E (Step 3)

  If you see message IZPPL0028E while starting the IBM Unified Experience app on the Zowe Desktop, it indicates that the Zowe login user lacks UMS user privileges.

  Verify user privileges by following the instructions in "Setting up UMS login users with their roles and privileges" on page 16.

  No IMS subsystem listed or some IMS subsystems missing (Step 4)

  When you select a subsystem type of IMS to display the list of IMS subsystem across all z/OS systems in the sysplex but you see no IMS subsystems or some IMS subsystems missing, check the Discovered tab page for IMS for any IZPDI*xxxxxx* or IZPMS*xxxxxx* messages. If such messages are present, look up the meaning of the messages in the *UMS User Guide* and follow the instructions in the user response sections.

Configuring IMS Administration Foundation

To activate the IMS Administration Foundation features, update the UMS configuration member ZWEYAML in the PARMLIB data set.

Procedure

1. Shut down the Zowe main server.
   You do not need to shut down any cross memory servers.
2. Apply all available PTFs to the IMS Administration Foundation installation.
   Ensure that PTFs are applied before you modify the UMS configuration member.
3. Edit the UMS configuration member ZWEYAML in the PARMLIB data set to modify the following configuration parameter: `components.izp.experiences`
   To activate IMS Administration Foundation or any UMS extension products (referred to as *experiences*), specify the runtime directories of the extension products as array values.

   For example, if you use the default installation directory path for IMS Administration Foundation, the parameter might look like the following example:

   ```
   components:
     izp:
       enabled: true
       debugShellScripts: false
       experiences:
       - /usr/lpp/IBM/afn/v1r7m0/bin
   ```

   This example assumes that the IMS Administration Foundation feature (FMID: HAFN170) was installed by using the default installation path. The directory path must be the parent directory of the file system path `/usr/lpp/IBM/afn/v1r7m0/bin/IBM`, which was specified for the DD name SAFNBIN in the SMP/E DDDEF JCL for FMID HAFN170.
4. Submit the IZPEXPIN JCL in the INSTLIB data set.
   This job generates IMS Administration Foundation configuration members in the UMS PARMLIB data set.
5. Install IMS Administration Foundation.
   Follow the instructions in "Installing IMS Administration Foundation" in the *UMS User Guide* to setup security for IMS Administration Foundation and reconfigure UMS with the IMS Administration Foundation configuration members in the UMS PARMLIB data set.
6. Optional: Configure IMS components and IMS Tools servers.
   Depending on the IMS Administration Foundation features to be used, you might need to set up additional security. For details, see item 4 of the "Before you begin" section in " Installing IMS Administration Foundation" in the *UMS User Guide*.
7. Restart the Zowe main server.

What to do next

After you have configured IMS Administration Foundation,submit the IZPEXPIN JCL again after applying one or more PTFs to IMS Administration Foundation (FMID: HAFN170). For information about applying a PTF, see "Installing IMS Administration Foundation PTF" in the *UMS User Guide*.

Always create a backup copy of the UMS PARMLIB data set or backup copies of IZPIMSPM and IZPIMFPM members of the UMS PARMLIB data set before submitting the IZPEXPIN JCL.

> **Important:** If your UMS installation is at maintenance level 1.2.0.9 or lower, do not submit the IZPEXPIN JCL again.

Validating IMS Administration Foundation configuration

Verify that you have successfully configured IMS Administration Foundation by registering an IMS data sharing group.

Procedure

- Register an IMS data sharing group.
  Follow the instructions in "Registering IMS subsystems as a data sharing group" in the *UMS User Guide*.

  If you receive any warning or error messages during registration, look up the meaning of the messages in the *UMS User Guide* and follow the user response section of the messages. If the problem persists, open a support case.

Troubleshooting

If you encounter issues while configuring a UMS server instance or IMS Administration Foundation, you must gather diagnostic information before you report them to IBM Software Support.

See "Troubleshooting" in the *UMS User Guide* before opening a support case. The troubleshooting topic explains how to gather diagnostic information and describes common configuration issues along with their possible causes and solutions.