IBM Unified Management Server for z/OS 1.2

User Guide

IBM Unified Experience for z/OS interface for IBM Db2 Tools and IBM IMS Tools

SC27-9151

Note:

Before using this information and the product it supports, read the information in "Notices" on page 269.

2023-12-22 edition

This edition applies to Unified Management Server 1.2 (program number 5698-UM1), and to all subsequent releases and modifications unless otherwise indicated in new editions. It also applies to the installed software that Unified Management Server supports.

Contents

About this information

This information provides instructions for installing and using IBM® Unified Management Server for z/OS® (also referred to as Unified Management Server or *UMS*) and software products that leverage the UMS architecture, such as IBM Db2® Administration Foundation for z/OS and IBM Db2 DevOps Experience for z/OS, or simply *data management products*.

Each *data management product* is a separately delivered program that offers a core set of functionality that users in z/OS roles, such as system administrators, database administrators, or application developers, can use to perform tasks. You can use these products through the modern web application interface called *IBM Unified Experience for z/OS*.

This documentation applies to the following component product versions:

- IBM Unified Management Server for z/OS 1.2.0
- IBM Db2 Administration Foundation for z/OS 1.2.0
- IBM Db2 DevOps Experience for z/OS 1.3.0
- IBM IMS Administration Foundation for z/OS in IBM IMS Tools Base for z/OS 1.7.0

The topics presented in this document are designed to help database administrators, system programmers, and application programmers perform the following tasks:

- Plan for the installation of IBM Unified Management Server for z/OS and any available data management software.
- Install and configure IBM Unified Management Server for z/OS and any available data management software.
- Manage Db2 for z/OS by using IBM Db2 Administration Foundation for z/OS.
- Manage IMS by using IBM IMS Administration Foundation for z/OS.
- Develop z/OS applications by using IBM Db2 DevOps Experience for z/OS.

Chapter 1. Overview

An organization's data is its core business asset. In today's digital and data-driven economy, the needs placed on data management are challenged to meet the needs of a growing and changing environment. Technology and data management architecture must respond to address these needs. The overall IBM data management offerings provide modern methods of working with and managing Db2 for z/OS, IMS, and its data. There are many unique and significant features that appeal to users of all levels of experience and different roles within your organization.

Architectural overview

You can use the features of the IBM Unified Management Server for z/OS by using the web application interface called IBM Unified Experience for z/OS, which provides modern ways of working with and managing Db2 for z/OS, IMS, and their data.

The main components that comprise the Unified Management Server product family are:

- IBM Unified Management Server for z/OS
- IBM Db2 Administration Foundation for z/OS
- IBM IMS Administration Foundation for z/OS, a component of IBM IMS Tools Base for z/OS.
- Mainframe software products that leverage the UMS architecture, such as IBM Db2 DevOps Experience for z/OS.
- IBM Unified Experience for z/OS (Zowe-based graphical interface)

The following figure illustrates a high-level architectural and offerings overview. It shows the relationship of these components:

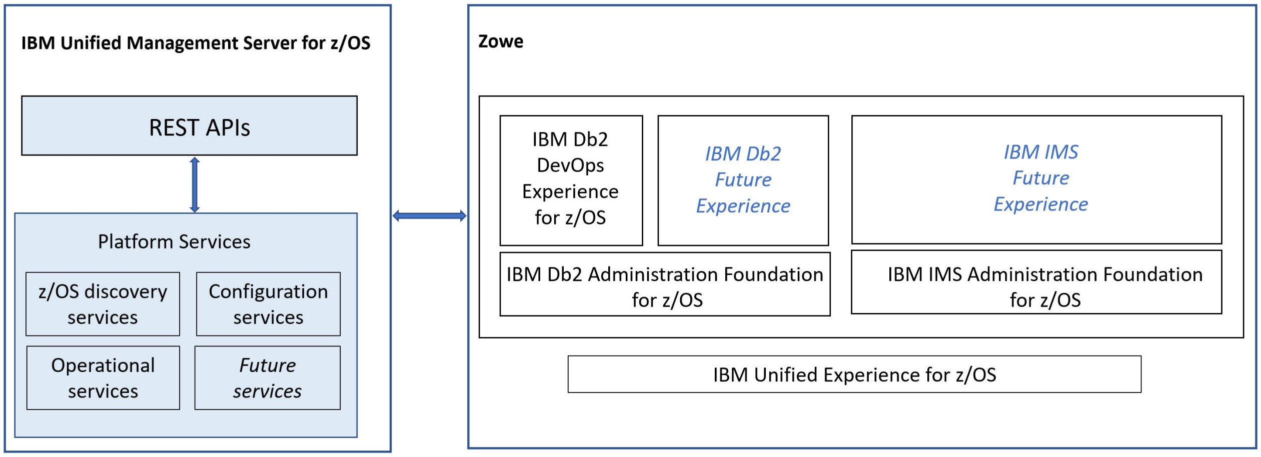

IBM Unified Management Server for z/OS
REST APIs
Platform Services
z/OS discovery services
Configuration services
Operational services
Future services
Zowe
IBM Db2 DevOps Experience for z/OS
IBM Db2 Future Experience
IBM IMS Future Experience
IBM Db2 Administration Foundation for z/OS
IBM IMS Administration Foundation for z/OS
IBM Unified Experience for z/OS

Figure 1. Relationship among IBM Unified Management Server for z/OS, data management products, and IBM Unified Experience for z/OS

IBM Unified Management Server for z/OS

The IBM Unified Management Server for z/OS (UMS) is the platform designed for modern mainframe data management. Its extensible capabilities can support IBM Db2 for z/OS and IMS subsystems.

Unified Management Server is the architecture server that provides all the common functions needed by the data management products that are installed and accessed, such as IBM Db2 DevOps Experience for z/OS.

Database as a Service (DBaaS)

Database as a Service (DBaaS) allows users to access and use part of a database system that adheres to established protocols and standards without the user having to separately install or re-create those services.

z/OS discovery services

When you install Unified Management Server, it securely "discovers" your mainframe environment, including subsystems, users, and other data, and displays that information through the user interface (IBM Unified Experience for z/OS) to help you get started.

Configuration services

The UMS configuration registry creates, stores, and manages policies and objects used in installed supported software. You manage and control various settings such as subsystems, environments, and teams through the Unified Experience (UI). UMS super administrators can create system environments that are configured with registered subsystems and assign them to application development teams that need to use the objects and resources of those registered subsystems.

Operational services

By using Db2 DevOps Experience with IBM Unified Management Server for z/OS, developers can group sets of database objects into applications, which establish these objects under source code control within Git. On-demand and self-service, they can provision instances of these applications into assigned environments and make database object changes as necessary in accordance with predefined site rules. These changes can be reviewed and merged into the originating application by issuing a pull request. Team administrators review, approve or decline, and merge the object changes.

REST APIs

REST APIs provide communication to and from the Unified Management Server and the installed data management products. REST services are also provided for user-designed workflows based on the specific installed application software.

For example, if you are using IBM Db2 DevOps Experience for z/OS, you can use REST APIs to automate development processes by using continuous integration and delivery tools such as IBM UrbanCode® Deploy and Jenkins.

The benefits of the Unified Management Server include:

- Common elements accessed by multiple installed data management products reduce overall maintenance and software management.
- Reduces amount and effort of installation and maintenance of code that would otherwise be required by multiple software applications.
- A single point of reference for common functions leads to easier problem detection and faster resolution.

IBM Db2 Administration Foundation for z/OS

The IBM Db2 Administration Foundation for z/OS is the foundation for managing Db2 for z/OS. It is a complimentary offering that provides several of the basic features and functions for different types of Db2 users, such as database administrator, application developer or administrator.

Db2 Administration Foundation is a complimentary offering that provides you an intuitive and powerful search facility. You need not separately install Db2 Administration Foundation. You can use its features even when you have installed only Db2 DevOps Experience with UMS. You can perform the following tasks:

- Search with or without filters across the entire sysplex to retrieve the Db2 objects you need.
- Drill down on any of the retrieved items, represented with a visual object type, to find related objects or issue Db2 commands or SQL statements.
- Simplify your administrative tasks by using an intelligent SQL editor.

IBM IMS Administration Foundation for z/OS

The IBM IMS Administration Foundation for z/OS is a component of IBM IMS Tools Base for z/OS. IMS Administration Foundation provides the foundation for managing IMS. It provides several of the basic features and functions for IMS system administrators and IMS database administrators.

IMS Administration Foundation provides capability of viewing IMS system and resource statuses. You can use its features after installing IMS Tools Base and activating the IMS Administration Foundation features on UMS. With IMS Administration Foundation, you can perform the following tasks:

- Simplify your administrative tasks by using an IMS command processor and a SQL processor.
- Drill down IMSplex components and IMS online resources to see their status and find their relations, or issue IMS commands.
- Search the entire sysplex to retrieve information on IMS DBD, PSB, DBRC-defined groups, and IMS online resources and their relationships.

If you also installed some of IMS Tools products, extended features such as the following become available:

- Viewing IMS Tools utility reports for databases, HALDB partitions, and DEDB areas.
- Visualizing the database segment structure and segment relationships defined in DBDs and PSBs.
- Detecting threshold exceptions for some selected database space statistics and reviewing those exceptions, history, and the trend of statistics associated with the exceptions.
- Detecting if a database reorganization or an image copy is needed based on the pre-defined criteria.

Data management experiences

Mainframe data management software typically requires a large investment to learn the various settings, configurations, tuning parameters, options, and so on, that have been added to the system over years of operation.

Complex individual tooling makes the learning curve steep for new users, while increasing the potential for errors and time to respond. Individual products lack the ability to share technology and data across individual product boundaries. The modernization of mainframe tooling described in this document removes these limitations.

Each software product that is supported by UMS has its own features and interacts with its own relevant z/OS data, but it is unified in the domain area it covers such as DevOps, administration, and performance. This allows for sharing between software and across domains, depending on what software is installed and what role the user has in the organization.

The benefits include:

- Logical components that correspond to different mainframe applications are controlled and maintained within the respective software.
- Allows each data management product to independently receive frequent updates with new features and enhancements.

IBM Db2 DevOps Experience for z/OS

Db2 DevOps Experience is a separate data management experience that you can install and activate on top of Unified Management Server. It is a browser-based graphical user interface with its own set of features and functionality that you can use to perform tasks for z/OS subsystems.

This role-based product provides users modern methods of working with and managing Db2 for z/OS. By using the multiple features available in this product, developers can easily provision instances, create applications, make changes, and send them for approval.

You can access services, such as discovery and operations by using IBM Unified Management Server for z/OS and you can access all the features of Db2 DevOps Experience using the browser-based user interface IBM Unified Experience for z/OS.

IBM Unified Experience for z/OS

The IBM Unified Experience for z/OS is a browser-based user interface that is built on top of the open-source Zowe Virtual Desktop.

One intuitive interface seamlessly displays all installed software, or data management products, by logical grouping. Each data management product has its own features and interacts with its own relevant z/OS data, but the features and data of all enabled software are integrated into single unified user interface called the *Unified Experience*. For example, IMS software might allow you to register IMS subsystems and work with IMS objects. But if you also install and enable Db2 for z/OS software, you can register Db2 and IMS subsystems in the same Unified Experience by using many of the similar features. Any of the Db2 for z/OS experiences include all the features and functions provided in the Db2 Administration Foundation product. There is seamless integration between the Db2 Administration Foundation and any of the expert functions available in the Db2 experiences.

The benefits of the Unified Experience include:

- Addresses the learning curve for new mainframe users.
- Removes individual product barriers, allowing the logical flow and sharing of information in one common display.

New and changed functions

The enhancements and changes in IBM Unified Management Server for z/OS are made available through a program temporary fix (PTF). The updates to UMS and the data management product are released through separate PTFs. To enable the new features, you must install all the required PTFs.

Unified Management Server

The UMS PTF contains the enhancements, changes, and fixes in Unified Management Server. For information about what is included in the latest PTF, see "What's new in Unified Management Server" on page 5.

Db2 DevOps Experience

The DevOps Experience PTF contains the enhancements, changes, and fixes in the DevOps Experience. To enable these features on the Unified Experience, you must separately install the DevOps Experience PTF with the compatible UMS PTF. For information about what is included in the latest DevOps Experience PTF, see "What's new in Db2 DevOps Experience" on page 9.

Db2 Administration Foundation

The Db2 Administration Foundation PTF contains the enhancements, changes, and fixes in the Db2 Administration Foundation. To enable these features on the Unified Experience, you must separately install the Db2 Administration Foundation PTF with the compatible UMS PTF. For information about what is included in the latest Db2 Administration Foundation PTF, see "What's new in Db2 Administration Foundation" on page 11.

IMS Administration Foundation

PTFs for the IMS Administration Foundation component of IBM IMS Tools Base 1.7 contain enhancements, changes, and fixes for the IMS Administration Foundation features for UMS. For information about what is included in the latest IMS Administration Foundation PTF, see "What's new in IMS Administration Foundation" on page 16.

What's new in Unified Management Server

The Unified Management Server has been updated to include new features, enhancements, and fixes.

- New and changed functions of Unified Management Server 1.2 UMS1.2.0.5 (UI94998)
- New and changed functions of Unified Management Server 1.2 UMS1.2.0.4 (UI93732)
- New and changed functions of Unified Management Server 1.2 UMS1.2.0.3 (UI93091)
- New and changed functions of Unified Management Server 1.2 UMS1.2.0.2 (UI92457)
- New and changed functions of Unified Management Server 1.2 UMS1.2.0.1 (UI91302)
- New and changed functions of Unified Management Server 1.2 UMS1.2.0.0

Table 1. Unified Management Server 1.2 UMS1.2.0.5 (UI94998) new and changed functions

Feature	Description	APAR
ZWEYAML enhancements	Added the following members in ZWEYAML: • `components.izp.dataset.jcllib` • `components.izp.dataset.parmlib` • `components.izp.server.tlsVersionList`	PH57118
Job enhancements	The IZPSYNCY sample job is added to assist with synchronizing example IZP yaml and existing ZWEYAML parmlib member. For more information, see "Installing a program temporary fix (PTF)" on page 74.	
TLS enhancements	The Unified Management Server 1.2 now supports TLS 1.3.	
IMS Administration Foundation: ODBM Dashboard	A new dashboard for each Open Database Manager (ODBM) server and its resources has been added. The IMS Connect dashboard has been enhanced to link it with the new ODBM dashboard.	
IMS Administration Foundation: Support delimiter selection in the export function	You can select a delimiter from a list when exporting an IMS command processor output or a set of statistics data for a DBD. The dialog to be displayed for the delimiter selection is the same as that for the SQL processor.	

Table 2. Unified Management Server 1.2 UMS1.2.0.4 (UI93732) new and changed functions

Feature	Description	APAR
IMS Administration Foundation: Supports the IMS registration of IBM IMS Administration Tool for z/OS	If IBM IMS Administration Tool for z/OS (Program Number: 5655-CAT) is installed and configured with Distributed Access Infrastructure (DAI) and IMS Tools Knowledge Base (IMS Tools KB) of IBM IMS Tools Base for z/OS (Program Number: 5655-V93). IMS registrations for IMS Administration Tool and its associated RECON IDs can be simplified by using IMS Administration Tool data sharing group registration process provided by IMS Administration Foundation. For details on the configuration requirements, see Installing IMS Administration Foundation. For details on configuration requirements for IMS Administration Tool, see IMS Administration Foundation and IMS Tools.	PH56865
Customize UMS started task name	You can now customize the UMS started task name. For more information, see components.izp.zowe.job.suffix member in `ZWEYAML`.	
Subsystem registration	Subsystem registration is improved when the subsystem's `SYSNAME` and `JES2` names are different.	
Export SQL query results	You can now export the results of SQL queries to a CSV file.	
ZWEYAML enhancements	Added the `components.izp.zowe.job.suffix` member in ZWEYAML. For more information see, `components.izp.zowe.job.suffix`.	
Logging enhancement	Improved logging for z/OSMF related errors.	
	Improved logging for the following JCLs: • IZPCPYML • IZPYAML • IZPGENER • IZPCPYM2 • IZPSTEPL • IZPALOPL	
	Error checking is improved for migration scripts.	

Table 3. Unified Management Server 1.2 UMS1.2.0.3 new and changed functions

Feature	Description	APAR
Retry ZSS at configurable interval	UMS attempts to retry Zowe System Services (ZSS) at a configurable interval when it fails to run the first time it is called at start up. This only applies to the initial discovery at start up.	PH55598
Reduce memory usage	Support to prevent application operations from consuming high memory during UMS tasks.	

Table 3. Unified Management Server 1.2 UMS1.2.0.3 new and changed functions (continued)

Feature	Description	APAR
IMS Administration Foundation: Supporting authentication type of `MFA_JWT`	IMS Administration Foundation supports the authentication type (`components.izp.server.authType`) of `MFA_JWT`. For configuration requirements, see Installing IMS Administration Foundation, IMS Administration Foundation and IMS Tools, and Configuring multifactor authentication for UMS. **Note:** If you are using the IMS command processor feature of IMS Administration Foundation with the `authType` of `MFA_JWT`, then certain IMS maintenance is required. For details, see "Software requirements for IMS Administration Foundation" on page 39.	PH55944

Table 4. Unified Management Server 1.2 UMS1.2.0.2 new and changed functions

Feature	Description	APAR
SAF-based management of users and teams	An alternative security mode that stores the team, user, and role information in IBM® System Authentication Facility (SAF) eliminates the need to maintain the user or team list data set. For details, see "Setting up users and teams" on page 46.	PH54452
Support for high availability	Unified Management Server offers high availability for Db2 Administration Foundation features through Zowe with an active/passive mode. For details, see "Preparation for high availability" on page 48.	
IMS Administration Foundation: Removing the DBDLIB and PSBLIB requirements for visualizing DBDs and PSBs when IMS catalog is used	For users of IMS Administration Foundation, the DBDLIB and PSBLIB requirement for DBD and PSB Map feature that enables visualization of IMS database segment structures and segment relationships is removed for IMS data sharing groups that use IMS catalog. **Note:** • To use this feature, the PTF `UI90630` must be applied to IMS Administration Foundation. For details of this PTF, see "What's new in IMS Administration Foundation" on page 16. • This support needs IBM IMS Library Integrity Utilities for z/OS v2.2 (Program Number 5655-U08) and PTF `UI92487` provided by APAR PH54565 in addition to the installation and maintenance of IMS Administration Foundation. • For additional configuration requirements, see Installing IMS Administration Foundation and IMS Administration Foundation and IMS Tools.	
ZWEYAML enhancements	Added the following members in ZWEYAML: • `components.izp.security.useSAFOnly` • `components.izp.dataset.loadLibrary.izp` • `components.izp.toolsDiscovery.enabled` • `components.izp.toolsDiscovery.discoverySearchPaths`	

Table 5. Unified Management Server 1.2 UMS1.2.0.1 new and changed functions

Feature	Description	APAR
IMS Administration Foundation: Support for DBRC-defined groups	Added support for DBRC-defined groups in the "explore objects" for IMS and the "data sharing group" page for IMS administration dashboard. Group views for all types of DBRC-defined groups are added. CAGRP exceptions can be displayed in the new CAGRP view. For details of the prerequisite IBM IMS Tools products to report change accumulation exceptions, see “IMS Administration Foundation and IMS Tools” on page 249.	PH52916
Logging enhancement	Improved error and informational messaging.	
ZWEYAML enhancements	Added the `components.izp.server.host` member in ZWEYAML. For more information see, `components.izp.server.host`.	

Table 6. Unified Management Server 1.2 UMS1.2.0.0 new and changed functions

Feature	Description
Zowe enhancements	The minimum required Zowe version for Unified Management Server 1.2 is version `2.3.0`. The UMS installation process is updated to integrate with the new procedures introduced by Zowe `2.3.0`.
Db2 Administration Foundation	Db2 Administration Foundation 1.2.0.0 is released with UMS 1.2.0.0. For more information, see “What's new in Db2 Administration Foundation” on page 11.
Db2 DevOps Experience	Db2 DevOps Experience 1.3.0.0 is released with UMS 1.2.0.0. For more information, see “What's new in Db2 DevOps Experience” on page 9.

What's new in Db2 DevOps Experience

The Db2 DevOps Experience has been updated to include new features, enhancements, and fixes.

To enable these features, you must install the latest DevOps Experience PTF and the compatible UMS PTF. To choose the appropriate compatible PTF, refer to the UMS PTF compatibility table.

- New and changed functions of Db2 DevOps Experience 1.3.0.5 (UI94997)
- New and changed functions of Db2 DevOps Experience 1.3.0.4 (UI93733)
- New and changed functions of Db2 DevOps Experience 1.3.0.3 (UI93092)
- New and changed functions of Db2 DevOps Experience 1.3.0.2 (UI92459)
- New and changed functions of Db2 DevOps Experience 1.3.0.1 (UI91303)
- New and changed functions of Db2 DevOps Experience 1.3.0.0

Table 7. Db2 DevOps Experience 1.3.0.5 (UI94997) new and changed functions

Feature	Description	APAR
UI support for object creation templates	Introduced a new **DDL template** option in the user interface. You can now use this option in the UI to manage templates and their links to different object types. Refer to "Managing Db2 object templates" on page 183.	PH58157
External Db2 security	You can now use an external Db2 security not only for subsystem registration but also for instance lifecycle workflow by moving the **Suppress Db2 grants** slider toward **True** during subsystem registration.	
Performance enhancement	Improved the performance of the application and the instance workflow.	

Table 8. Db2 DevOps Experience 1.3.0.4 (UI93733) new and changed functions

Feature	Description	APAR
API support for object creation templates	You can now use API to perform CRUD operations for object creation templates. You can also link the object creation templates to other required object creation templates.	PH56863
External Db2 security	You can now use an external Db2 security for subsystem registration by moving the **Suppress Db2 grants** slider toward **True** during subsystem registration. For more information, see "Registering Db2 subsystems" on page 113.	

Table 9. Db2 DevOps Experience 1.3.0.3 (UI93092) new and changed functions

Feature	Description	APAR
Instance add object supports specific name and version	Support for specific name and version in the Db2 object instance when adding or searching for object types such as user-defined functions, stored procedures, and triggers.	PH56032
Performance improvement	Reduced file I/O operations by implementing the cache mechanism for Applications, Teams, Environments, and Subsystems.	

Table 10. Db2 DevOps Experience 1.3.0.2 (UI92459) new and changed functions

Feature	Description	APAR
Creating object templates for different object types	Supports object template creation for all object types in IBM Db2 DevOps Experience. Every object type will be delivered with a template file that contains a CREATE statement, which will be used when a new object is requested or as a required object of another object type. For details, see "Adding Db2 instance objects" on page 177.	PH54940
View pull requests for super admin and team users	Allows the super admin and team users to access and view the pull request and the associated reports. For details, see "Configuring pull request privileges" on page 173.	

Table 11. Db2 DevOps Experience 1.3.0.1 (UI91303) new and changed functions

Feature	Description	APAR
Instance management	If the object is core and the associated application creation team and instance provisioning team are the same, associated application objects can now be modified in the instance.	PH53501
	The add object workflow now supports the addition of non-core objects.	
	You can now click the **Preview changes** button to review an impact report of pull request for application or associated applications .	
	To mark an application as complete, required objects from associated application will now be added as non-core objects.	
Site rule enhancement	Support is added for triggering site rules for associated applications core objects.	
	A site rule applied to the parent application will be enforced on the associated applications. The reverse is not available.	
DDL enhancement	Object DDL changes resulting in the dropping or recreation of objects or associated objects will be highlighted with a warning icon and a tooltip.	

Table 12. Db2 DevOps Experience 1.3.0.0 new and changed functions

Feature	Description
Application management	You can change settings for an application. To perform this action from the applications page, click the overflow menu on the required application and click **Change settings**. For more information, see "Changing application settings" on page 165.

What's new in Db2 Administration Foundation

Db2 Administration Foundation has been updated to include new features, enhancements, and fixes.

To enable these features, you must install the latest Db2 Administration Foundation PTF and the compatible UMS PTF. To choose the appropriate compatible PTF, refer to the UMS PTF compatibility table.

- New and changed functions of Db2 Administration Foundation 1.2.0.4 (UI94996)
- New and changed functions of Db2 Administration Foundation 1.2.0.3 (UI93734)
- New and changed functions of Db2 Administration Foundation 1.2.0.2 (UI93093)
- New and changed functions of Db2 Administration Foundation 1.2.0.1 (UI92458)
- New and changed functions of Db2 Administration Foundation 1.2.0.0

Note: Some of the new and changed features require additional licenses and maintenance. For more information, refer to "Requirements for additional capabilities" on page 38.

Table 13. Db2 Administration Foundation 1.2.0.4 (UI94996) new and changed functions

Feature	Description	APAR
IBM Db2 Query Workload Tuner for z/OS	• Capture SQL from a z/OS data set enables you to identify and display SQL statements that exist in a sequential data set or partitioned data set member and work with those statements by using other Db2 Administration Foundation features. For more information, see "Capturing SQL from a z/OS data set" on page 141. • Workload-level support for Capture Query Environment enables you to capture information about the environments in which all the queries in a workload run. Before this update, the Capture Query Environment supported only single queries. For more information, see "Capturing a query's environment" on page 149.	PH58303
Db2 Analytics Accelerator Loader for z/OS enhancements	• You can now extract queries running on the Db2 Analytics Accelerator Loader as a CSV file. • To support smart table management, the following new columns are introduced on the **Tables** tab: – **Last load time** – **Last statistics collection** – **Is timestamp collected** • A Db2 applications table is added to the **Monitor** tab to enhance monitoring capability. • The following two columns are added to the **Queries** tab to enhance query monitoring: – **Application stall time** – **Application handle**	

Table 14. Db2 Administration Foundation 1.2.0.3 (UI93734) new and changed functions

Feature	Description	APAR
IBM Db2 Query Workload Tuner for z/OS	The following new features are available to licensed Db2 Query Workload Tuner users: • Index Impact Analysis Generates a report that shows the impact of applying the index changes that are recommended by Index Advisor and Workload Index Advisor. For more information, see "Evaluating index recommendations before deploying them" on page 147. • Workload Virtual Index Analyzer Enables you to test indexes to determine if the performance of the SQL statements that comprise a workload can be improved by creating indexes, dropping indexes, or both. For more information, see "Analyzing the effects of applying index changes" on page 148. • Workload Refine Enables you to modify an existing workload. For more information, see "Refining SQL workloads" on page 150. • Workload Explain Failure Reason Explains why SQL statements failed during a Workload Explain operation.	PH56864
Support for DSN commands	Command Execution API support is added to process the requested DSN Command.	
IDAA Collect System Performance Data	You can now request to change the default values given for collecting system performance data.	

Table 15. Db2 Administration Foundation 1.2.0.2 new and changed functions

Feature	Description	APAR
IBM Db2 Query Workload Tuner for z/OS	The following new features are available to licensed Db2 Query Workload Tuner users: • Workload Index Advisor Generates recommendations for creating, altering, or dropping indexes that can improve the performance of an SQL workload. This feature provides the DDL scripts that you can run and information about the existing indexes from the workload that you are tuning. • Workload Statistics Advisor Generates recommendations and `RUNSTATS DDL` for creating or modifying statistical views. • Workload Explain Workload Explain has been enhanced to support `CURRENT QUERY ACCELERATION=ALL`. • Workload Access Path Comparison Workload Access Path Comparison has been updated with the following enhancements: – You can now specify specific types of tuning jobs. – Internal jobs are no longer displayed during a Workload Access Path Comparison action. For more information, see “Tuning SQL workloads” on page 146. • Virtual Index Analyzer Virtually tests indexes to determine if the performance of a single query can be improved by creating indexes, dropping indexes or both. For more information, see “Analyzing the effects of applying index changes” on page 148.	PH56033
Statistics tab for index and tablespace	Implemented infrastructure changes to discover Db2 Administration Tool for z/OS from IBM Db2 Administration Foundation for z/OS. Db2 Administration Foundation is enabled with the Statistics tab for index and tablespace. For details, see “Configuring Statistics tab” on page 76.	PH56033
Running Db2 command using templates	You can view or edit the template of the Db2 command for one or more objects by using the Run Db2 command feature. For details, see “Running Db2 commands using template” on page 134.	
Alert for OUT parameter	SQL call statement now displays an error when only OUT parameter is available.	

Table 16. Db2 Administration Foundation 1.2.0.1 new and changed functions

Feature	Description	APAR
IBM Db2 Query Workload Tuner for z/OS	The following new features are available to licensed Db2 Query Workload Tuner users: • Workload Explain Gathers explain information for all statements in a workload and stores this information in the repository database. This information is used by other workload advisors to analyze these statements. • Workload Query Rewrite Advisor now provides a **Results** page. • The Results pages for **Workload Access Path Advisor** and **Workload Access Path Comparison** now enable you to display both high-level and low-level tuning results.	PH54968
Tools Discovery	Added the capability to find or discover the definition of external Db2 Tools that have been installed on the same z/OS system as the UMS instance. The definition of the tools is stored in YAML files located on UNIX System Services or within an MVS data set. For details, see the following links: • "Configuring UMS for External Tooling" on page 89 • "Configuring UMS for Db2 Analytics Accelerator Loader" on page 88 • "Configuring Storage tab" on page 76	PH54968
Loading tables to accelerators	Implemented infrastructure changes to discover IBM Db2 Analytics Accelerator Loader for z/OS from IBM Db2 Administration Foundation for z/OS. Db2 Administration Foundation user interface changes can load tables to accelerators with Db2 Analytics Accelerator Loader. For details, see "Configuring UMS for Db2 Analytics Accelerator Loader" on page 88.	
Parameters substitution	Support for replacing SQL and call procedure parameters of the registered subsystems. For details, see "Using SQL processor for Db2" on page 125.	
Cancel long-running search operation	Support for canceling an ongoing search request in the 'Explorer objects' when the page takes a long time to load.	
Storage tab for index and tablespace	Implemented infrastructure changes to discover Db2 Administration Tool for z/OS from IBM Db2 Administration Foundation for z/OS. Db2 Administration Foundation is enabled with the Storage tab for index and tablespace. For details, see "Configuring Storage tab" on page 76.	

Table 17. Db2 Administration Foundation 1.2.0.0 new and changed functions

Feature	Description
IBM Db2 Analytics Accelerator for z/OS	You can now perform new actions on the Db2 Analytics Accelerator and its queries: • Add a new Db2 accelerator. To perform this action from subsystem **Overview** tab, click **Add new** and provide new accelerator details. • To save space, you can archive tables on a Db2 accelerator. To perform this action from the **Tables** tab, select tables to archive and click the **Archive** option from the **Archiving** menu. • You can restore archived tables to a Db2 accelerator. To perform this action from the **Tables** tab, select tables to restore and click the **Restore** option from the **Archiving** menu. • Manage federated access for a Db2 accelerator. To perform this action from the **Tables** tab, select required tables and click the **Federation** menu. The following options are available: – Grant access You can grant access to tables in another subsystem. Upon completion of this action, you can create reference tables from those subsystems. – Remove references The remove reference option lets you delete references from an accelerator. • You can create a reference to a table in another subsystem. Upon completion of this action, the reference is created on the shared accelerator. To perform this action from the **Tables** tab, click the **Create references** button. • You can revoke access from a table that will prevent it from referencing a Db2 subsystem. To perform this action from the **Tables** tab, click the **Revoke access** button.

What's new in IMS Administration Foundation

IMS Administration Foundation has been updated to support the Unified Management Server 1.2.

To enable a feature listed in the following table, you must install the PTF that is provided by the APAR that is listed in the table row. The APAR numbers listed in the table are APARs for the IMS Administration Foundation component of IBM IMS Tools Base 1.7, not the Unified Management Server. To choose the appropriate compatible PTF, refer to the UMS PTF compatibility table.

- New and changed functions of IMS Administration Foundation 1.7.0.2 (UI93094)
- New and changed functions of IMS Administration Foundation 1.7.0.1

Table 18. IMS Administration Foundation 1.7.0.2 new and changed functions

Feature	Description	APAR
Support for `MFA_JWT` authentication type	A new configuration parameter for IMS Administration Foundation features is added to support PassTicket-based authentication to the Distributed Access Infrastructure (DAI) TCP Server of IBM IMS Tools Base for z/OS 1.7.	PH55878

Table 19. IMS Administration Foundation 1.7.0.1 new and changed functions

Feature	Description	APAR
Unified Management Server 1.2 support	After the PTF UI90630 provided by the APAR PH49803 has been applied to a single installation of IMS Administration Foundation, it can be used by both Unified Management Server 1.1 and Unified Management Server 1.2.	PH49803

UMS PTF compatibility

The new features and fixes for Unified Management Server and the data management products are released through separate PTFs.

Use this table to choose the PTFs that you must install to enable the features that you need.

Table 20. UMS PTF compatibility

UMS version	Db2 DevOps Experience version	Db2 Administration Foundation version	IMS Administration Foundation version
UMS1.2.0.5 (UI94998)	Db2 DevOps Experience 1.3.0.5 (UI94997)	Db2 Administration Foundation 1.2.0.4 (UI94996) Integrated products: • SQL Tuning Services (UI94641).	No PTF released
UMS1.2.0.4 (UI93732)	Db2 DevOps Experience 1.3.0.4 (UI93733)	Db2 Administration Foundation 1.2.0.3 (UI93734) Integrated products: • SQL Tuning Services (UI93741).	No PTF released
UMS1.2.0.3 (UI93091)	Db2 DevOps Experience 1.3.0.3 (UI93092)	Db2 Administration Foundation 1.2.0.2 (UI93093) Integrated products: • SQL Tuning Services (UI92499).	IMS Administration Foundation 1.7.0.2 (UI93094)
UMS1.2.0.2 (UI92457)	Db2 DevOps Experience 1.3.0.2 (UI92459)	Db2 Administration Foundation 1.2.0.1 (UI92458) Integrated products: • SQL Tuning Services (UI91459).	No PTF released
UMS1.2.0.1 (UI91302)	Db2 DevOps Experience 1.3.0.1 (UI91303)	No PTF released	No PTF released
UMS1.2.0.0	Db2 DevOps Experience 1.3.0.0	Db2 Administration Foundation 1.2.0.0 Integrated products: • SQL Tuning Services (UI81582). • IBM Db2 Analytics Accelerator for z/OS version (7.5.9.0).	IMS Administration Foundation 1.7.0.1

Deprecated and removed functions in Unified Management Server

Certain capabilities of the Unified Management Server 1.1 are deprecated. Their use is discouraged, and support will be removed in the future. Avoid creating new dependencies that rely on these deprecated functions, and also develop plans to remove the dependencies on such functions.

Table 21. Removed functions in Unified Management Server

Removed function	Recommended alternative	Related APAR	Target removal date

Registering, updating, or deleting the Db2 applications can stop with multiple errors.
The Unified Management Server returns error information by using the following JSON response format:
• The **error** field lists the top-level messages
• The **errors** field lists the details of each error.
Each error in the **errors** field has **message** and **localizedMessage** fields. The **message** field contains the message in English and the **localizedMessage** field contains the translated message.
`{`
`  "error": "Message text 1",`
`  "errors": [`
`    {`
`      "message": "Message text 2",`
`      "localizedMessage": "Message text 2"`
`    },`
`    {`
`      "message": "Message text 3",`
`      "localizedMessage": "Message text 3"`
`    }`
`  ]`
`}`
This **localizedMessage** field is deprecated and will be removed. The **message** field will be updated to list the translated message.
The following REST API endpoints are affected:
`POST /ws/policy/applications/dry-run`
`POST /ws/policy/applications`
`PUT /ws/policy/applications`
`PUT /ws/policy/applications//{appId}/objects/reassign`
`DELETE /ws/policy/applications`
`POST /ws/policy/pull-requests/{pullRequestId}/{verb}`
	If you use the application REST APIs to automate development processes, parse the **message** field instead of the **localizedMessage** field.
	If you want to continue using the **localizedMessage** field, specify **`X-API-VERSION=`**`1.1.0.8` in REST calls. For more information, see Chapter 12, “APIs,” on page 265.
	Important: If you specify an old version in the **`X-API-VERSION`** header, you will not be able to use new features of the application APIs in the future.
		PH44497	March 18, 2022

Table 22. Deprecated functions in Unified Management Server

Deprecated function	Recommended alternative	Related APAR	Target removal date
The security configuration requires the security administrator to create and manage the USERLIST and TEAMLIST profiles. The parameters `components.izp.dataset.userList` and `components.izp.dataset.teamList` will be removed to have an alternative security mode that caches stored team, user, and role information from SAF.	The default value is set as `false` for backward compatibility reasons. You should specify `true` for `useSAFOnly` to ignore the following parameters: • `components.izp.dataset.userList` • `components.izp.dataset.teamList` For details, refer to "Setting up users and teams" on page 46.	PH54452	Beginning August 11, 2023, the use of `useSAFOnly` set to `false` is deprecated but will be supported until August 11, 2024.

Service updates and support information

Service updates and support information for this product, including software fix packs, PTFs, frequently asked questions (FAQs), technical notes, troubleshooting information, and downloads, are available from the web.

To find service updates and support information, see the following website:

https://www.ibm.com/support/home/

Product documentation and updates

Information about IBM Unified Management Server for z/OS and any installed data management products, such as IBM Db2 DevOps Experience for z/OS, is available on the web. You can receive updates automatically by registering with the IBM My Notifications service.

Information on the web

The most current version of this information is available on IBM Documentation:

https://www.ibm.com/docs/en

A PDF version of this information is available on the IBM Unified Management Server for z/OS Product Documentation web page; however, IBM Documentation is updated more frequently than PDF books. The IBM Unified Management Server for z/OS Product Documentation web page is located at:

https://www.ibm.com/docs/en/umsfz/1.2.0

Receiving documentation updates automatically

To automatically receive emails that notify you when new technote documents are released, when existing product documentation is updated, and when new product documentation is available, you can register with the IBM My Notifications service. You can customize the service so that you receive information about only those IBM products that you specify.

To register with the My Notifications service:

1. Go to http://www.ibm.com/support/mysupport
2. Enter your IBM ID and password or create one by clicking **register now**.
3. When the My Notifications page is displayed, search for the products that you want to receive information updates about and click **Subscribe**.

   You might want to subscribe for one or more of the following products:

   - IBM Unified Management Server for z/OS
   - IBM Db2 DevOps Experience for z/OS
   - IBM Db2 Administration Foundation for z/OS
   - IBM IMS Administration Foundation for z/OS
4. Specify the types of updates that you want to receive.
5. Click **Submit** to save your profile.

How to send your comments

Your feedback helps IBM to provide quality information. Send any comments that you have about this book to ibmdocs@us.ibm.com. Include the name and version number of the product and the title and number of the book. If you are commenting on specific text, provide the location of the text (for example, a chapter, topic, or section title).

Accessibility features

IBM is committed to accessibility. Accessibility features that follow compliance guidelines are included in the content and documentation to benefit users with disabilities. Parts of the user interface are accessible, but not entirely. Only documentation is compliant, with a subset of parts of the overall product.

The documentation uses the latest W3C Standard, WAI-ARIA 1.0 to ensure compliance with the United States Access Board Section 508 Standards, and the Web Content Accessibility Guidelines (WCAG) 2.0.

The online product documentation is enabled for accessibility. Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully. Documentation is provided in HTML so that it is easily accessible through assistive technology. With the accessibility features, you can do the following tasks:

- Use screen-reader software and digital speech synthesizers to hear what is displayed on the screen. Consult the product documentation of the assistive technology for details on using assistive technologies with HTML-based information.
- Use screen magnifiers to magnify what is displayed on the screen.
- Operate specific or equivalent features by using only the keyboard.

For more information about the commitment that IBM has to accessibility, see IBM Accessibility.

TTY service

In addition to standard IBM help desk and support websites, IBM has established a TTY telephone service for use by deaf or hard of hearing customers to access sales and support services:

800-IBM-3383 (800-426-3383) within North America

Additional interface information

The user interfaces do not have content that flashes 2 - 55 times per second.

The web user interfaces rely on cascading style sheets to render content properly and to provide a usable experience. If you are a low-vision user, you can adjust your operating system display settings, and use settings such as high contrast mode. You can control font size by using the device or web browser settings.

Chapter 2. Security of UMS for z/OS

Unified Management Server 1.2 leverages the strength of IBM® z/OS Resource Access Control Facility (RACF®) and IBM® System Authentication Facility (SAF) to manage the user credential for Unified Management Server 1.2.

Security overview

The security overview covers how secure communication is established between different IBM Unified Management Server for z/OS components.

You can restrict access to Unified Management Server 1.2 by selecting one of the following security models:

- IBM® System Authentication Facility (SAF)-based security
- Data set-based security

Note: Only one security model can be active at a time. It is recommended to configure SAF-based security because data set-based security, although supported, will be deprecated in a future release. For details, see "Deprecated and removed functions in Unified Management Server" on page 18.

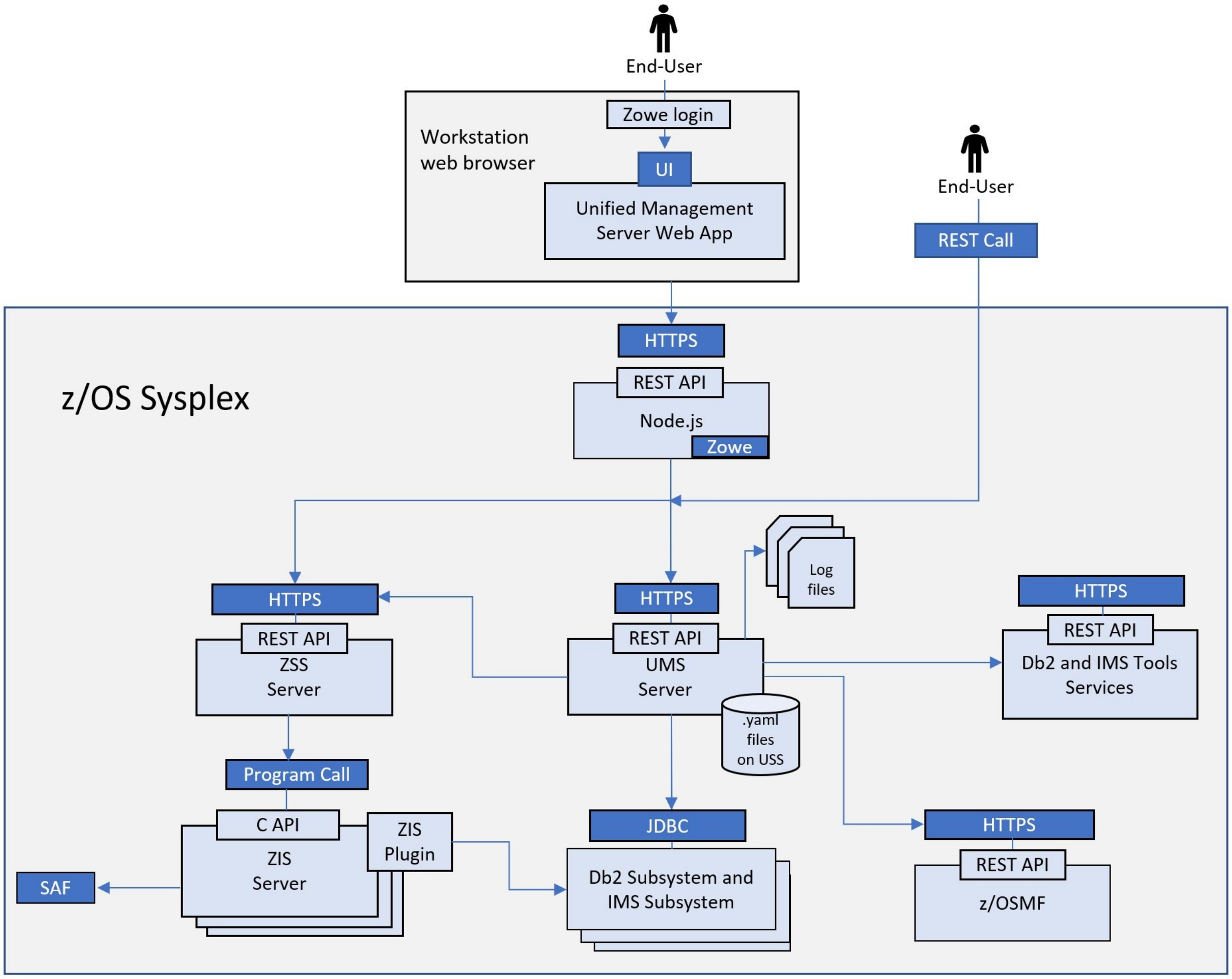
You can enable the SAF-based security by specifying the following in your ZWEYAML member:

```
components:
  izp:
    security:
      useSAFOnly: true
```

For details, see "Setting up users and teams" on page 46.

The following figure illustrates a high-level security architectural overview for Unified Management Server 1.2.

Figure 2. Security architecture for IBM Unified Management Server for z/OS



The following steps list the relationship of the IBM Unified Management Server for z/OS security components.

- The end user uses the Zowe login process for authentication. The Unified Management Server Web App passes the user request over HTTPS to the Zowe server which is a `Node.js` component.
- The `Node.js` component communicates with the UMS server over HTTPS.
- The UMS server securely communicates with the backend services provided by Db2 or IMS subsystem tools, which include the following:

Table 23. Db2 or IMS subsystem tools for Unified Management Server

Product name	Note
IBM SQL Tuning Services	Before tuning a SQL query in UMS, you must install and activate at least one Db2 experience product and configure the UMS server for IBM SQL Tuning Services. UMS uses HTTPS to communicate with IBM SQL Tuning Services.
IBM Db2 Analytics Accelerator for z/OS	Before using the accelerator feature for Db2 queries, you must install and activate at least one Db2 experience product and configure the UMS Java™ Server for Db2 accelerator services. UMS uses HTTPS to communicate with IBM Db2 Analytics Accelerator for z/OS.

Table 23. Db2 or IMS subsystem tools for Unified Management Server (continued)

Product name	**Note**
Distributed Access Infrastructure (DAI) of IBM IMS Tools Base for z/OS	Before using IMS Tools services in IMS Administration Foundation features on UMS, you must install DAI servers and IMS Administration Foundation components of IBM IMS Tools Base for z/OS and configure the UMS server to activate IMS Administration Foundation features with DAI environment. UMS uses IBM z/OS Communications Server Application Transparent Transport Layer Security (AT-TLS) for secure connections to DAI TCP Servers.

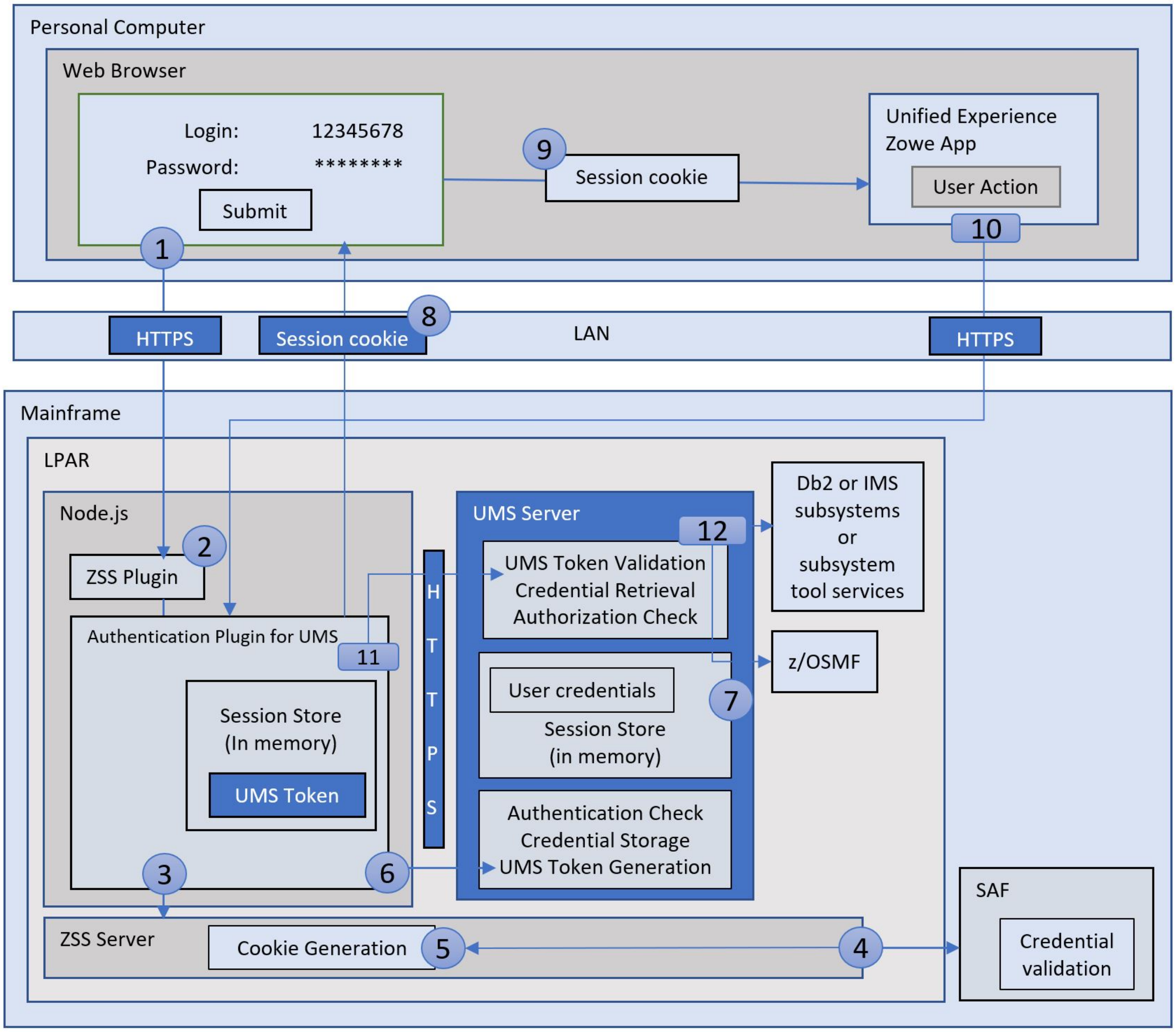
- The network communication between the Zowe and ZSS Server is secured using HTTPS.
- The ZSS Server initiates a program call to the Zowe Cross-Memory (ZIS) Server.
- The ZIS Server communicates with SAF. For example, RACF, ACF2, or Top Secret.
- The ZIS Server auxiliary address space communicates with the Db2 subsystem or the IMS subsystem.
- The UMS server communicates with the Db2 subsystem over JDBC. If IMS Administration Foundation is activated, the UMS server can communicate with IMS Connect servers over JDBC for SQL processing and over an IMS-provided protocol for IMS commands. UMS uses AT-TLS for secure connections to IMS Connect servers.
- The UMS server communicates with the z/OSMF component over HTTPS.
- The end user can also communicate with the UMS server using REST API calls over HTTPS.

Credential management by UMS JWT tokens

Credential management involves securing user credentials through multiple stages in IBM Unified Management Server for z/OS.

The following figure illustrates the credential management process for IBM Unified Management Server for z/OS when the UMS login user authentication type (authType) of STANDARD_JWT is selected.

Figure 3. Credential management in IBM Unified Management Server for z/OS



The following steps list the credential management process for IBM Unified Management Server for z/OS.

1. The user logs into Zowe by providing credentials.
2. The request is passed to the Zowe System Services (ZSS) server plug-in.
3. The ZSS plug-in passes the request to the ZSS server.
4. The ZSS server communicates with System authorization facility (SAF) for credential validation. The SAF could be RACF, ACF2, or Top Secret.
5. Once the user is authenticated by SAF, a session cookie is generated.
6. The Zowe App Server authentication plug-in for IBM Unified Management Server for z/OS sends a login request to the UMS server along with user credentials.
7. The user credentials are verified again in the UMS server through SAF. The user credentials are then stored in the UMS server and a UMS token is generated.
8. The response to the login request is sent back to the browser along with a session cookie, completing the login request process.
9. The session cookie is used for granting access to the Unified Experience Zowe App.
10. Any login user action along with the session cookie is passed to the authentication plug-in using HTTPS.
11. The authentication plug-in uses the session cookie to procure the UMS token from the session store. The user request and the UMS token is then passed to the UMS server for validation.

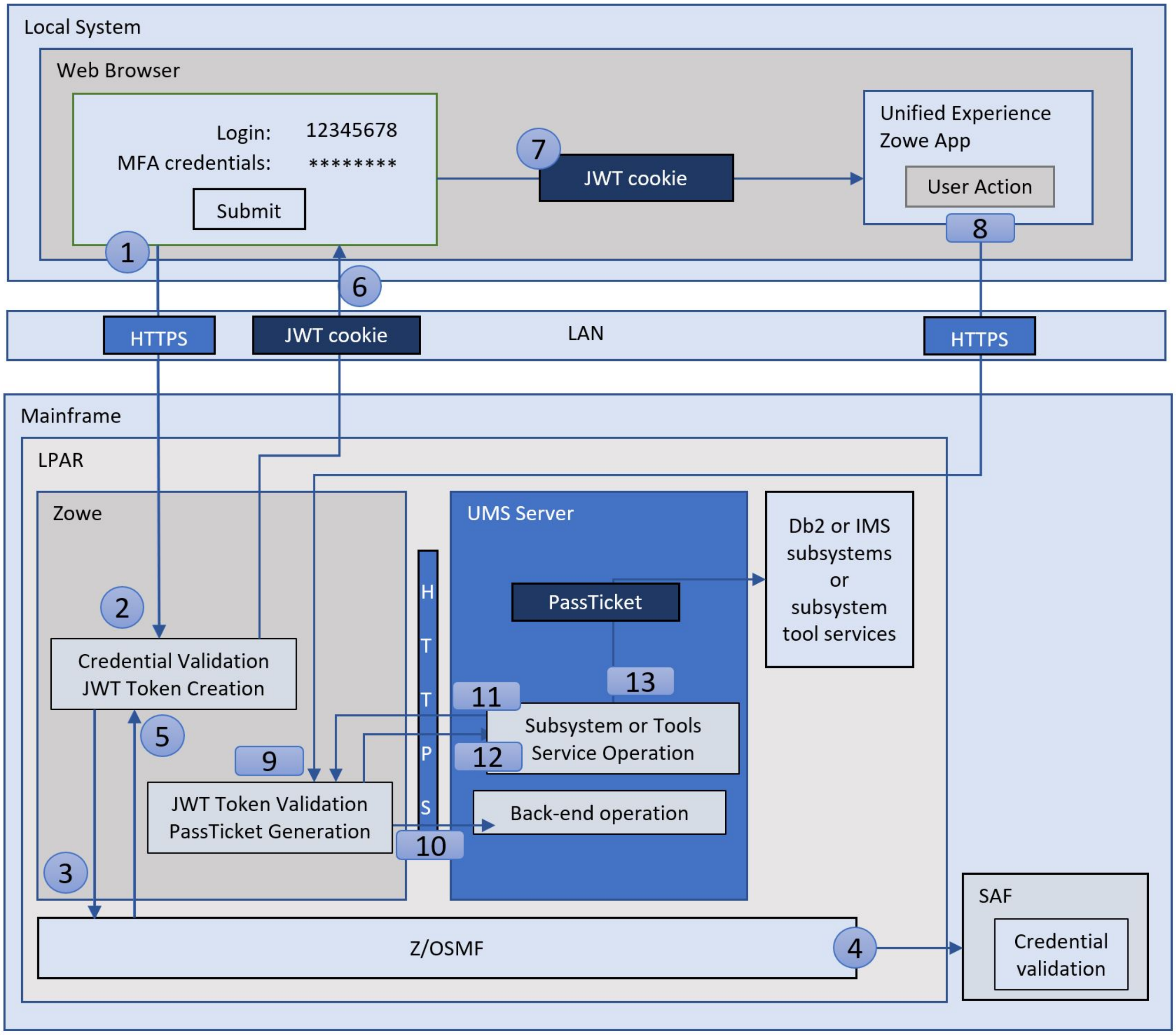
12. The UMS server first validates the UMS token, and then proceeds to carry out the user request, including any request meant for subsystems, subsystem tool services, or z/OSMF.

Credential management by Zowe JWT tokens

You can make the UMS login user authentication more secure with the UMS authentication type (authType) of MFA_JWT. With this authentication type, you can use Multi-Factor Authentication (MFA) with IBM Unified Management Server for z/OS although the use of MFA is optional for this UMS authentication type.

The following figure illustrates the credential management process for IBM Unified Management Server for z/OS when using MFA. For details on configuring MFA for the IBM Unified Management Server for z/OS, see Configuring multifactor authentication for UMS.

Figure 4. Credential management in IBM Unified Management Server for z/OS when using multifactor authentication



The following steps list the credential management process for IBM Unified Management Server for z/OS when using MFA.

1. The user logs into the Zowe desktop by providing MFA credentials.
2. The request is passed to the Zowe server.
3. The Zowe server communicates with z/OSMF for credential validation.

4. The z/OSMF communicates with System authorization facility (SAF) for credential validation. SAF works with IBM MFA to authenticate the user.
5. Once the user is authenticated, the Zowe server creates a JWT cookie for the session.
6. To complete the login process, the response to the login request is sent to the browser along with the JWT cookie.
7. The JWT cookie is used for granting access to the Unified Experience UI.
8. Any user action along with the JWT cookie is passed to the Zowe server.
9. The Zowe server validates the JWT cookie in the incoming request.
10. To perform the user request, the JWT token and the user request is passed to the UMS server for back-end operations.
11. For any requests requiring access to a backend subsystem service, such as Db2 or IMS subsystem or subsystem tool service, the UMS Server presents the JWT token to the Zowe server, and requests for a PassTicket.
12. The Zowe server validates the JWT token, generates a PassTicket, and provides it to the UMS Server.
13. The UMS Server sends the request to backend subsystem service along with the PassTicket to perform the backend subsystem service operation.

Chapter 3. Preparation for installation

Product installation involves completing multiple tasks in a sequence. These tasks are best performed by people in different roles, each with a unique set of skills and authorities.

By planning and preparing for the installation, you can minimize errors and successfully install all the components of the Unified Management Server.

Installation roadmap

The installation of Unified Management Server involves installing multiple components. After installing Zowe and IBM Unified Management Server for z/OS, you can install the data management products.

Before you start the installation, make sure that you've completed the following tasks:

- Plan system capacity as described in "Capacity planning" on page 31.
- Meet the hardware and software requirements of the various components of Unified Management Server 1.2 as described in Prerequisite hardware and software.
- For details about the array parameter `components.izp.experiences` to be specified in the `ZWEYAML` member of `PARMLIB`, see "Step 2: Installing Unified Management Server" on page 56.
- For the notation of `ZWEYAML` parameters, see "YAML format" on page 237.

Table 24. Installation roadmap

Component	Tasks included in the installation	User role	Instructions
Zowe	• Install the Zowe prerequisite software • Install the Zowe z/OS runtime • Configure the Zowe z/OS runtime • Verify the installation	System administrator	• Zowe prerequisite software • Installing and configuring Zowe
IBM Unified Management Server for z/OS	• Run the SMP/E installer • Complete the post-SMP/E installation tasks • Start multiple instances of the ZIS server • Start Zowe	System administrator	"Performing SMP/E installation tasks" on page 52

Table 24. Installation roadmap (continued)

Component	Tasks included in the installation	User role	Instructions
IBM Db2 Administration Foundation for z/OS	• Stop Zowe • Run the SMP/E installer • Add the location of the SMP/E installation for the experience to ZWEYAML member of parmlib in the `components.izp.experiences` array. • Specify the required parameters • Start Zowe **Note:** Some of the new and changed features require additional licenses and maintenance. For more information, refer to "Requirements for additional capabilities" on page 38.	System administrator	"Installing Db2 Administration Foundation" on page 74
IBM IMS Administration Foundation for z/OS	• Install HAFN170 with SMP/E. • Apply all prerequisite PTFs to HAFN170. • Stop Zowe • Add the location of the SMP/E installation for the experience to ZWEYAML member of parmlib in the `components.izp.experiences` array. • Specify the required configuration parameters • Check and configure IMS Connect security • Check and configure other IMS security settings • Check and configure IMS Tools Base server security settings • Start Zowe	System administrator	"Installing IMS Administration Foundation" on page 77

Table 24. Installation roadmap (continued)

Component	Tasks included in the installation	User role	Instructions
IBM Db2 DevOps Experience for z/OS	• Stop Zowe • Add the location of the SMP/E installation for the experience to ZWEYAML member of parmlib in the `components.izp.experiences` array. • Specify the required parameters • Start Zowe	System administrator	"Installing Db2 DevOps Experience " on page 83

To explore the features of the data management products, see the following topics:

- Working with subsystems, teams, environments, and users
- Using Db2 DevOps Experience
- Using Db2 Administration Foundation
- Using IMS Administration Foundation

Capacity planning

The IBM Unified Management Server for z/OS has the following capacity requirements.

Table 25. Basic system capacity

Capacity required for	Hardware	Number of LPAR/ Server	CPU (Per LPAR/ Server)	Memory (GB) (Per LPAR/ Server)	DASD/Disk Space (GB) (Per LPAR/ Server)
Unified Management Server 1.2	IBM zSystems	**1** (Installed on 1 LPAR per SYSPLEX.)	No specific recommendation. Parts of the workload are zIIP eligible.	**18** GB	**2** GB
IBM Db2 DevOps Experience for z/OS	IBM zSystems	No additional capacity required.	No specific recommendation. Parts of the workload are zIIP eligible.	More memory will be needed depending on the number of Db2 subsystems.	**1** GB
IBM Db2 Administration Foundation for z/OS	IBM zSystems	No additional capacity required.	No specific recommendation. Parts of the workload are zIIP eligible.	More memory will be needed depending on the number of Db2 subsystems.	**1** GB

Table 25. Basic system capacity (continued)

Capacity required for	Hardware	Number of LPAR/ Server	CPU (Per LPAR/ Server)	Memory (GB) (Per LPAR/ Server)	DASD/Disk Space (GB) (Per LPAR/ Server)
IBM IMS Administration Foundation for z/OS	IBM zSystems	No additional capacity required.	No specific recommendation. Parts of the workload are zIIP eligible.	More memory will be needed depending on the number of IMS subsystems, the numbers of resources, DBDs, and PSBs that are defined for those IMS subsystems.	**1** GB

UMS roles and responsibilities

UMS installation requires different privileges to perform certain tasks.

UMS installation is done by a system administrator. The security administrator gives access rights and privileges to certain roles. To understand the functioning of UMS and the roles involved, refer to UMS roles and responsibilities.

For the notation of ZWEYAML parameters that are referred to in the table, see “YAML format” on page 237.

Table 26. UMS roles and responsibilities

User ID	Role	Description	Required privilege or permission	User role management
`<system_admin_id>`	System administrator	The system administrator facilitates UMS installation, such as configuring UMS and running non-security post-installation tasks.	The system administrator must be in the same group as the UMS started task user and must be able to perform the following functions: • Run SMP/E to create data sets and USS files for UMS. • Install Zowe plug-ins.	System administrator role management is outside the scope of the product. z/OS security administrator usually assigns this role.

Table 26. UMS roles and responsibilities (continued)

User ID	Role	Description	Required privilege or permission	User role management
`<security_admin_id>`	Security administrator	The security administrator performs SAF administration tasks during installation that require elevated authority on your system. If the security is SAF-based, then security administrator is responsible for managing users (optionally DevOps Experience teams) by providing READ access to SAF profiles in the IZP class: `IZP.SUPER` or `IZP.ADMIN`.	The security administrator must be able to perform the following functions: • Define a new class called IZP. • Define a new profile in IZP. • Refresh the IZP class (`RACF AND ACF2 ONLY`). • Create new users: – with a password (`ACF2 ONLY`). – without a password (`RACF AND TSS ONLY`). • Define a new data set profile. • Grant access to new profiles in IZP. • Associate a user with a new started task. • Define new profiles in the CRYPTOZ (or equivalent) class. • Grant access to the new profiles in CRYPTOZ.	Security administrator role management is outside the scope of the product. z/OS security administrator usually assigns this role.

Table 26. UMS roles and responsibilities (continued)

User ID	Role	Description	Required privilege or permission	User role management
`<super_admin_ID>`	UMS super administrator	The UMS super administrator performs administrative functions in UMS and can perform the following tasks: • Register and assign subsystems. • Create environments. • Establish site rules. • Create and manage teams. • Create and manage applications.	• The UMS super administrator must have the appropriate access authority to a specific SAF profile, such as `IZP.SUPER*` or `IZP.SUPER-`. • The UMS super administrator for Db2 data management products must have access to Db2 and Db2 catalog tables.	Any z/OS user can be assigned this role by running the provided JCL (`IZPUSRMD`). When using SAF-based security, any z/OS user can be assigned this role by providing READ access to `IZP.SUPER` in the IZP class.
`<regular_user_ID>`	UMS regular user	The UMS regular user performs developer functions in UMS.	The UMS regular user is not a UMS super administrator and must have access to `IZP.ADMIN*` or `IZP.ADMIN-`.	Any z/OS user can be assigned this role by running the provided JCL (`IZPUSRMD`). When using SAF-based security, any z/OS user can be assigned this role by providing READ access to `IZP.ADMIN` in the IZP class.

Table 26. UMS roles and responsibilities (continued)

User ID	Role	Description	Required privilege or permission	User role management
`<ums_user_id>`	UMS users	UMS users are a set of all users with the role of a UMS super administrator or a UMS regular user.	UMS users must have a record in the data set specified by `components.izp.dataset.userList` in ZWEYAML.	• If SAF-based security is not used, UMS users must have a record in the data set specified by `components.izp.dataset.userList` in ZWEYAML. • A UMS user can either be a UMS super administrator or a UMS regular user, but not both.
`<team_member_id>`	UMS team member	A UMS team member is a UMS user who is assigned to be a part of a UMS team and performs the following functions: • View registered and discovered subsystems. • View environments. • View users. • View site rules. • Update instance objects. • Issue pull requests. • Merge updates.	UMS team members must have access to `IZP.ADMIN*` or `IZP.ADMIN-`.	This role assignment is done via UMS, and a UMS user can be assigned this role for multiple UMS teams.

Table 26. UMS roles and responsibilities (continued)

User ID	Role	Description	Required privilege or permission	User role management
`<team_admin_id>`	UMS team administrator	The UMS team administrator is assigned to administer a UMS team and performs the following functions: • Perform all UMS team member tasks. • View users. • Update team members. • Manage applications. • Approve pull requests. • Commit changes.	• UMS team administrators must have access to `IZP.ADMIN*` or `IZP.ADMIN-`. • UMS team administrators for Db2 data management products must have access to Db2 and Db2 catalog tables.	This role assignment is done via UMS and a UMS user can be assigned this role for multiple UMS teams.

Prerequisite hardware and software

Unified Management Server 1.2 uses both IBM proprietary and open source technologies and requires the installation of various hardware and software products on the z/OS. Make sure that you procure, install, and configure all the prerequisite products prior to installing IBM Unified Management Server for z/OS.

Hardware requirements for UMS

For IBM Unified Management Server for z/OS and all data management products, the following hardware is required:

- Amount of memory: 18 GB (default) recommended
- Amount of disk space: 2 GB minimum

Software requirements for UMS

Before you install and configure IBM Unified Management Server for z/OS, make sure that your environment meets the following minimum software requirements.

- IBM z/OS 2.4 or later.
  - If you are running UMS on z/OS 3.1, there are additional requirements. For details, see "Additional requirements for z/OS 3.1" on page 38.
- IBM z/OS Integrated Cryptographic Service Facility (ICSF) must be installed and configured with a token data set (TKDS) and the ICSF started task started, as described in the *z/OS Cryptographic Services ICSF: System Programmer's Guide*.
- IBM z/OS Management Facility (z/OSMF) (version is same as z/OS version). For information, see the z/OSMF documentation.
- IBM Resource Access Control Facility (RACF®) or another equivalent external security manager that supports z/OS system authorization facility (SAF) interface.
- Refer to the following table for the supported Zowe version for each UMS release or PTF level.

Table 27. Supported Zowe versions

UMS version	Zowe version
UMS1.2.0.5 (UI94998)	Zowe 2.3.0 and later
UMS1.2.0.4 (UI93732)	Zowe 2.3.0 and later
UMS1.2.0.3 (UI93091)	Zowe 2.3.0 and later
UMS1.2.0.2 (UI92457)	Zowe 2.3.0 and later
UMS1.2.0.1 (UI91302)	Zowe 2.3.0 and later
UMS1.2.0.0	– Zowe 2.3.0 – Zowe 2.4.0 – Zowe 2.5.0 – Zowe 2.7.0 and later **Note:** Zowe 2.6.0 is not supported with UMS1.2.0.0.

Restriction: If you are running Zowe 2.12.0, you must remove the `zowe.sysMessages` section from the `zowe.yaml` file.

If you are using Zowe distributed by IBM Z Distribution for Zowe, see the Fix list for IBM Z Distribution for Zowe for the relationship between the Zowe releases, APARs, and PTFs for IBM Z Distribution for Zowe. For more information, see the IBM Zowe documentation.

IBM Z® Distribution for Zowe is a no-charge offering with optional Subscription and Support (S&S).

- Zowe support for components used by the listed offerings is included as part of the licensing of the below offerings when the corresponding product S&S entitlement is current. No separate Zowe S&S entitlement is required.
  - IBM Db2 DevOps Experience for z/OS & IBM Unified Management Server for z/OS
  - IBM Db2 Administration Foundation for z/OS & IBM Unified Management Server for z/OS
  - IBM IMS Administration Foundation for z/OS (a component of IBM IMS Tools Base for z/OS) & IBM Unified Management Server for z/OS
- Zowe can be supported for a fee by IBM Software Support in the below use cases:
  - Zowe components not used by any of the offerings listed above.
  - Zowe issues unrelated to the offerings listed above.
- The following Zowe components are mandatory:
  - Zowe Application Framework (ZLUX)
  - Zowe Cross-Memory (ZIS) Server
  - Zowe Cross-Memory Auxiliary Server
- The following Zowe component is mandatory for multifactor authentication and high availability support:
  - Zowe API Mediation Layer
- The Zowe CLI component is not supported.

For more information on how Zowe configures, validates, and starts all of its components, see Zowe component lifecycle.

Important: To bring Unified Management Server up and running, you must complete all the Zowe steps that are related to Zowe cross-memory server setup. This includes configuring the Zowe auxiliary address space. For details, see the following topics:

- Installing and configuring the Zowe cross memory server

  - Zowe Auxiliary Address space
- IBM 64-bit SDK for z/OS, Java Technology Edition 8. For information, see IBM Java SDK.
- IBM SDK for Node.js - z/OS, Node.js v14.x (except v14.17.2), v16.x, or v18.2 (except v18.12.1). For more information, see Zowe documentation.
- Any of the browsers supported by Zowe. For details, see the Zowe documentation.

Additional requirements for z/OS 3.1

If you are running UMS on z/OS 3.1, the following are required:

- UMS 1.2 must be 1.2.0.4 (APAR PH56865, PTF UI93732) or later.
- A Zowe release that supports z/OS 3.1 must be used:
  - If you are using Open Source Zowe, V2.11 or later is required.
  - If you are using IBM Z Distribution for Zowe V2, the following PTFs are required:
    - PTF UO90031 (APAR IO29286)
    - PTF UO90032 (APAR IO29287)
- Fixpack 15 (PTF `UI94465`/ APAR `PH58170`/ PH58174) is required for IBM 64-bit SDK for z/OS, Java Technology Edition, version 8 SR8.

For details on Zowe releases, see Fix list for IBM Z Distribution for Zowe. For Java version, maintenance level, and fixpack download, see Java SDK Products on z/OS.

Software requirements for Db2 Administration Foundation

Before you install and configure IBM Db2 Administration Foundation for z/OS for use with the Unified Management Server, make sure that your environment meets the following minimum software requirements.

Important: Unified Management Server 1.2.0 requires Db2 Administration Foundation 1.2.0. Db2 Administration Foundation 1.1.0 is not supported by Unified Management Server 1.2.

- The Zowe version 2.3.0 or later is installed.
- All of these versions of Db2 for z/OS are supported as of the General Availability (GA) date of Unified Management Server 1.2. If a version of Db2 for z/OS is withdrawn from service, it will no longer be supported by Unified Management Server:
  - Db2 for z/OS, 12.1 (5650-DB2)
  - Db2 for z/OS Value Unit Edition, 12.1 (5770-AF3)
  - Db2 for z/OS, 13.1 (5698-DB2)
  - Db2 for z/OS Value Unit Edition, 13.1 (5698-DBV)
  - Db2 supplied administrative enablement stored procedures ADMIN_COMMAND_DSN, ADMIN_COMMAND_DB2, ADMIN_INFO_SYSPARM, and DSNUTILV must be operational.

    Note: IBM Db2 Administration Foundation for z/OS does not require the Db2 Administrative Scheduler.

Requirements for additional capabilities

Several features of Db2 Administration Foundation require additional software. The following list describes the requirements to use these features.

- To tune SQL statements from the Db2 Administration Foundation user interface, you must install and configure one of the following SQL Tuning Services products:
  - IBM Database Services Expansion Pack is available at no additional cost to licensed Db2 Accessories Suite for z/OS users and includes a subset of all the available SQL Tuning Services features.

  - IBM Db2 Query Workload Tuner for z/OS 6.1 can be purchased separately or as part of the Db2 Performance Solution Pack for z/OS, which contains all the features that are included in IBM Database Services Expansion Pack along with a more robust set of SQL analysis and tuning features.

    For more information about the features included in these two offerings, see Overview of IBM SQL Tuning Services.

- To manage accelerators and accelerated queries from the Db2 Administration Foundation user interface, install and configure IBM Db2 Analytics Accelerator 7.5.9.0.
- To load table data to one or more accelerators in a Db2 for z/OS subsystem, install Db2 Analytics Accelerator Loader version 2.1 and apply APAR PH54984. Also, ensure that APARs PH54968 and PH54452 are applied.
- To enable the **Storage** tab for indexes and table spaces, ensure that Db2 Administration Tool version 13 is installed and that APARs PH54968 and PH55177 are applied.

Note: These additional software components must be installed on the same system that IBM Unified Management Server is installed. Otherwise, UMS will shut down when you attempt to use these components.

Software requirements for Db2 DevOps Experience

Before you install and configure IBM Db2 DevOps Experience for z/OS for use with the Unified Management Server, make sure that your environment meets the following minimum software requirements.

- All of these versions of Db2 for z/OS are supported as of the General Availability (GA) date of Unified Management Server 1.2. If a version of Db2 for z/OS is withdrawn from service, it will no longer be supported by Unified Management Server:
  - Db2 for z/OS, 12.1 (5650-DB2)
  - Db2 for z/OS Value Unit Edition, 12.1 (5770-AF3)
  - Db2 for z/OS, 13.1 (5698-DB2)
  - Db2 for z/OS Value Unit Edition, 13.1 (5698-DBV)
  - Db2 supplied administrative enablement stored procedures ADMIN_COMMAND_DSN, ADMIN_COMMAND_DB2, ADMIN_INFO_SYSPARM, and DSNUTILV must be operational.

    Note: IBM Db2 Administration Foundation for z/OS does not require the Db2 Administrative Scheduler.
  - Git for z/OS 2.14.4 or later. For information, see Rocket Open Source Languages and Tools.

Software requirements for IMS Administration Foundation

Before you install and configure IBM IMS Administration Foundation for z/OS for use with the Unified Management Server, make sure that your environment meets the following minimum software requirements.

To use IMS Administration Foundation, one or more of the IMS 15 releases are required. IMS Administration Foundation supports the following IMS releases and configurations:

- IMS supported releases:
  - IBM Information Management System 15.2.0 (PID: 5635-A06)
  - IBM Information Management System Database Value Unit Edition 15.2.0 (PID: 5655-DS5)
  - IBM Information Management System Transaction Manager Value Unit Edition 15.2.0 (PID: 5655-TM4)
  - IBM Information Management System 15.3.0 (PID: 5635-A06)
  - IBM Information Management System Database Value Unit Edition 15.3.0 (PID: 5655-DS5)
  - IBM Information Management System Transaction Manager Value Unit Edition 15.3.0 (PID: 5655-TM4)

  - IBM Information Management System 15.4.0 (PID: 5635-A06)
  - IBM Information Management System Database Value Unit Edition 15.4.0 (PID: 5655-DS5)
  - IBM Information Management System Transaction Manager Value Unit Edition 15.4.0 (PID: 5655-TM4)
- IMS system supported configuration:
  - DB/DC
  - DBCTL
  - DCCTL
- Dynamic Resource Definition (DRD) for IMS resource management, using the resource definition data set (RDDS), the IMSRSC repository, or both
- An IMS subsystem to be used for IMS Administration Foundation must be configured as a member of an IMSplex.
- The IMSplex to be used for IMS Administration Foundation must be configured with the following features activated:
  - IMS Common Service Layer (CSL)

    CSL must be activated for all IMSplex systems to be used with IBM Unified Management Server for z/OS.
  - IMS Operations Manager (OM)

    OM must be configured for all IMSplex systems to be used with IBM Unified Management Server for z/OS.

If you want to use the IMS command processor feature for an IMSplex or pages for IMSplex, IMS, and IMS Connect, at least one IMS Connect server instance that is connected to the IMSplex must be configured and the IMS Connect OM Command exit routines HWSCSLO1 and HWSSMPL1 must be specified in the EXIT= parameter of the TCPIP statement for the IMS Connect server. For details, see the sections "IMS Connect support for IMSplex", "IMS Connect OM Command exit routines (HWSCSLO0 and HWSCSLO1)", and "Ping support for IMS Connect" in the *IMS Communications and Connections Guide*.

If you want to use the Unified Management Server authentication type of `MFA_JWT` to manage user credentials using Zowe JWT tokens, the IMS Connect maintenance provided by the IMS APAR PH51844 is required. For details, see RACF PassTicket support for commands from IMS Connect to IMS OM in the *IMS 15.4 Release Planning Guide*.

If you want to use the SQL processor feature for an IMS data sharing group, the following IMS components must be activated for the data sharing group and the IMSplex to which the data sharing group belongs:

- IMS catalog
- An IMS Connect server instance connected to the IMSplex
- An Open Database Manager (ODBM) server instance

For IMS catalog, both `ACBMGMT=ACBLIB` and `ACBMGMT=CATALOG` are supported. A data sharing group that can be registered to the Unified Management Server must share a single IMS catalog. If the data sharing group is composed of IMS subsystems of mixed `ACBMGMT` types, the IMS catalog must be set up to support ACB management by running the `DFS3PU00` utility with the `MANAGEDACBS=SETUP` control statement specified. For this data sharing group configuration, it is user's responsibility to synchronize the active ACB timestamps and content of the IMS catalog and the ACB timestamps and content of ACBLIBs for the IMS subsystems of `ACBMGMT=ACBLIB`.
For IMS Connect and ODBM, at least one DRDA port needs to be configured and at least one ODBM datastore alias for an IMS data sharing group for which the IMS SQL processor is to be used needs to be defined.

For other conditional requirements, see “IMS Administration Foundation and IMS Tools” on page 249.

Function level support for the data management products

When you activate new Db2 for z/OS function levels in a Db2 subsystem or data sharing group, enhancements might become available that impact IBM Unified Experience for z/OS.

The levels of function level support are defined as follows:

Tolerated
The product works as it did on a previous release or function level of Db2 for z/OS, but it does not support the new features of this function level.

Supported
The product supports applicable features of the new function-level.

Function level support for Db2 Administration Foundation

The following function levels are tolerated or supported by IBM Db2 Administration Foundation for z/OS and are provided with the corresponding PTF, if any.

Table 28. IBM Db2 Administration Foundation for z/OS PTFs in support of Db2 13 for z/OS function levels

Db2 13 for z/OS function level	Toleration PTF	Support PTF
FL504	No PTF required	No PTF required
FL503	No PTF required	No PTF required
FL502	No PTF required	No PTF required
FL501	No PTF required	No PTF required
FL500	No PTF required	No PTF required

Table 29. IBM Db2 Administration Foundation for z/OS PTFs in support of Db2 12 for z/OS function levels

Db2 12 for z/OS function level	Toleration PTF	Support PTF
FL510	No PTF required	No PTF required
FL509	No PTF required	No PTF required
FL508	No PTF required	No PTF required
FL507	No PTF required	No PTF required
FL506	No PTF required	No PTF required
FL505	No PTF required	No PTF required
FL504	No PTF required	No PTF required
FL503	No PTF required	No PTF required
FL502	No PTF required	No PTF required
FL501	No PTF required	No PTF required
FL500	No PTF required	No PTF required

Function level support for Db2 DevOps Experience

The following function levels are tolerated or supported by IBM Db2 DevOps Experience for z/OS and are provided with the corresponding PTF, if any.

Table 30. IBM Db2 DevOps Experience for z/OS PTFs in support of Db2 13 for z/OS function levels

Db2 13 for z/OS function level	Toleration PTF	Support PTF
FL504	No PTF required	No PTF required
FL503	No PTF required	No PTF required
FL502	No PTF required	No PTF required
FL501	No PTF required	No PTF required
FL500	No PTF required	No PTF required

Table 31. IBM Db2 DevOps Experience for z/OS PTFs in support of Db2 12 for z/OS function levels

Db2 12 for z/OS function level	Toleration PTF	Support PTF
FL510	UI75959	No PTF required
FL509	UI74629	UI75959
FL508	UI73530	No PTF required
FL507	No PTF required	UI73530
FL506	No PTF required	No PTF required
FL505	No PTF required	No PTF required
FL504	No PTF required	No PTF required
FL503	No PTF required	No PTF required
FL502	No PTF required	No PTF required
FL501	No PTF required	No PTF required
FL500	No PTF required	No PTF required

Required networks and ports

The Unified Management Server 1.2 requires dedicated ports for communicating across component systems and services.

Note: For the notation of ZWEYAML parameters that are referred to in the table, see "YAML format" on page 237.

This table lists the required USS TCP/IP port numbers and their default values.

Table 32. Required USS TCP/IP port numbers

Component	Use	Port
IBM Unified Management Server for z/OS	Internal communication	The following ports are used and must be specified by the configuration parameter mentioned for each port. The parameters starting with components.izp are for the Unified Management Server and are specified in the UMS PARMLIB member `ZWEYAML`. Other parameters are for other Zowe components and are specified in the `zowe.yaml` file for Zowe. The default value for each parameter is also mentioned. • Port for UMS to listen for API requests: (`components.izp.server.port.http : 12023`). The parameter is specified in the PARMLIB member `ZWEYAML`. • Port for UMS to listen for internal requests: (`components.izp.server.port.agent: 3444`). The parameter is specified in the PARMLIB member `ZWEYAML`. • Port for Gremlin Graph to listen: (`components.izp.server.port.gremlin: 8182`). The parameter is specified in the PARMLIB member `ZWEYAML`. • Port for ZSS server in the `zowe.yaml` file for the Zowe instance to be used for the installation of the Unified Management Server: (`components.zss.port: 8542`). **Note:** The parameter `components.zss.port` is specified in the `zowe.yaml` file for the Zowe instance. • Port for z/OSMF: (`zOSMF.port: 443`). The parameter `zOSMF.port` is specified in the `zowe.yaml` file for the Zowe instance.
IBM Db2 Administration Foundation for z/OS	Internal communication	Same as IBM Unified Management Server for z/OS.
IBM IMS Administration Foundation for z/OS	Internal communication	Same as IBM Unified Management Server for z/OS.
IBM Db2 DevOps Experience for z/OS	Internal communication	Same as IBM Unified Management Server for z/OS.

Chapter 4. Installation

This section provides information and instructions for installing the Unified Management Server *(UMS)* and other software products that are supported by UMS.

A system administrator will install UMS, and a security administrator will create security profiles and grant access to them. The installation script has several job control language (JCL) files. Instructions in this section will guide you to edit and submit the JCL jobs.

Before you begin

This section helps you understand the resources, persona, and external security managers involved in the Unified Management Server (UMS) installation, and lists the pre-installation tasks that you must perform before starting the installation task.

To install the Unified Management Server and IBM Unified Experience for z/OS, the following persona must have privileged access to the system:

- A subsystem administrator that has SYSADMIN privileges for each Db2 subsystem to be used for UMS if you are installing a Db2 product and that has IMS system administrator privileges or IMS system programmer privileges for each IMS subsystem to be used for UMS if you are installing an IMS product.

  Note: The administrator who has these special privileges for subsystem management is called a DBA user throughout this document.
- A security administrator with privileges to create SAF profiles and grant access to them.
- A user account with read access to the Zowe installation directory and data sets. The user account also needs access to create and write to new data sets for UMS.

Note: The terms UMS and IZP are interchangeably used throughout this document.

Important: If you need to rollback to UMS 1.1, customer support will help you with this process. There is no automatic backup of work done in UMS 1.2. You must retain backups of the following before initiating the rollback process:

- PROCLIB, PARMLIB, JCLLIB members from UMS 1.1.
- USS folder referred to by `IZP_UMS_VARDIR`.

You must also:

- Understand UMS roles and responsibilities.
- Perform the pre-installation steps.

z/OS resources that are involved with installation

Various z/OS resources are involved with IBM Unified Experience for z/OS.

The following figure illustrates relationships among major z/OS resources that must be configured during installation of Unified Management Server. The Zowe application is labeled IBM Unified Experience for z/OS.

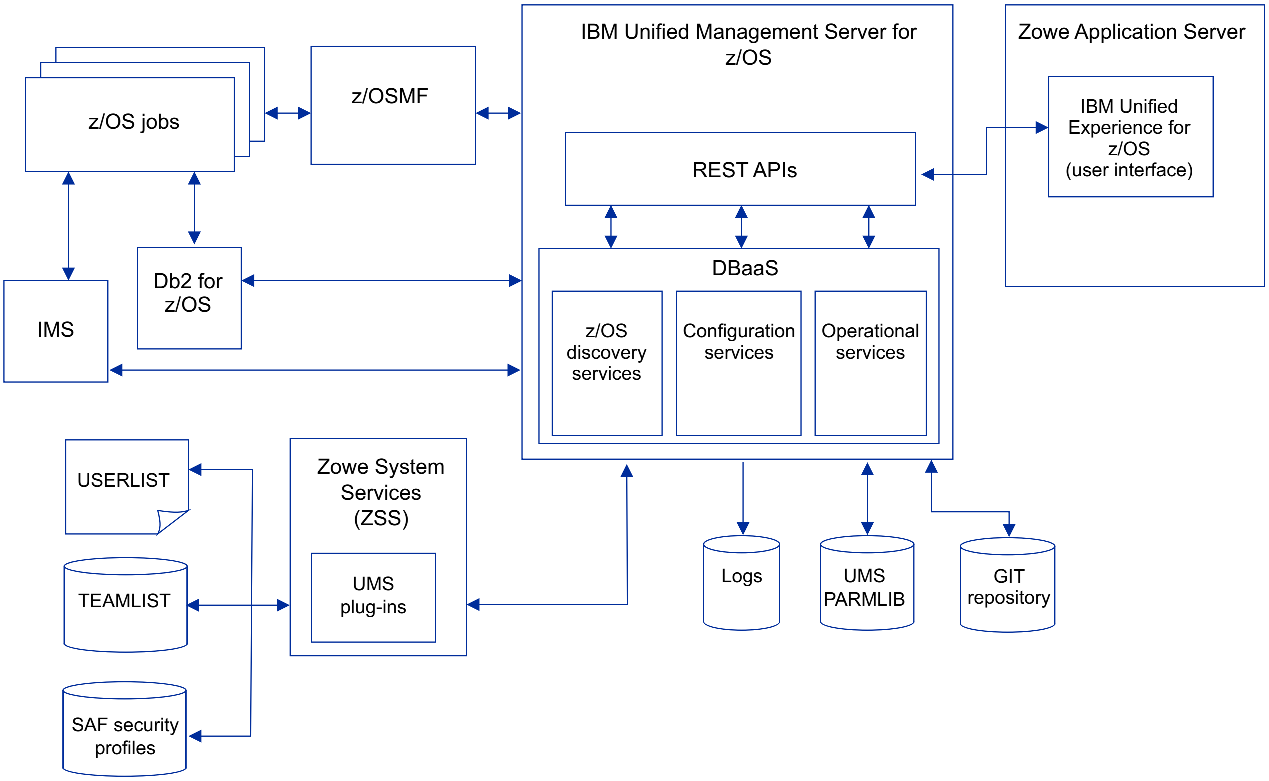


Figure 5. Relationships among UMS and different z/OS resources

Note:

- DBaaS, which is a collection of micro-services that enable administration, definition, and operation features, will be available only after one or more data management products are installed and activated.
- The USERLIST and TEAMLIST data sets are optional. These data sets are required only when `components.izp.security.useSafOnly` is set to `false`.
- The GIT repository is optional. It is required only when Db2 DevOps Experience is used.

Setting up users and teams

You can restrict access to Unified Management Server 1.2 by selecting one of the following security models: SAF-based and data set-based security.

Note: Only one security model can be active at a time. It is recommended to configure SAF-based security because data set-based security, although supported, will be deprecated in a future release. For details, see "Deprecated and removed functions in Unified Management Server" on page 18.

Important: For more information on SAF-based management of users and teams, see Security enhancement for IBM Unified Management Server.

You can enable SAF-based security by specifying the following in your ZWEYAML member:

```
components:
  izp:
    security:
      useSAFOnly: true
```

Notes:

- Ensure the assigned user has permission to refresh user metadata or team metadata in Unified Management Server.

- Ensure that the security administrator has provided super administrator with READ access to `IZP.FUNCTION.USERS.GET` or `IZP.FUNCTION.TEAMS.GET`.
- It is recommended to using the User and Team management methods, although the USERLIST and TEAMLIST data sets are the default options for backward compatibility.

To setup a user or team profile, the super administrator should perform the following:

1. Configure and migrate the user profile or team profile.
2. Define new profiles and assign users to them. For details, refer to "Defining a security class for UMS" on page 259.

   Note: These assigned users will have access to refresh the users and teams.

   Table 33. IZP profiles and their privileges

Class	Profile name	Access required	Function
IZP	IZP.FUNCTION. USERS.GET	READ	Refresh user information.
IZP	IZP.FUNCTION. TEAMS.GET	READ	Refresh team membership information.
IZP	IZP.FUNCTION. ROLES.GET	READ	Determine the role of a user.

   If the UMS server and UI start successfully after the UMS installation and configuration, the refresh button 🔄 will be visible for the user.

   Important: If you are not a super administrator, you can only view your own storage usage and limits when `useSafOnly` is enabled.

Configuring user profile

Unified Management Server will query data from SAF through RACROUTE to acquire a list of users with access to a role profile. These users will represent all sets of UMS users. For details, refer to "UMS roles and responsibilities" on page 32 and "Defining a security class for UMS" on page 259.

Table 34. Users with access to role profiles

Class	Profile	Role
IZP	IZP.SUPER*	Super User
IZP	IZP.ADMIN*	Admin User

- Regardless of whether `useSafOnly` is set to true or false, access to UMS depends on having READ access to a role profile in the IZP class. If you have already provided a set of user access to the profiles, then migration is not required. You can always add or remove profiles by permitting or revoking their access.
- When the `useSafOnly` is set to false, only the users with access to a role profile are included in the USERLIST, which is required for login.

Therefore, the list of UMS users can differ depending on whether `useSafOnly` is true or false.

Configuring team profile

The security administrator creates and manages team profiles. Each Unified Management Server team will have a corresponding profile. This could be a generic profile or one that specifies the eight-character SAF qualifier (or saf_id). Using this profile and ID, you can assign membership to a team.

For details on the UMS teams, see the following topics: “UMS roles and responsibilities” on page 32, “Key concepts” on page 105, and “Managing teams” on page 120.

- If `useSafOnly` is set to false, the teams are stored as a member in the TEAMLIST.
- If `useSafOnly` is set to true, a team is instead stored as a profile in the IZP class `IZP.TEAM.{saf_id}`. The *{saf_id}* is a unique qualifier assigned to a team, which is used in the SAF profile to determine team membership.

After configuration the team profiles, you need to migrate them using the following steps:

1. Create a team profile in the IZP class in the format:

   ```
   IZP.TEAM.{saf_id}
   ```

   Where, *{saf_id}* is the name of the data set member of the team you want to migrate. For example,

   ```
   RDEFINE IZP IZP.TEAM.{saf_id} UACC(NONE)
   ```

2. Grant the required access to the created profiles.

   Where, READ is a team member and UPDATE is a team administrator. You should compare to the JSON data stored in the TEAMLIST member that corresponds to the team you want to migrate to verify the ID and team role. For example,

   ```
   PERMIT IZP.TEAM.{saf_id}  CLASS(IZP) ID(<ID>) ACCESS(READ)
   ```

   ```
   PERMIT IZP.TEAM.{saf_id}  CLASS(IZP) ID(<ID>) ACCESS(UPDATE)
   ```

   ```
   PERMIT IZP.TEAM.{saf_id}  CLASS(IZP) ID(<ID>) DELETE
   ```

Assigning membership

To assign membership, you must provide user access to a team profile. The UPDATE access or higher indicates a team administrator and READ access indicates a team member.

For example, to add a user as a team administrator to team 'A' (with a generated *{saf_id}*) issue the following command:

```
'PERMIT IZP.TEAM.A CLASS(IZP) ID(USERID) ACCESS(UPDATE)'
```

Note: For the membership to be reflected in UMS, a user with access to the required function profile (`IZP.FUNCTION.TEAMS.GET`) should refresh the team membership. To refresh, use the ⟳ icon displayed on the upper-right corner of the **Users** page.

External security managers

An external security manager provides a layer of security to your operating system.

UMS allows you to work with the following ESMs:

- RACF
- Top Secret
- ACF2

Preparation for high availability

UMS offers high availability for Db2 Administration Foundation features through Zowe with an active/passive mode, whereas most other Zowe processes use active/active.

For UMS to support high availability:

- The API Mediation Layer (gateway component) must be enabled.
- The external host and port specified for the API Mediation Layer in zowe.yaml must point to the DVIPA address, which is set up according to Configuring Sysplex for high availability from the Zowe documentation.

To run UMS under Zowe high availability mode, configure DVIPA with the UMS main HTTP port defined as "hot standby", listing the UMS IP addresses of the machines that are running Zowe.

The high availability instance name specified in the Zowe `haInstances` parameter will be included in the UMS log file name located in the UNIX System Services `zowe.logDirectory`.

Note: If there is any unexpected application behavior, exit the UMS screen within the Zowe application manager and reopen the UMS application after a few seconds for the backup process to finish initializing.

A sample DVIPA configuration is shown below.

```
VIPADYNAMIC
  VIPADEFINE       <DVIPA mask> <DVIPA IP address>
    VIPADISTRIBUTE
       DISTMETHOD HOTSTANDBY <DVIPA IP address> PORT <UMS port>
       DESTIP
          <IP-address-1> PREFERRED
          <IP-address-2> BACKUP
ENDVIPADYNAMIC
```

In addition to the UMS port, the ZSS port should also be configured with DVIPA, and the DVIPA address should be selected as the first element in the Zowe externalDomains host array.

To use high availability, MFA_JWT authentication is required. However, the users are not required to be MFA users. The MFA_JWT authentication type invokes the API Mediation Layer Single Sign-On (SSO) mechanism that is leveraged by UMS high availability. For details on configuring the `authType parameter` to enable Zowe API Mediation Layer, refer to "Configuring multifactor authentication for UMS" on page 235.

Note:

- If UMS is switching to the backup server when the active server is no longer available, the user will experience unexpected behaviors, including access denied error messages and changing menu items. You need to exit the UMS application within Zowe and reopen it to fix the issue.
- Db2 DevOps Experience and IMS Administration Foundation do not support the UMS high availability configuration.

Setting up secure communication for UMS

By default, Unified Management Server uses the keystore and truststore that are specified in the `zowe.yaml` configuration file for Zowe. You can use a different keystore or truststore by specifying its location and type in the ZWEYAML member to be used to configure UMS.

Before you begin

Confirm in which USS directory the Zowe configuration YAML file (`zowe.yaml`) for the Zowe instance to be used for your UMS server installation is located. For details of the Zowe configuration for keystore and truststore, see the section "YAML configurations - certificate" and Zowe certificate configuration.

About this task

The setup procedure described in this section is required only when you want to use a keystore or truststore that is different from the one used by Zowe components. The parameters that can be specified in the UMS PARMLIB member ZWEYAML are as follows:

- `components.izp.security.certificate.keystore.location`
- `components.izp.security.certificate.keystore.type`
- `components.izp.security.certificate.keystore.alias`

- `components.izp.security.certificate.truststore.location`
- `components.izp.security.certificate.truststore.type`

Zowe supports Public Key Cryptography Standards #12 (PKCS12) and SAF keyrings of JCERACFKS keystore types for both keystores and truststores. Unified Management Server supports the Java KeyStores (JKS) type for both keystores and truststores in addition to PKCS12 and JCERACFKS-type keyrings.

Recommendation: For any environment other than proof-of-concept testing environments, it is highly recommended to use keyrings for both keystores and truststores.

Important: If you are sharing the keystore between Zowe and Unified Management Server, check whether the requirements on Extended Key Usage (EKU) and Subject Alternate Name (SAN) described in the Zowe certificate requirements section of the Zowe documentation are satisfied for the certificate, which will be commonly used by all Zowe components, including UMS. If any of these requirements are not satisfied, connection errors can occur during Zowe server startup.

For TLS connection requests sent to a UMS server, the UMS server acts on the "server" side and the "client" is either one of the following:

- The Zowe App Server
- The Zowe API Gateway
- A REST API client that does not use the API Gateway

For TLS connection requests sent from a UMS server, the UMS server acts as a "client" and the "server" in this case is one of servers running on z/OS. Those z/OS servers include the following:

- A Zowe ZSS server
- A z/OSMF server
- A Db2 subsystem
- A SQL Tuning Services server for Db2
- An Administration Services server for Db2 Analytics Accelerator
- An IMS Connect server

Keystores and truststores are used in TLS communications. They are repositories that contain cryptographic artifacts like certificates and private keys that are used for cryptographic protocols. The procedure below describes how to set up these repositories to enable TLS communications for UMS.

Procedure

1. Specify key ring or file-based keystore parameters.

   Using a key ring as a keystore

   The most secure and recommended method of storing certificates is using a key ring. If you are using your own keystore, you must specify that in the following ZWEYAML items, and manage it as per your security procedures:

   a) `components.izp.security.certificate.keystore`

   b) `components.izp.security.certificate.keystore.type`

   c) `components.izp.security.certificate.keystore.alias`

   d) `components.izp.security.certificate.keystore.location`

   - Know the PARMLIB member that contains the Unified Management Server configuration parameters. This was first edited when you installed Unified Management Server. The default is `{components.izp.dataset.parmlib}(ZWEYAML)`. You must specify the keyring you set up as `components.izp.security.pkcs11.certificate.keystore.location` in the PARMLIB.
   - If you self-generated your server certificate and you want to enable client authentication, your server certificate must contain the TLS Web Client Authentication (`1.3.6.1.5.5.7.3.2`) value in the extended key usage section.

To add the server certificate and the certificate authority used to sign it to your key ring, run the following commands.

Note: The default UMS started task ID is the same as the Zowe started task ID, which is referred to as `<zowe_started_task_id>` in the example below.

```
RACDCERT CONNECT(ID(<zowe started task id>) LABEL(‘<SERVER_CERTIFICATE_LABEL>’)
RING(<RINGNAME>) USAGE(PERSONAL) DEFAULT) ID(<zowe started task id>)
RACDCERT CONNECT(CERTAUTH LABEL(‘<CA_LABEL>’) RING(<RINGNAME>) USAGE(CERTAUTH))  ID(<zowe
started task id>)
SETROPTS RACLIST(DIGTCERT,DIGTRING) REFRESH
```

Using a file-based keystore

To use a file-based keystore, you must import your certificates into the Zowe file-based key store, which is used by default. Zowe provides a certificate import function. For details, see Zowe documentation.

2. Specify key ring or file-based truststore parameters.

Using a key ring as a truststore

In order to facilitate secure communication between UMS and z/OSMF, you must add the certificate authority (CA) of z/OSMF to the UMS truststore. To enable access to UMS during runtime, the key ring must be accessible by the `<ZOWE_STARTED_TASK_ID>`. You can use the same key ring for your truststore and keystore. If you have not created a key ring, run the following commands:

```
RACDCERT ADDRING(<RINGNAME>) ID(<ZOWE_STARTED_TASK_ID>)
SETROPTS RACLIST(DIGTRING) REFRESH
```

To add certificate authority of z/OSMF to your key ring, run the following commands:

```
RACDCERT CONNECT(CERTAUTH LABEL(‘<ZOSMF_CA_LABEL>’) RING(<RINGNAME>) USAGE(CERTAUTH))
ID(<ZOWE_STARTED_TASK_ID>)
SETROPTS RACLIST(DIGTCERT,DIGTRING) REFRESH
```

Using a file-based truststore

By default, when UMS launches, it will use the truststore specified in the `zowe.yaml` by key `zowe.certificate.truststore.file`, which already contains a certificate for z/OSMF.

UMS will not import any certificates into the Zowe location, so you must import Db2 certificates or a certificate authority into the Zowe truststore. If you are using SQL Tuning Services or Db2 Analytics Accelerator Administration Services, you must also import certificates for the services into that same truststore.

If you are using IMS command processor or IMS SQL processor in IMS Administration Foundation, you must include certificates for root CAs for the server certificates for all IMS Connect servers to be used. If, in addition, you are using any IMS Tools feature in IMS Administration Foundation, you must include certificate for the root CA for the server certificate for the IMS Tools Base Distributed Access Interface (DAI) TCP Server to be used for IMS Administration Foundation. Zowe provides a certificate import function. For details, see Zowe documentation.

If you specify your own location for truststore in `components.izp.security.certificate.truststore.location`, UMS will use that location when launching, but you must separately import certificates for services, including z/OSMF.

If you specify the location as `components.izp.workspaceDirectory/conf/cacerts`, then UMS will automatically import the certificates for z/OSMF, SQL Tuning Services, and Db2 Analytics Accelerator Administration Services during the Zowe configure step. You must use `root` or a user with group access to import the Db2 certificate into the `conf/cacerts` file. The `conf/cacerts` file requires ownership by the Zowe started task user. The password to `conf/cacerts` is "`password`". This is same for root CA certificates to be used for TLS connections to IMS Connect servers and the IMS Tools Base DAI TCP server.

You can use the Java program keytool for certificate import:

```
$JAVA_HOME/bin/keytool -noprompt -keystore truststoreLocation -importcert -alias
anyAlias -file certificateFile
```

What to do next

Applicable to Db2 experiences: Users are required to create connection profiles for the SQL Tuning Service. If you are using SSL encryption for the Db2 connectivity, UMS will pass the name of the UMS truststore to the SQL Tuning Service when the profile is created.

UMS will automatically create Db2 connection profiles for the Db2 Analytics Accelerator Administration Services. Again, if you are using SSL encryption for Db2 connectivity, UMS will pass the name of the UMS truststore when the profile is created.

- Using a file-based keystore

  If you are using a file-based keystore with UMS, ensure that the SQL Tuning Service started task user ID and the Db2 Analytics Accelerator Administration Services started task user ID have read permissions on the UMS truststore file.

- Using a key ring as a truststore

  If you are using keyrings with UMS, SQL Tuning Service started task user ID and the Db2 Analytics Accelerator Administration Services started task user ID also need access to read the UMS keyring. You can select one of the following options:

  - Define and permit `UPDATE` on `IRR.DIGTCERT.LISTRING` in class `FACILITY` to the SQL Tuning Service and the Db2 Analytics Accelerator Administration Services started task user IDs. This action permits those user IDs the authority to read any keyring on the system.
  - Define and permit `CONTROL` on `<UMS keyring owner>.<UMS keyring name>.LST` in class `RDATALIB` to the UMS, SQL Tuning Service, and the Db2 Analytics Accelerator Administration Services started task user IDs.
  - Create the connection profiles outside UMS and pass the name of a truststore already used for the SQL Tuning Service and the Db2 Analytics Accelerator Administration Services started tasks. Make sure to connect the appropriate Db2 Root CAs to the keyring specified in `components.izp.security.certificate.truststore.location`, or if blank, the keyring specified in `zowe.certificate.truststore.file` so the services can connect securely to Db2.

Performing SMP/E installation tasks

This section lists the pre-installation tasks required to install the UMS.

Before you begin with the installation, you must complete the following tasks:

1. Choose whether you will install the SMP/E components on a separate read-only file system.
2. Run SMP/E:
   a. Refer to the *IBM Unified Management Server for z/OS Program Directory* and validate if hardware and software requirements are met.
   b. Validate if prerequisite software is installed.
   c. Run the SMP/E installer using instructions provided in the *IBM Unified Management Server for z/OS Program Directory*.
3. Note down the USS extraction folder used for SMP/E installation. The folder path name will need to be specified in `ZWEYAML` parameter `components.izp.runtimeDirectory` when you perform post-SMP/E installation. For details, see "Item 2: Identify the UMS z/OS UNIX System Services directories" on page 55.
4. UMS Zowe plug-ins require Program Control authorization. In order to tag the files with this bit, the SMP/E install user requires `BPX.FILEATTR.PROGCTL` permission on the system. If the install user does not have this permission, SMP/E will install the UNIX System Services files without the bit set, after attempting to extract paxes with the bit set. Even when the install user has permissions and the

program control bit is properly set, if the SMP/E results are copied to another file system, the program control bit might not be retained.

- To remedy this problem, a user with write access and BPX.FILEATTR.PROGCTL permission must run the following UNIX System Services commands:

  a. Set your working directory to the UMS runtime directory as installed by SMP/E using the cd command. This is the value you specified as components.izp.runtimeDirectory in PARMLIB(ZWEYAML).
  b. Run the following command to change the attributes on all plug-in shared objects:

  ```
  extattr +p */zssServer/lib/*
  ```

Note: If you are using spool management or archive system, ensure that the jobs submitted by this product are not immediately removed from the spool. Jobs should remain in the spool for the duration of the operation, which may be several minutes.

If you are installing Db2 DevOps Experience, you also need to install the FMIDs (H0IHD10 and H25GD10). These are included with the restricted license for Db2 DevOps Experience, or you can reuse an existing installation of Db2 Administration Tool and Db2 Object Comparison Tool. The HLQs are required for the libraries of FMIDs (H0IHD10 and H25GD10) during installation, see "Step 2: Installing Unified Management Server" on page 56.

Post-SMP/E installation of UMS

Before you proceed with the installation, you must first understand the installation workflow.

Installation workflow

After you have run the SMP/E installer and performed all the pre-installation tasks, you must proceed to the UMS installation.

The UMS install script provides JCL jobs for each step. Each JCL has a set of instructions that guide you to perform the operations you must perform for the script to complete successfully.

Note: For some systems, listing the set of instructions might be different as the external security manager (ESM) and the installations vary.

Workflow description	Required/Optional
Collect required parameters • Prepare a DBA user ID and ensure its privileges • Identify the UMS z/OS UNIX System Services folder • Collect installation details of prerequisite software • Validate Integrated Cryptographic Service Facility (ICSF) • Determine locations for UMS data sets • Identify users who will need access to UMS	Required
Stop Zowe.	Required
Stop all ZSS cross memory servers and their auxiliary address spaces.	Required
Copy SIZPSAMP to another location.	Optional
Edit and submit IZPALOPL for allocating PARMLIB.	Required
Edit and submit IZPCPYML for copying SIZPPARM (IZPYAML) to PARMLIB (ZWEYAML).	Required

Workflow description	Required/Optional
(Optional) Edit and submit IZPMIGRA for copying previous experience PARMLIB members to new PARMLIB location.	Optional Edit and submit this JCL if migrating from UMS 1.1
Edit ZWEYAML.	Required
Edit and submit IZPGENER for validating ZWEYAML and creating JCLLIB data sets.	Required
Run the IZPA3 JCL. The IZPA1 and IZPA2 JCLs are required only if useSAFOnly=false.	Required
Edit and submit IZPCPYM2 for copying ZWEYAML to USS IZP.yaml.	Required for Zowe version 2.5.0 and below
Run ESM JCLs generated by the IZPGENER JCL.	Required
Encrypt DBA credentials.	Required
Edit and submit Zowe component install (IZPIPLUG) to associate UMS and plugins with Zowe.	Required
Submit IZPEXPIN to install experiences	Required
Edit the Zowe started task to recognize the IZP yaml specification.	Required
Start all ZSS cross memory servers and their auxiliary address spaces on all LPARs.	Required
Start Zowe to start UMS as a component of Zowe.	Required
Validate the UMS installation.	Required

Step 1: Collecting required parameters

This task is performed by a system administrator.

About this task

This is the first step in the UMS installation process. The UMS installation script uses the PARMLIB(ZWEYAML) member that is created by IZPCPYML. For details of the IZPCPYML JCL and the PARMLIB member ZWEYAML, see “Step 2: Installing Unified Management Server” on page 56. For the notation of ZWEYAML parameters that are referred to in the table, see “YAML format” on page 237.

Item 1: Prepare a DBA user ID and ensure its privileges

The UMS will need a DBA user ID and its password to run the backend job. You can set the password to never expire by running the following command:

```
PASSWORD NOINTERVAL USER(IZP_DBA)
```

The DBA user ID will be specified in PARMLIB(ZWEYAML) as components.izp.security.pkcs11.dbaUser.

The DBA user ID must have the following privileges:

- z/OSMF user privileges for its z/OS jobs REST services and z/OS data set and file REST services.
- SYSADM privileges for each Db2 subsystem used for UMS.

- IMS system administrator privileges or IMS system programmer privileges for each IMS subsystem used for UMS.
- ALTER access to the temporary data sets that are defined by `tempDatasetHLQ` parameter. For details about this parameter, see the description for the PARMLIB member IZPDB2PM in "Installing Db2 DevOps Experience " on page 83.
- If IMS Administration Foundation is activated, ALTER access to the temporary data sets that are defined by `imsTempDatasetHLQ` parameter. For details about this parameter, see the description for the PARMLIB member IZPIMSPM in "Installing IMS Administration Foundation" on page 77.

In addition, the `DBA user ID`:

- Must have an OMVS segment.
- Must be assigned a UID and a GID in OMVS.

For details of Db2 privileges, see Db2 security setup topics in Db2 for z/OS documentation.

Important: If multifactor authentication was used for the UMS server, the DBA user ID must not be a regular UMS user or be defined as an MFA user.

Item 2: Identify the UMS z/OS UNIX System Services directories

You must identify the z/OS UNIX System Services directory where UMS is installed.

Note that this is the SMP/E directory, which is already installed. This directory should be on the same LPAR as Zowe. This directory will be referred to as `components.izp.runtimeDirectory` in the rest of this document.

1. Adjust access rights to the UMS installation directory and Zowe installation directory.
2. The UMS started task ID is same as the Zowe started task ID. The ID must have read/write access to the directory that will be specified by the parameter `components.izp.workspaceDirectory` in the member ZWEYAML of the UMS PARMLIB. This directory and its sub directories will be used as the UMS workspace where UMS files to be installed, added, or updated will be placed. The UMS started task ID is same as Zowe started task ID by default. The UMS started task ID will be referred to as <ZOWE_STARTED_TASK_ID> or `zowe_started_task_id` in this document.

Item 3: Confirm the location of Zowe configuration YAML file

Confirm in which USS directory the Zowe configuration YAML file (`zowe.yaml`) for the Zowe instance to be used for the UMS server installation is located. The `zowe.yaml` file location will be specified in the IZPGENER JCL to be used in the UMS server installation. For details, see "Step 2: Installing Unified Management Server" on page 56. For details of the Zowe configuration file, see Zowe documentation.

Item 4: Validate Integrated Cryptographic Service Facility (ICSF)

The credentials of the `DBA user ID` will be encrypted by using a secure key that is to be generated and store in the ICSF TKDS.

Make sure that the Integrated Cryptographic Service Facility (ICSF) is installed with TKDS feature enabled, and ensure that the TSO ID that is to be used to install UMS and the <Zowe_STARTED_TASK_ID> have privileges to use ICSF token services. For details of the SAF CSFSERV resource access privileges that are required to use ICSF token services, see the topic on the SAF controls used by the PKCS #11 Token Browser in *z/OS Cryptographic Services Integrated Cryptographic Service Facility Administrator's Guide*. Especially, see the section titled "Setting up profiles in the CSFSERV general resource class". For example, if any of the necessary permissions is missing for the UMS installer user, you may see the error message IZPSC0003E when you run the shell script that is used to encrypt the DBA user credentials.

Item 5: Determine locations for UMS data sets

You must determine the locations for UMS data sets.

Select a high-level qualifier (HLQ) for UMS data sets.	UMS uses data sets to store configuration information. The high-level qualifiers must be identified at this time. It will be used during security configuration and will be recorded as a configuration parameter. The value you select will be referred to as *config_hlq* in the rest of this document. This qualifier will be used to create multiple new data sets. The started task ID and the installer user must have access to these data sets. For example: IZP
PARMLIB	UMS needs a partitioned dataset to store its parameter files. There will be a main UMS PARMLIB member called ZWEYAML and additional PARMLIB members for each installed experience.
JCLLIB	UMS creates a number of JCL jobs from a template that is installed by SMP/E based on the values in your PARMLIB. These will be stored in a partitioned dataset.

Item 6: Identify users who will need access to UMS

You need to collect the initial set of users. These users must exist and have access to Zowe. You are not required to create new or dedicated user accounts. You must use the same accounts that are used to access Zowe. If SAF-based security is not used, UMS users must have a record in the data set specified by `components.izp.dataset.userList` in ZWEYAML.

Step 2: Installing Unified Management Server

A new installation process is introduced for Unified Management Server 1.2.

Before you begin

- If you are migrating from UMS 1.1, the minimum required PTF level is UMS1.1.0.13 (UI81668).
- UMS 1.2 runs under the Zowe started task user. The default Zowe started task jobname is ZWESLSTC.
- If you are not familiar with the YAML file structuring, see "YAML format" on page 237.
- The HLQ.SADBLINK library should be APF authorized. For details, see IBM documentation.

Procedure

1. Stop Zowe.

   Refer to Zowe documentation for detailed steps on stopping Zowe.

   ```
   /p <job_name>
   ```

   If Zowe defaults have been chosen, job_name should be replaced with ZWESLSTC.
2. Stop ZSS cross-memory server. The ZSS cross-memory server runs on each z/OS LPAR. All cross-memory servers on all LPAR that are used by the ZSS server will need to be stopped. Refer to the Zowe documentation for details on stopping a Zowe cross-memory server. Its associated cross-memory auxiliary server address space will be stopped automatically by the Zowe cross-memory server. You do not need to manually stop the address space.

   ```
   /p <job_name>
   ```

   If Zowe defaults have been chosen, job_name should be replaced with ZWESISTC.

3. Copy SIZPSAMP to another data set of the same attributes. For example, you could use the data set name of *config_hlq*.SAMPLIB or *config_hlq*.INSTLIB, where *config_hlq* is the HLQ that was selected in Item 5.

   The following JCLs are available in SIZPSAMP. It is recommended to copy these JCLs to a new read/write data set before editing for your requirements:

JCL name	Description
IZPALOPL	Allocates a new PARMLIB dataset if the provided dataset does not exist.
IZPCPYML	Creates the ZWEYAML JCL under PARMLIB dataset.
IZPCPYM2	Copies ZWEYAML to `izp.yaml` under USS.
IZPGENER	Generates other JCLs required for configuring UMS.
IZPMIGRA	Migrates current values from version 1.1 to version 1.2.

4. Edit and submit IZPALOPL from your copy of SIZPSAMP.

   The IZPALOPL file allocates a new PARMLIB dataset. Replace the following values and submit the IZPALOPL JCL:

Value	Description
`{v1.2_parmlib}`	Replace it with the location of your v1.2 PARMLIB.
`{components.izp.dataset.runtimeHlq}`	Replace it with the HLQ of your SMP/E installed data sets. **Note:** The `components.izp` structure in the ZWEYAML member lists items related to UMS.

5. Edit and submit IZPCPYML from your copy of SIZPSAMP.

   The IZPCPYML creates the ZWEYAML JCL under PARMLIB. Replace the following values and submit the IZPCPYML JCL:

Value	Description
`{components.izp.dataset.parmlib}`	Replace it with the PARMLIB location.
`{components.izp.dataset.runtimeHlq}`	Replace it with the required HLQ.

6. (Optional) Edit and submit IZPMIGRA from your copy of SIZPSAMP.

   The IZPMIGRA is required if you are migrating from UMS 1.1. The minimum required PTF level is UMS1.1.0.13 (UI81668). The first installation of UMS 1.2 does not require this JCL. The IZPMIGRA JCL performs the following functions:

   - Converts some of the UMS v1.1 configuration variables in the UMS v1.1 PARMLIB member IZPUMSPM to the UMS v1.2 configuration variables in the UMS v1.2 PARMLIB member ZWEYAML.
   - Copies the UMS v1.1 PARMLIB members for UMS experience products to the UMS v1.2 PARMLIB.

   Replace the following values and submit the IZPMIGRA JCL:

Value	Description
`{v1.1_parmlib}`	Replace it with the location of your v1.1 PARMLIB.
`{v1.2_parmlib}`	Replace it with the location of your v1.2 PARMLIB.
`{components.izp.dataset.runtimeHlq}`	Replace it with the HLQ of your SMP/E installed data sets of UMS 1.2.

7. Edit the following values in ZWEYAML from your PARMLIB.

Value	Description
`components.izp.enabled`	This must always be true.
`components.izp.debugShellScripts`	Whether or not to set -x to enable shell debugging.

`components.izp.experiences`	If you are installing any of the UMS experience products and planning to activate one or more of them in the configuration, the runtime directories of those experience products must be specified as an array value for this key. For example:
	`components:`
	`  izp:`
	`    enabled: true`
	`    debugShellScripts: false`
	`    experiences:`
	`    - /usr/lpp/IBM/doe/v1r3m0/bin`
	`    - /usr/lpp/IBM/afx/v1r2m0/bin`
	`    - /usr/lpp/IBM/afn/v1r7m0/bin`
	Important: If you are activating IMS Administration Foundation, you must apply a PTF to your IMS Administration Foundation installation before configuring UMS or changing the UMS configuration to specify the array value for this key. For details, see "Installing IMS Administration Foundation" on page 77.

`components.izp.jobCard`	The job card that will be used for generated jobs when submitting `IZPGENER`. The job card is specified as a YAML array. For example:
	`jobCard:`
	`  - //IZPCUST1 JOB`
	`  - //*`
	`  - //*`
	To erase existing values, place empty brackets after the element. For example: `jobCard: []`.

`components.izp.runtimeDirectory`	The path where the SMP/E installer places the runtime files. The path where the SMP/E installer places the runtime files. This location may be read-only so modifications should be done in `{components.izp.workspaceDirectory}`. If you accepted the SMP/E installation default, the path name will be `/usr/lpp/IBM/izp/v1r2m0/bin`.
`components.izp.workspaceDirectory`	The writable location where Unified Management Server can store files and other data important to core functionality. This should not be the same as `{components.izp.runtimeDirectory}/ums/var`. If you make your workspace directory inside of your runtime directory, then validation will fail and you will not be able to proceed with starting the server.
`components.izp.migrationDirectory`	If applicable, the path of Unified Management Server 1.1 read/write directory. This is the value of the `IZP_UMS_VARDIR` variable. It is not supported to use the same value for `workspaceDirectory` above.
`components.izp.security`	Properties that will interface with your security manager.
`components.izp.security.pkcs11`	Encryption using ICSF PKCS #11 services.

Value	Description
`components.izp.security.dbaUser`	The user name of dba that goes with the encrypted password.
`components.izp.security.token`	The PKCS #11 token where the secret key material is stored.
`components.izp.security.library`	Path to PKCS #11 provider module.
`components.izp.security.certificate`	Used to secure communication over https.
`components.izp.security.certificate.allowSelfSigned`	Whether Unified Management can use self-signed certificates.
`components.izp.security.certificate.truststore`	Contains trust material used by Unified Management Server.
`components.izp.security.certificate.truststore.location`	The path or user and key ring name to a trust store.
`components.izp.security.certificate.truststore.type`	The type of trust store that is being used.
`components.izp.security.certificate.keystore`	Contains key material used by Unified Management Server.
`components.izp.security.certificate.keystore.location`	The path or user and key ring name to a trust store.
`components.izp.security.certificate.keystore.type`	The type of trust store that is being used.
`components.izp.security.certificate.keystore.alias`	The name of the alias of the server certificate.
`components.izp.security.certificate.allowSelfSigned`	Boolean. Set to true by default.
`components.izp.security.profilePrefix`	Prefix to generic profiles created during installation. The profiles are used to denote roles.
`components.izp.security.profilePrefix.super`	Super role
`components.izp.security.profilePrefix.admin`	Administrator role
`components.izp.security.surrogateUser`	If `useSAFOnly=false` for `surrogateUser` and `surrogateGroup`, password-less users will be created during installation.
`components.izp.security.surrogateGroup`	The default group for above users.

Value	Description
`components.izp.server`	Properties that are read by Java to configure the server when running.
`components.izp.server.authType`	The desired authentication type for Unified Management Server. It supports two authentication types, which are STANDARD_JWT and MFA_JWT. The default is STANDARD_JWT. For details on these authentication types, see the following sections: • "Credential management by UMS JWT tokens" on page 25 for STANDARD_JWT • "Credential management by Zowe JWT tokens" on page 27 for MFA_JWT
`components.izp.server.port`	The port that Unified Management Server will listen on.
`components.izp.server.log`	The log location. Specify FILE, STDOUT (i.e. job output), or BOTH
`components.izp.server.apiRateCapacityByUser`	The maximum number of HTTP requests per user.
`components.izp.server.memorySize`	The memory to be allocated for Unified Management Server in megabytes.
`components.izp.server.jobPrefix`	The job code for Unified Management Server jobs. The jobPrefix associated for a team overrides this value.
`components.izp.server.failsafeTimeout`	Default value is 100 minutes.
`components.izp.server.graphQLTimeout`	Default value is 300 seconds.
`components.izp.server.objectDiscoveryInterval`	How frequently to refresh object discovery for registered subsystems in hours.
`components.izp.server.allSysnames`	A space separated list of lpar names that you would like to do system discovery on.
`components.izp.server.javaArgs`	An array of -D arguments to pass to the server.
`components.izp.server.tlsVersionList`	Comma separated list of enabled TLS protocols used by Unified Management Server as a client and server. Default value is `TLSv1.2,TLSv1.3`, indicating both TLS 1.2 and TLS 1.3 versions are allowed. **Note:** You must be running Zowe 2.12.0 for this feature to be available.
`components.izp.server.host`	Optional parameter. The host that UMS is running on. Default value is the first element in `Zowe.externalDomains`. Enter a host name if UMS is running on a different host. For example, if you are using a host or port distributor such as DVIPA.
`components.izp.dataset.loadLibrary.izp`	Optional parameter. The data set name where the UMS load library resides. If this value is blank, the default data set (`runtimeHlq.SIZPLOAD`) is used. If you move the library, you need to specify this data set name to override the default.

Value	Description
`components.izp.dataset`	These data sets will be used during installation and run time. They shouldn't exist unless otherwise specified. The rest will be created at some point during the configuration process.
`components.izp.dataset.runtimeHlq`	The hlq where the SMP/E data sets reside.
`components.izp.dataset.hlq`	The high-level qualifier for miscellaneous data sets created during runtime and installation.
`components.izp.dataset.parmlib`	The location of your PARMLIB dataset. Default value is `{components.izp.dataset.hlq}.PARMLIB`.
`components.izp.dataset.jcllib`	The location for JCL members generated by `IZPGENER`. Default value is `{components.izp.dataset.hlq}.JCLLIB`.
`components.izp.dataset.loadLibrary`	Load libraries for various parts of Unified Management Server.
`components.izp.dataset.dbaEncryption`	Contains encrypted password of the dba user in yaml format after running the dba encryption shell script.
`components.izp.security.useSafOnly`	An alternative security mode that caches stored team, user, and role information from SAF. This has eliminated the need to manage the USERLIST or TEAMLIST data sets. This solution works on all three External Security Managers (ESM) by using RACROUTE. **Note:** If the value is set to true, the `components.izp.dataset.userList` and `components.izp.dataset.teamList` parameters are ignored.
`components.izp.dataset.userList`	Contains cache of UMS user data.
`components.izp.dataset.teamList`	Contains cache of UMS team data. This data set should never be modified manually.

`components.izp.toolsDiscovery`	Contains the necessary information for UMS to discover external tools. For example:
	`toolsDiscovery:`
	`  enabled: true`
	`  discoverySearchPaths:`
	`  - "DIR:/path/to/discover/tools"`
	`  - "DSN:HLQ.LLQ.DISCOVER.DATASET(MEMBER)"`

`components.izp.toolsDiscovery.enabled`	Allows discovery of external tools. **Note:** This parameter is not applicable to IMS Tools. For IMS Tools and IMS Administration Foundation, the configuration member `IZPIMFPM` in the UMS PARMLIB is used. If the `IZPIMFPM` configuration parameters are properly specified, then IMS Tools and their maintenance levels are automatically detected by UMS. For details, see Installing IMS Administration Foundation.

Value	Description
`components.izp.toolsDiscovery.discoverySearchPaths`	The list of locations the discovery process will search to find files that contain external tool definitions. For details on the configuration, see the following topics: • "Configuring UMS for Db2 Analytics Accelerator Loader" on page 88 • "Configuring UMS for External Tooling" on page 89 • "Configuring Storage tab" on page 76 **Note:** This parameter is not applicable to IMS Tools.
`components.izp.zowe.job.suffix`	Allows you to configure the UMS task name. The default value is `IZP`. The UMS task name is composed of the following elements: • Prefix: Zowe job name (`zowe.job.prefix` element in `zowe.yaml`) • X: The X character is used to denote this as a Zowe extension. It is inserted between the prefix and suffix. • Suffix: Zowe job suffix (`components.izp.zowe.job.suffix` element in ZWEYAML) The default UMS task name will be ZWEXIZP. If the UMS task name string is longer than eight characters, it will be truncated.
`zowe`	These zowe items are required by the IZP setup. Do not edit.

Refer to the `example-izp.yaml` file for detailed explanation of parameters listed in the ZWEYAML member.

8. The `IZPGENER` JCL is required for generating other jobs that are needed to configure the Unified Management Server. Edit and submit `IZPGENER` from your copy of `SIZPSAMP`.

   Note: Add required datasets to the `SYSPROC` specification in `IZPGENER`.

   Replace the following values and submit the `IZPGENER` JCL:

Value	Description
`{components.izp.dataset.runtimeHlq}`	Replace it with the HLQ of your SMP/E installed data sets.
`{zowe.setup.dataset.loadlib}`	Replace it with the data set that contains the Zowe executable. **Note:** `zowe.*` yaml elements are specified in the zowe yaml file.
`{components.izp.runtimeDirectory}`	Replace it with your IZP runtime directory. This is the same location where you placed SMP/E files in USS.
`{zowe.runtimeDirectory}`	Replace it with your Zowe runtime directory.
`{components.izp.dataset.parmlib}`	Replace with the location of the UMS PARMLIB.

The `IZPGENER` JCL generates the following members in the data set `{components.izp.dataset.jcllib}` as specified in `ZWEYAML`. The default is `{components.izp.dataset.hlq}.JCLLIB`.

Member	Description
ESM specific numbered IZP prefixed members.	• Members ending with "V" are meant for validating the specific JCL. Some validation members will end with a letter indicating the ESM. • Files ending with "R" are RACF specific members. • Files ending with "T" are TSS specific members. • Files ending with "A" are ACF2 specific members.
IZPEXPIN	Launches the IZP experience copy script.
IZPIPLUG	Installs Zowe plugins using the zwe command.
IZPSTEPL	Concatenates a dataset into the Zowe AUX started task definition.
IZPUSRMD	Configures user list dataset. **Note:** This can be used only when `useSafOnly` is set to `false`.

The following JCLs apply to all ESMs:

Member	Description
IZPA1	Allocate PDSE (extended partitioned dataset) for teamList. This is not required if `useSAFOnly=true`.
IZPA2	Allocate sequential (flat) dataset for userList. This is not required if `useSAFOnly=true`.
IZPA3	Allocate sequential (flat) dataset for dba Encryption.

The RACF specific JCLLIB members are listed in the following table:

Member	Description
IZPB1R	Create IZP class and add to the CDT.
IZPB2R	Add security role profiles to the IZP class.
IZPB3R	Create generic profiles to secure userList and teamList data sets. This is not required if `useSAFOnly=true`.
IZPC1R	Add surrogate users to impersonate when accessing the userList and teamList data sets during runtime. This is not required if `useSAFOnly=true`.
IZPC2R	Grant surrogate users access to the userList and teamList profiles. This is not required if `useSAFOnly=true`.
IZPD1R	Define CRYPTOZ resource profiles for the PKCS #11 token for UMS. **Note:** The PKCS #11 token was specified by the key `components.izp.security.pkcs11.token` in the ZWEYAML PROCLIB member.
IZPD2R	Grant system programmer and started task access to PKCS #11 resources.
IZPD3R	Create the PKCS #11 token for UMS.

Member	Description
IZPD4R	Optional: Add a new user to serve as the DBA user ID. If the user already exists, then there isn't a need to run this job. If you do so anyways, you can ignore a non-zero return code. **Note:** The DBA user ID was specified by the key `components.izp.security.pkcs11.dbaUser` in the `ZWEYAML PROCLIB` member.
IZPD5R	Connect the DBA user ID to the IZUUSER group for z/OSMF.
IZPD6R	Grant the DBA user ID access to applications. **Note:** If the APPL class is active and `OMVSAPPL` is defined, submit the job `IZPD6R` to permit `IZPSRGSP`, `IZPSRGAD`, DBA user ID (`components.izp.security.pkcs11.dbaUser`), UMS user, and the Zowe STC user (`ZWESLSTC`) read access on the `OMVSAPPL` resource.
IZPD7R	Creates function profiles in IZP class that are used when `useSafOnly` is enabled, which allow users to refresh the security cache. The user should permit those who should have the ability to refresh the security cache after making changes in SAF.

The TSS specific JCLLIB members are listed in the following table:

Member	Description
IZPB0T	Create a new group for surrogate users. This is not required if `useSAFOnly=true`. **Note:** Group should not exist. If the group already exists, then do not run this job. If you decide to run it, you need to ignore a non-zero return code.
IZPB1T	Create IZP class and add it to the RDT.
IZPB2T	Add security role profiles to the IZP class.
IZPB3T	Create generic profiles to secure userList and teamList data sets.
IZPC1T	Add surrogate users to impersonate when accessing the userList and teamList data sets during runtime. This is not required if `useSAFOnly=true`.
IZPC2T	Grant surrogate users access to the userList and teamList profiles. This is not required if `useSAFOnly=true`.
IZPD1T	Define our user and security officer resources for the PKCS #11 token.
IZPD2T	Grant system programmer and started task access to PKCS #11 resources.
IZPD3T	Create a new PKCS #11 token.
IZPD4T	Connect the DBA to the IZUUSER group for z/OSMF.
IZPD5T	Create function profiles for useSafOnly security mode and grant read access to the dbaUser.

The ACF2 specific JCLLIB members are listed in the following table:

Member	Description
IZPB0A	Create IZP roles in X(ROL).
IZPB1A	Create IZP class and add it to the RDT.

Member	Description
IZPB2A	Add security role profiles to the IZP class.
IZPC1A	Add surrogate users to impersonate when accessing the userList and teamList data sets during runtime. This is not required if useSAFOnly=true.
IZPC2A	Create profiles to secure userList and teamList data sets and grant access to surrogate users. This is not required if useSAFOnly=true.
IZPD1A	Make the rules for crypto resource residents in INFODIR.
IZPD2A	Grant system programmer and started task access to PKCS #11 resources.
IZPD3A	Create a new PKCS #11 token.
IZPD4A	Connect the DBA to the IZUUSER group for z/OSMF.
IZPD5A	Create function profiles for useSafOnly security mode.

9. Run the following JCLs from your JCLLIB:

   - IZPA1
   - IZPA2
   - IZPA3

   IZPA1 and IZPA2 are not required if you migrated from UMS 1.1 and are retaining the same USERLIST and TEAMLIST datasets or if useSAFOnly=true.

10. When using Zowe version 2.5.0 and below, edit and submit IZPCPYM2 from your copy of SIZPSAMP. The purpose of IZPCPYM2 is to handle the versions of Zowe which do not fully support PARMLIBs, and thus require a USS file for starting Zowe. You run this job in order to place the ZWEYAML member into a location on USS that you specify. You can change the file name from izp.yaml to another filename if needed.

    Replace the following values and submit the IZPCPYM2 JCL:

Value	Description
{directory}	Replace it with the path of the izp.yaml file. The directory location must match the value that you use in IZPIPLUG @izp_yaml substitution, and when you update the Zowe started task.
{components.izp.dataset.parmlib}	Replace it with the location of PARMLIB.

11. Run ESM JCLs generated by the IZPGENER JCL from your JCLLIB, as needed.

    Run the following JCLs after running ESM specific files:

    - IZPSTEPL
    - (Optional) IZPUSRMD

      If you are migrating from UMS 1.1, you don’t need to run the IZPUSRMD JCL.

      Note: This JCL can be used only when useSafOnly is set to false.

12. Encrypt DBA credentials.

    As an install user, run izp-encrypt-dba.sh from your UMS installation location. You need to provide the high-level qualifier of the environment data set. This is the YAML variable listed as {components.izp.dataset.hlq}.

```
{components.izp.runtimeDirectory}/ums/opt/bin/izp-encrypt-dba.sh
{components.izp.dataset.hlq}
```

    Replace {components.izp.runtimeDirectory} and {components.izp.dataset.hlq}, on your command line, with the values of these parameters in the ZWEYAML member.

The `izp-encrypt-dba.sh` shell script creates the `components.izp.dataset.dbaEncryption` data set used to store the encrypted DBA credential. It is not recommended to use OMVS to complete this step because the password is visible on the screen in plain text. Use SSH to perform this step.

13. Edit and submit `IZPIPLUG`.

    If you are using Zowe version 2.5.0 or below, you can submit the `IZPIPLUG`:

Value	Description
`@izp.yaml`	Replace it with the location of the IZP yaml file in USS that was created by `IZPCPYM2`.

    Note: If you want to rerun `IZPIPLUG`, you must remove the USS soft link file specified by `{zowe.extensionDirectory}/izp` or Zowe will give an error that IZP is already installed.

14. Submit `IZPEXPIN` to install experiences.

    A customized copy of `IZPEXPIN` JCL should have been built and placed in `JCLLIB` based on the parameters that you specified in the member `ZWEYAML` of the UMS `PARMLIB`.

    Important:

    - You must verify the result of the `IZPEXPIN` job for each experience product to activate.
    - If any Db2 Experience is enabled, the following fields must have values in `IZPDB2PM` `PARMLIB` member. Follow the instructions below after the `IZPEXPIN` job run.

Value	Description	Sample value
`IZP_DB2_USR_HLQ`	High-level qualifier (HLQ) for user data sets created and written to during various JCL Jobs execution.	`HLQ.IZP.DSN`
`IZP_DB2_USR_PREFIX`	Prefix for user data sets created and written to during various JCL Jobs execution.	`SIZP`

    - If you are installing Admin Foundation and do not have IBM Db2 Administration Tool, the following fields must have values:

Value	Description	Sample value
`IZP_DB2_ADB_HLQ`	High-level qualifier (HLQ) is used by Admin Foundation to extract various modules. For example, single object DDL generation.	`HLQ.IZP.DSN`
`IZP_DB2_ADB_PREFIX`	Prefix for Admin Foundation data sets must be SAFX.	`SAFX`

    - If you are installing Admin Foundation and have IBM Db2 Administration Tool or if you are installing Db2 DevOps Experience, the following fields must have values:

Value	Description	Sample value
`IZP_DB2_ADB_HLQ`	High-level qualifier (HLQ) for Db2 Administration Tool (ADB) for related data sets to be concatenated in various generated JCLs.	`HLQ.ADB.DSN`
`IZP_DB2_ADB_PREFIX`	Prefix for Db2 Administration Tool (ADB) for related data sets to be concatenated in various generated JCLs.	`SADB`

    - If Db2 DevOps Experience is enabled, the following two fields must have values in `IZPDB2PM` `PARMLIB`.

Value	Description	Sample value
IZP_DB2_GOC_HLQ	High-level qualifier (HLQ) for Db2 Object Comparison Tool (GOC).	HLQ.GOC.DSN
IZP_DB2_GOC_PREFIX	Prefix for Db2 Object Comparison Tool (GOC).	SGOC

- If IMS Administration Foundation is enabled, follow the configuration steps that are described in the topic "Installing IMS Administration Foundation" on page 77.

15. Update the Zowe started task.

    You need to update the Zowe started task JCL to recognize the IZP yaml specification. The default Zowe started task name is ZWESLSTC.

    The file name izp.yaml is the file name you specified in the IZPCPYM2 JCL in step 10. If you changed the file name, you must use the name you specified. Note that if you are using Zowe version 2.5.0 or below, you must run the IZPCPYM2 JCL every time you modify the ZWEYAML member in the UMS PARMLIB and before you restart the Zowe server. The Zowe version 2.6.0 or above will access the PARMLIB values directly through the PARMLIB member.

    Important: While specifying the config attribute value, the IZP yaml specification must be listed first, followed by the Zowe YAML file. Refer to the notes in the JCL for detailed instructions on specifying the YAML path. The Zowe started task has a CONFIG= line in its STDENV section to specify the input yaml configuration. Depending on the version of Zowe, you need to add the Unified Management Server information, either as an izp.yaml file or as a ZWEYAML PARMLIB member. For Zowe versions 2.3.0 to 2.5.0, use the following syntax to specify the UMS information as a file:

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

    For Zowe versions 2.6.0 and above, use the following syntax to specify the UMS information as a PARMLIB member:

    ```
    CONFIG=PARMLIB(<Location of PARMLIB>(ZWEYAML))\
    :FILE(/path/to/zowe.yaml)
    ```

    Prior to the specification of the CONFIG= line, ensure that the STDENV section in the Zowe started task also lists the following line:

    ```
    _CEE_ENVFILE_CONTINUATION=\
    ```

16. Start ZSS cross-memory servers on all LPARs.

    Refer to Zowe documentation for details on starting a ZSS cross-memory server.

    Its associated cross-memory auxiliary server address space will be started automatically by the Zowe cross-memory server. You do not need to manually start the address space.

17. Start Zowe.

    Refer to Zowe documentation for detailed steps on starting Zowe.

What to do next

To validate the UMS installation, see "Step 3: Validating the installation" on page 67.

Step 3: Validating the installation

If everything went well, you should be able to access the Zowe URL, log in to Zowe, and view and start the UMS web interface that after successful validation, will appear with the title **IBM Unified Experience for z/OS**.

To verify if the installation is successful, perform the following steps:

1. In a supported browser, open the following URL:

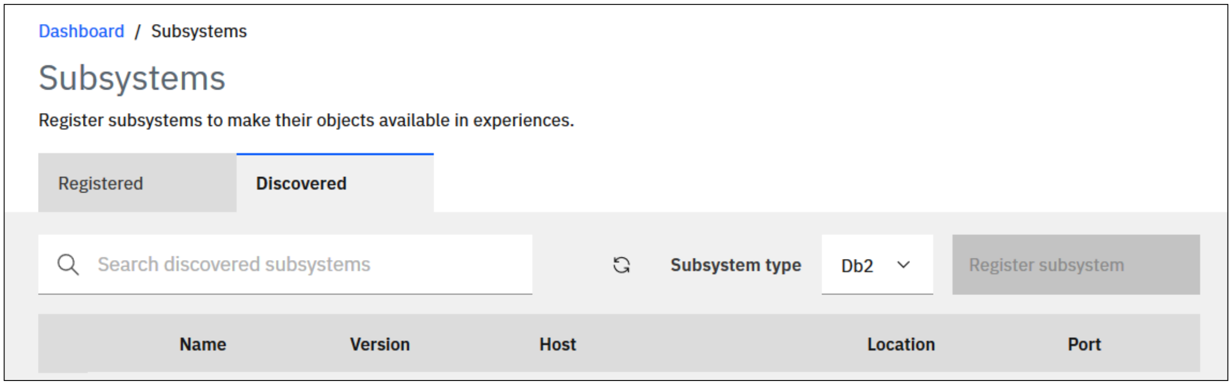
```
https://<hostname>:<portname>/ZLUX/plugins/org.zowe.zlux.bootstrap/web/#/
```

where:

<hostname>: Host name or IP address of the Zowe App Server. The host name and IP address are specified as an array value for the key `zowe.externalDomains` in the `zowe.yaml` file.

<portname>: Port of the Zowe App Server host. The port is specified by the value for the key `components.app-server.port` in the `zowe.yaml` file.

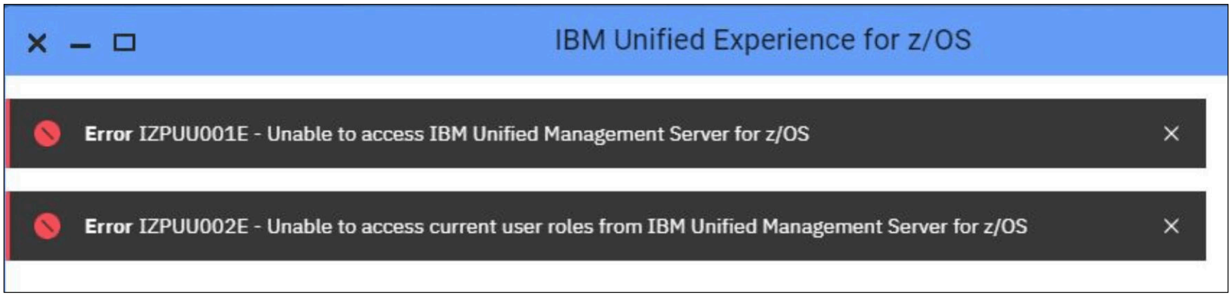
2. Log in to Zowe.
3. Navigate to the Zowe Desktop Start menu, and click > **IBM Unified Experience for z/OS**. The **IBM Unified Experience for z/OS** welcome screen is displayed for a few seconds, followed by the **Subsystems** page.



Subsystems

Register subsystems to make their objects available in experiences.

Note: You might get errors if the Unified Management Server is not up and running.



Error IZPUU001E - Unable to access IBM Unified Management Server for z/OS

Error IZPUU002E - Unable to access current user roles from IBM Unified Management Server for z/OS

For more information on these errors, see Chapter 10, “Messages,” on page 187.

4. Navigate further by clicking the **Discovered** tab on the **Subsystems** page.

Congratulations: You have successfully installed the Unified Management Server!

Post-installation tasks

There are several standard maintenance procedures that you can complete after installing Unified Management Server. Some of these procedures include starting and stopping Unified Management Server, changing the default Zowe certificate to your own HTTPS certificate, updating the database administrator credentials, adding new users to UMS, and deactivating a user in UMS.

- “Disabling TCP timestamps” on page 69
- “Applying local Java security properties” on page 69
- “Updating database administrator credentials” on page 69
- “Adding and deleting user records” on page 70
- “Reviewing UMS server logs” on page 72
- “Controlling the log file destination” on page 73
- “Defining a security class for UMS” on page 259

After installing Unified Management Server, if you want to tune SQL queries by leveraging the features of the SQL Tuning Services, complete the steps that are described in "Configuring UMS for SQL Tuning Services for Db2" on page 86.

Disabling TCP timestamps

The OS fingerprinting probe examines the remote server's implementation of TCP timestamps. Timestamps provide the attacker with a means to guess the operating system of the target.

For z/OS, TCP timestamps can be disabled by executing the following steps:

1. Find the TCP configuration data set, named as `Profile.TCPIP` or similar.
2. Edit the TCP configuration data set by appending `NOTCPTIMESTAMP` in the line `TCPCONFIG`.
3. Using a user with `CONTROL` access, run **`VARY TCPIP,SYNTAXCHECK, tcpConfigurationDatasetName`** to validate the syntax of the configuration file.
4. Using a user with `CONTROL` access, run **`VARY TCPIP,,OBEYFILE, tcpConfigurationDatasetName`** to apply the changes.

For more information, see:

- Security considerations for the **VARY** command
- Understanding the **`VARY TCPIP,,SYNTAXCHECK`** command
- Understanding the **TCPCONFIG** statement

Applying local Java security properties

It is recommended to apply local Java security properties to contain any security impact within UMS.

The UMS server relies on Java security properties to configure the SSL/TLS security features. There are known vulnerabilities existing in SSL/TLS protocols, such as `Lucky13, Logjam, BEAST`, and so on. For more information, see IBM Java™ Secure Socket Extension (JSSE) Provider.

In order to allow UMS to leverage the most secure cipher suites, it is recommended to apply local security properties rather than the default IBM Java security properties. Using local security properties would ensure changes only apply to UMS and will not affect other programs using Java

Note: Make sure all services, including Zowe and ZSS services, share strong cipher suites, otherwise the SSL/TLS handshake will not succeed.

Perform the following steps:

1. Copy `${JAVA_HOME}/lib/security/java.security` to `<alternate_location>`.
2. Specify this alternate location in the `components.izp.server.javaArgs` location of `PARMLIB(ZWEYAML)`. For example:

```
javaArgs:
  - -Djava.security.properties=<alternate_location>
```

Updating database administrator credentials

If the database administrator credentials change (for example if the password changes), you must update them in Unified Management Server. You must start the UMS server after regenerating the credentials.

Procedure

1. Navigate to `runtime-directory/ums/opt/bin`, where runtime directory is defined in `ZWEYAML` as `components.izp.runtimeDirectory`. This must be same as the SMP/E installed location.
2. Ensure that the following variables are added to your z/OS UNIX System Services profile, or issue export commands for these variables to get them applied in an individual session:

```
export _BPXK_AUTOCVT=ON
export _CEE_RUNOPTS="FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)"
export _TAG_REDIR_IN=TXT
export _TAG_REDIR_OUT=TXT
export _TAG_REDIR_ERR=TXT
```

3. To run the credential update script, enter the following command:

   ```
   ./izp-encrypt-dba.sh
   ```

   Note: The user who runs this script must have CONTROL access to previously created the PKCS #11 token and also the password to the DBA user. An optional parameter of the UMS read/write HLQ is permitted. If not, you will be prompted for this value. This HLQ contains the ENVIRON data set which is used to access Zowe functionality.
4. If the variables are set in `ZWEYAML` member of `PARMLIB` then those values are used, otherwise, you will be prompted.
5. When prompted, specify updated values for the database administrator username and password. If you have specified a username for database administrator in `components.izp.security.pkcs11.dbaUser` in `ZWEYAML` member of `PARMLIB` then you must either remove that value (to be prompted at the command line) or update it within `ZWEYAML` member of `PARMLIB`. You will only be prompted for Db username if the variable is not set in `ZWEYAML` member of `PARMLIB`.
6. Stop and start Unified Management Server.

Adding and deleting user records

As part of business operations, you can add new users and remove old users from the UMS. Sometimes, deleting user records might affect the functionality of the application.

If you specified true for the `useSafOnly` parameter, the following section is not applicable. For user management when the `useSafOnly` parameter is `true`, see "Setting up users and teams" on page 46.

Notes:

- The **IZPUSRMD** script backs up your data set before updating the user records.
- When using the ACF2 ESM, Unified Management Server needs a UNIX environment variable soft link in the home directory of the user running the `IZPUSRMD` job. To create this link, before running `IZPUSRMD`, run the following UNIX System Services command as the user who will be adding and removing users:

  ```
  ln -e ACFUNIX ~/acfunix
  ```

IZPUSRMD starts the user list modification script, which enables you to add and remove users from UMS. It is located in the *<components.izp.dataset.jcllib>* library after `IZPGENER` has been run.

1. Select the *<components.izp.dataset.jcllib>*`(IZPUSRMD)` JCL job for your system.
2. Modify the JCL to change parameters as specified in the comments and submit the job.

```
//IZPUSERL  JOB ,'IZP USER LIST'
//* CONFIGURE USER LIST DATA SET
//*
//* NOTES:
//*
//*    This script can be run as many times as needed and should be the
//*    only means of configuring the IZP User List Data Set. If you'd like
//*    to do this process manually, please use the dry run option. That
//*    option can be used to generate the proposed state of the data set
//*    as well as any commands that need to be run for every user. You may
//*    then take that information and manually edit the data set and grant
//*    or revoke access to the security profiles that were created in
//*    IZPSECUR.
//*
//* 1) YOU MUST EDIT THIS JCL WITHOUT ISPF NUMBERING (NUM OFF).
//*    NOTE:  SOME SUBSTITUTIONS MAY HAVE ALREADY BEEN EXECUTED BY IZP
//*    POST INSTALL PROCESS.
//*
//* 2) THIS MEMBER IS CASE SENSITIVE. YOU MUST EDIT WITH CAPS OFF.
```

```
//*
//* 3) CHANGE #dry_run_option TO INDICATE WHETHER OR NOT YOU WANT PERMIT
//*    COMMANDS TO BE EXECUTED FOR YOU AS WELL AS THE USER LIST DATA SET
//*    MODIFIED.
//*
//*    VALID VALUES:
//*
//*    DRY_RUN_OPTION=-DRYRUN
//*    DRY_RUN_OPTION=-dryrun
//*    DRY_RUN_OPTION=
//*
//* 4) CHANGE #connect_option TO INDICATE WHETHER OR NOT YOU WANT USERS
//*    CONNECTED TO OR REMOVED FROM IZUUSER.
//*
//*    VALID VALUES:
//*
//*    CONNECT_OPTION=-CONNECT
//*    CONNECT_OPTION=-connect
//*    CONNECT_OPTION=
//*
//* 4) CHANGE #action TO INDICATE WHETHER YOU WANT TO ADD OR REMOVE
//*    USERS.
//*
//*    VALID VALUES:
//*
//*    ACTION=ADD
//*    ACTION=add
//*    ACTION=REMOVE
//*    ACTION=remove
//*
//* 5) CHANGE #saf_role TO BE THE ROLE THAT WILL BE PERMITTED TO.
//*
//*    VALID VALUES:
//*
//*    SAF_ROLE=SUPER
//*    SAF_ROLE=super
//*    SAF_ROLE=ADMIN
//*    SAF_ROLE=admin
//*
//* 6) CHANGE #users_and_groups TO A SPACE SEPARATED LIST OF USERS AND
//*    AND GROUPS THAT YOU WOULD LIKE TO BE PERMITTED TO SAF_PROFILE.
//*
//*    EXAMPLE:
//*
//*    USERS_AND_GROUPS=USERA USERB GROUPA GROUPB
//*
//*-------------------------------------------------------------------*
//SET1 SET UMSVLOC='{components.izp.runtimeDirectory}'
//SERVER    EXEC PGM=BPXBATCH,REGION=800M,TIME=NOLIMIT,
//   PARM='SH &UMSVLOC/ums/opt/bin/izp-modify-users.sh'
//STDOUT   DD SYSOUT=*
//STDENV   DD *
IZP_HLQ={components.izp.dataset.hlq}
DRY_RUN_OPTION=#dry_run_option
CONNECT_OPTION=#connect_option
ACTION=#action
SAF_ROLE=#saf_role
USERS_AND_GROUPS=#users_and_groups
_BPXK_AUTOCVT=ON
_CEE_RUNOPTS=FILETAG(AUTOCVT,AUTOTAG) POSIX(ON)
_TAG_REDIR_IN=TXT
_TAG_REDIR_OUT=TXT
_TAG_REDIR_ERR=TXT
/*
```

Notes:

- {components.izp.runtimeDirectory} is automatically set to the high-level qualifier for miscellaneous read/write data sets. No modification is required.
- Valid values for the ACTION parameter are ADD and Remove.
- There are two ways to use #dry_run_option:
  - -DRYRUN or -dryrun - When you run the user list script with this option, nothing is modified.
  - Blank - When you run a script without the -dryrun option. This backup data set is overwritten after doing a normal run with the last state of the user list.

  Here is an example of a dry run:

```
DRY_RUN_OPTION=-DRYRUN
CONNECT_OPTION=
ACTION=ADD
SAF_ROLE=SUPER
USERS_AND_GROUPS=USERA GROUPA USERB GROUPB
```

Here is an example of a normal run:

```
DRY_RUN_OPTION=
CONNECT_OPTION=
ACTION=ADD
SAF_ROLE=SUPER
USERS_AND_GROUPS=USERA GROUPA USERB GROUPB
```

- There are two ways to use `#connect_option`:
  - `-CONNECT` or `-connect` - When you run the user list script with the connect option, additional commands will be printed or run, depending on whether or not you specify the dry run option. If you specify the dry run option, these additional commands will be printed instead of run.
  - Blank - If you don't want the users to be connected to **`IZUUSER`**, run the script without the `-connect` option.

  Here is an example of the `-connect` option:

  ```
  DRY_RUN_OPTION=
  CONNECT_OPTION=-CONNECT
  ACTION=ADD
  SAF_ROLE=SUPER
  USERS_AND_GROUPS=USERA GROUPA USERB GROUPB
  ```

- `#saf_role` should be `SUPER` or `ADMIN` depending on the user role. You will need to run this script once with `SUPER` and once with `ADMIN`.
- `#users_and_groups` is an inline list of TSO IDs and groups.

  Here is an example of the `#users_and_groups` list:

  ```
  USERS_AND_GROUPS=TSOID1 TSOID2 TSOID3 GROUP1 GROUP2
  ```

- After the script is run successfully, the following files are populated with data in the `/tmp/ums-`*`<username>`* temporary directory, where username is the user who invoked IZPUSRMD:
  - `izpUserlist` - This file contains the current or proposed state of the user list.
  - `izpCommands` - This file contains a list of commands that would be run, or in the case of dry run, need to be run.
- Both `izpUserlist` and `izpCommands` are copied to separate sequential data sets in the case of a dry run.
- The names of the following data sets are generated by the script. They follow the following naming scheme:
  - *`components.izp.dataset.hlq.USERLIST`*`.PROPOSED`
  - *`components.izp.dataset.hlq.USERLIST`*`.COMMANDS`

Important:

An issue has been found with ACF2 dry run option, such that the `USERLIST.COMMANDS` data set might be truncated to 80 columns.

Reviewing UMS server logs

At this point of the installation process, you might want to review the server logs for any possible monitoring and troubleshooting.

UMS logs are generated in the Zowe log folder, specified by the Zowe yaml location `zowe.logDirectory`. The internal log for graph_server is generated in the `components.izp.workspaceDirectory`/log directory. If the <Number> value in the log file name is greater than zero, it means the log file name was rotated due to exceeding the maximum length and a new log file was created. The latest log will be numbered 0, the older log will be numbered 1, and so on.

UMS generates multiple log files:

`izp-server-`*`<instanceID>`*`-`*`<TimeStamp>`*`.`*`<Number>`*`.log`
: This is the main log. It contains entries that are related to the UMS server operations. The *<instanceID>* will be either the SYSNAME of the lpar or the high availability instance identifier.

There are additional specialized logs:

`graph_server.`*`<Number>`*`.log`
: This is a log dedicated to the modeling component of the UMS. It contains entries related to the graph database.

`izp-start-`*`<TimeStamp>`*`.log`
: This is a startup log. A new log entry is generated every time the IZP component is started.

`izp-stop-`*`<TimeStamp>`*`.log`
: This is a shutdown log. A new log entry is generated every time the IZP component is stopped.

Controlling the log file destination

This section guides you through an optional procedure for controlling server log files.

Note: You must start and stop the UMS server before changing the log file settings.

The log destination is specified in the ZWEYAML member of the UMS PARMLIB by using the key `components.izp.server.log.destination`. This variable takes one of the following values: FILE, STDOUT, or BOTH. If the value of the variable is not specified, the `components.izp.server.log.destination` takes the default value, BOTH. Logs will be placed in the `zowe.logDirectory` folder on the z/OS UNIX System Services, with a date stamp.

`components.izp.server.log.destination: FILE`

Running this command logs to the USS file only.

`components.izp.server.log.destination: STDOUT`

Running this command logs to "standard output" of started task only. Be aware of the size of the log if debug is turned on.

`components.izp.server.log.destination: BOTH`

Running this command is similar to using STDOUT option, however, it also writes the output to a single z/OS UNIX System Services log file. Be aware of potential for large log size.

Log rotation details:

To control the size and count of logs, modify the ZWEYAML member of the UMS PARMLIB by using the key `component.izp.server.javaArgs` as an array of Java arguments.

- `- -Dcom.rocketsoft.logCount=max_number_of_logs`
- `- -Dcom.rocketsoft.logLimit=size`

  where, size is the number of bytes, the number of kilobytes appended with k, or the number of megabytes appended with m. For example, 1048576, 1024k, and 1m are equivalent.
- Log file rotation is supported. By default, the logs are capped at 10 MB, and a maximum of *logCount* logs will be kept. The default value of *logCount* is 32.

Installing a program temporary fix (PTF)

The PTF contains new features and fixes. To apply these improvements to your product, you must install the PTF.

Before you begin

Use your system programmer credentials. You must shut down UMS and Zowe processes before applying the program temporary fix (PTF) in SMP/E. Upgrade instructions are required only if you have previously configured Unified Management Server prior to applying the program temporary fix. If Unified Management Server 1.2 and PTF are an initial install, complete the steps mentioned in "Post-installation tasks" on page 68.

About this task

As a system programmer, you need to shut down UMS and all Zowe processes. You can then proceed to apply the program temporary fix (PTF) in SMP/E.

Procedure

1. Optional: Run `IZPSYNCY` if performing an upgrade.

   The `IZPSYNCY` job imports new elements from `example-izp.yaml` into your `ZWEYAML`. Copy this member to the location where prior members of `SIZPSAMP` were copied. Replace the following values:

   a) Replace `{v1.2_parmlib}` with the dataset where your v1.2 `PARMLIB` is located.

   b) Replace `{components.izp.dataset.runtimeHlq}` with the HLQ where SMP/E installed data sets are located. You need to replace this value at three places in `IZPSYNCY`.

      Note: IZPSYNCY will take the entire structure from the source, including the comments, and insert values from the existing ZWEYAML, thus removing the user comments that were added during IZPSYNCY upgrades.
2. Rerun `IZPGENER` to regenerate the working JCLs from the PTF source versions.
3. Edit and run `IZPIPLUG` to update Zowe with the current UMS code base.
4. Restart Zowe.

Installing Db2 Administration Foundation

To work with and manage Db2 for z/OS on your subsystems, you must have Db2 Administration Foundation installed and configured properly.

Before you begin

1. Review the software requirements for Db2 Administration Foundation. For details, see the Software requirements for Db2 Administration Foundation section.
2. The latest PTF for IBM Unified Management Server for z/OS must be installed.
3. You need to retrieve the UMS installation folder path on z/OS UNIX System Services and the *`config_hlq`* of your UMS installation data sets.

   Note: This is not the installation path. It is the path of `components.izp.workspaceDirectory`.
4. The SMP/E install for Db2 Administration Foundation should be in the same SMP/E global zone as UMS.

Procedure

1. Shut down Zowe.

   a) Go to System Display and Search Facility (SDSF) in 3270.

   b) Stop ZSS cross-memory server.

```
/p <job_name>
```

If Zowe defaults have been chosen, `job_name` should be replaced with `ZWESISTC`.

c) Stop Zowe server.

```
/p <job_name>
```

If Zowe defaults have been chosen, `job_name` should be replaced with `ZWESLSTC`.

2. To perform the Db2 Administration Foundation SMP/E installation, see the *Db2 Administration Foundation Program Directory*. Note the z/OS UNIX location where Db2 Administration Foundation is installed.
3. To copy modules into libraries for Db2 experiences and to create a `PARMLIB` member `IZPDB2PM`, which are required in later steps, complete the following steps:

   a) Add the experience SMP/E installation location to `ZWEYAML` array element `components.izp.experiences`:

   ```
   experiences:
       - /uss/db2-afn/installation/directory
   ```

   The default location is: `/usr/lpp/IBM/afn/`*`<vrm>`*`/bin`.

   b) Submit `JCLLIB(IZPEXPIN)`. You can run `IZPEXPIN` multiple times if you need to install multiple experiences. It is necessary to include all your installed experiences in the `components.izp.experiences` array.
4. If you receive a return code other than `0`, examine the output, correct errors, and resubmit the job.
5. Grant users the access to the Db2 catalog tables by completing the steps described in "Granting catalog access to Db2 tables" on page 90.
6. Start Zowe.

   a) Start ZSS cross-memory server.

   ```
   /s <job_name>,reusasid=yes
   ```

   If Zowe defaults have been chosen, *job_name* should be replaced with `ZWESISTC`. You must specify `reusasid=yes` after *<job_name>*.

   b) Start Zowe server.

   ```
   /s <job_name>
   ```

   If Zowe defaults have been chosen, *job_name* should be replaced with `ZWESLSTC`. Replace *<zowe_instance>* with the absolute path of your Zowe instance directory.
7. In a supported browser, log in to Zowe at the following URL and open IBM Unified Experience for z/OS.

   ```
   https://<hostname>:<port>/ZLUX/plugins/org.zowe.zlux.bootstrap/web/#/
   ```

   where *<hostname>* and *<port>* are the host name (or IP address) and port of the Zowe host computer.
8. Optional: Verify that Db2 Administration Foundation is correctly installed by completing the steps mentioned in "Verifying the installation of Db2 Administration Foundation " on page 91.

What to do next

If you want to generate the DDL of an explored object, you must configure a Workload Manager (WLM) environment for your subsystem:

1. Create a WLM environment by using the template in `JCLLIB(WLMPROC)` that points to the load libraries in *<HLQ>*`.SIZPLLIB`.
2. Start the WLM by using the template in `JCLLIB(WLMPROC)` that points to the WLM you created in step 1.

3. While registering the subsystem, specify the name of the WLM in the **Workload Manager environment** field on the **Configuration** tab.

   Note: If the subsystem is already registered, edit the subsystem, specify the WLM environment, and click **Update**.

Configuring Storage tab

Before you enable the Storage tab for index and tablespace, you must configure Db2 Administration Foundation to use Db2 Administration Tool.

Before you begin

- Ensure that Db2 Administration Foundation is installed and APAR PH54968 is applied.
- Ensure Db2 Administration Tool version 13 is installed and APAR PH55177 is applied.

Procedure

To enable the Storage tab, complete the following steps:

1. Navigate to the ZWEYAML member of PARMLIB.
2. Enable discovery by setting `components.izp.toolsDiscovery.enabled` to `true`.
3. Specify the locations of the YAML files in `components.izp.toolsDiscovery.discoverySearchPaths`:

   ```
   - "DSN:hlq.SHLOSAMP(ADB131P)"
   ```

   For example:

   ```
   toolsDiscovery:
     enabled: true
     discoverySearchPaths:
     - "DSN:HLQ.SHLOSAMP(ADB131P)"
   ```

4. Restart Zowe.

   Zowe is configured for Db2 Administration Tool.

Configuring Statistics tab

Before you enable the Statistics tab for index and tablespace, you must configure Db2 Administration Foundation to use Db2 Administration Tool.

Before you begin

- Ensure that Db2 Administration Foundation is installed and APAR PH56033 is applied.
- Ensure Db2 Administration Tool version 13 is installed and APAR PH55177 is applied.

Procedure

To enable the Statistics tab, complete the following steps:

1. Navigate to the ZWEYAML member of PARMLIB.
2. Enable discovery by setting `components.izp.toolsDiscovery.enabled` to `true`.
3. Specify the locations of the YAML files in `components.izp.toolsDiscovery.discoverySearchPaths`:

   ```
   - "DSN:hlq.SHLOSAMP(ADB131P)"
   ```

   For example:

```
toolsDiscovery:
  enabled: true
  discoverySearchPaths:
  - "DSN:HLQ.SHLOSAMP(ADB131P)"
```

4. Restart Zowe.

   Zowe is configured for Db2 Administration Tool.

Installing IMS Administration Foundation

To work with and manage IMS on your subsystems, you must have IMS Administration Foundation installed and configured properly.

Before you begin

1. Review the software requirements for IMS Administration Foundation. For details, see "Software requirements for IMS Administration Foundation" on page 39.
2. To enable IMS Administration Foundation base features on an UMS server instance, you must complete the following:
   a. Install the HAFN170 feature of IBM IMS Tools Base for z/OS 1.7.0 (program number: 5655-V98). The installation of the FMID HAFN170 must be completed as a part of the SMP/E installation of the IBM IMS Tools Base for z/OS. For details of the SMP/E installation, see the Program Directory for IBM IMS Tools Base for z/OS.
   b. Apply the PTF UI90630 that was provided by the APAR PH49803 for the FMID HAFN170 and succeeding PTFs. Each PTF for IMS Administration Foundation FMID HAFN170 requires a specific PTF or PTFs for Unified Management Server FMID IZP120. For details, see "UMS PTF compatibility" on page 17.
3. You must follow the UMS installation steps to activate IMS Administration Foundation features. Confirm that you have followed the installation steps in the topic "Step 2: Installing Unified Management Server" on page 56 up to the step for running an `IZPEXPIN` JCL for the IMS Administration Foundation. You must confirm the following:
   a. The PARMLIB location of the UMS server instance on which you activate IMS Administration Foundation and the UMS configuration member `ZWEYAML` in the `PARMLIB`.
   b. A correct IMS Administration Foundation runtime USS directory is specified in the array value for the key `components.izp.experiences` of the UMS configuration member `ZWEYAML` in the UMS `PARMLIB`.
   c. The `IZPEXPIN` was run as a configuration step for UMS after you have added the IMS Administration Foundation runtime USS directory to the `ZWEYAML` member in the `PARMLIB`.
4. You must set up security to use some of the IMS Administration Foundation features. The requirements for security set up are as follows:
   - Security setup for the auxiliary address space of Zowe cross-memory server
   - Security setup for IMS Administration Foundation
   - Setting up security for IMS Connect servers if you plan to use the IMS SQL processor feature.
     - This is required for a secure connection from UMS to the IMS Connect servers.
     - If you want to use UMS authentication type of MFA_JWT, see Enabling IMS Connect to receive RACF PassTickets.
     - If you are to use multifactor authentication for UMS, see Configuring multifactor authentication for UMS.
   - Setting up security for IMS Tools TCP server if you plan to use any of the IMS Tools features.
     - This is required for a secure connection from UMS to the IMS Tools TCP server.

  - If you want to use UMS authentication type of MFA_JWT, see Enabling IMS Tools TCP server to receive RACF PassTickets, and TCP server configuration parameters in *IBM IMS Tools Base for z/OS 1.7 Configuration Guide*.
  - If you want to use multifactor authentication for UMS, see Configuring multifactor authentication for UMS.
- Security setup for IMS Administration Tool
  - This is required only when you want to use IBM IMS Administration Tool for z/OS (also referred to as IMS Administration Tool) from the IMS Administration Foundation.

Procedure

1. If you received a return code other than 0 for the IZPEXPIN JCL, examine the output, correct errors, and resubmit the job.
2. Confirm that a new UMS configuration members `IZPIMFPM` and `IZPIMSPM` have been created in the UMS `PARMLIB`.
3. To configure IMS Administration Foundation, specify the following parameters in `IZPIMFPM` in YAML format. You can use the parameter values in the original `IZPIMFPM` member as a sample for specifying parameter values in YAML format.

   imsUtilExeclib

   Specifies the EXEC library SAFNEXEC that was installed by IBM IMS Tools Base for z/OS using SMP/E. This is a required parameter.

   ```
   Example:
   imsUtilExeclib:
   - HLQ.SAFNEXEC
   ```

   imsToolsLoadlib

   Specifies the IMS Tools load module libraries to be used for backend IMS Tools processing. This is an optional parameter. If you want to register at least one IMS data sharing group for which the IMS catalog is not used, you must specify the SHKTLOAD load module library that was installed by IMS Tools Base for z/OS. If you want to use the DBD and PSB map features, you must specify the SHPSLMD0 load module library of IBM IMS Library Integrity Utilities for z/OS. For details of the relationship between IMS Administration Foundation features, IMS Tools products, prerequisite setup, and configuration for servers provided by the IBM IMS Tools Base for z/OS, see "IMS Administration Foundation and IMS Tools" on page 249.

   ```
   Example:
   imsToolsLoadlib:
   - HLQ.SHKTLOAD
   - HLQ.SHPSLMD0
   ```

   imsToolsServers

   Specifies the IBM IMS Tools Base servers that are used to provide the following features:

   - Reports for a DBD
   - Statistics for a DBD, a HALDB partition, or a DEDB area
   - Database exceptions and databases with exceptions
   - DBD and PSB map feature provides visualization of database segment tree structure defined by a DBD or by a DB PCB in a PSB

   This parameter is optional. If it is specified, all of the following three sub parameters must be specified as a group in YAML format:

   - `dai`

     Specifies the host address of the TCP server of IBM IMS Tools Base for z/OS Distributed Access Infrastructure (DAI). All of the following sub parameters must be specified:

     - `address`

    Specifies the host address or IP address of the TCP server. If the secure connection is used by specifying true for the sub parameter `sslConnection`, consult your security administrator on how to specify the host address.

  - `port`

    Specifies the port number of the TCP server.

  - `sslConnection`

    Specifies if the port is secured by the TLS protocol. If the port is secured, specify `true`. If the port is not secured, specify `false`. If the port is secured, the UMS server must be able to access at least the root CA certificate that was used to sign the DAI server's certificate chain. If the root CA certificate is missing in the truststore to be used by the UMS server, import the CA certificate into the truststore file or connect the CA certificate to the keyring to be used by the UMS server. For details, see "Setting up security for IMS Tools TCP server" on page 257.

  - `applName`

    Specifies the APPL name for the subject TCP server. The APPL name must be same as the name specified for the SecurityAppl configuration parameter for the TCP server. This parameter is optional, but it is mandatory when you want to use the UMS authentication type of MFA_JWT. For details of TCP server parameters, see TCP server configuration parameters of *IBM IMS Tools Base for z/OS Configuration Guide*. For details on security setup requirements for the TCP server, see Setting up security for IMS Tools TCP server. If you want to use multifactor authentication, see also Configuring multifactor authentication for UMS.

- `itkb`

  Specifies the IBM IMS Tools Base IMS Tools Knowledge Base (IMS Tools KB) server. The following sub parameter must be specified:

  - `xcfGroupName`

    Specifies the XCF group name of the IMS Tools KB server group to be used.

- `ad`

  Specifies the IBM IMS Tools Base Autonomics Director (AD) server. The following sub parameter must be specified:

  - `xcfGroupName`

    Specifies the XCF group name of the AD server group to be used. This is the name prefixed with the character string "IAV" and followed by the string that is specified by the `XCFGROUP` parameter in the `IAVPCOM` member of the `PROCLIB` for the subject AD server. For example, if `XCFGROUP=ADM00` is specified, specify the `xcfGroupName` parameter as shown in Example 1 below.

If two or more pairs of IMS Tools KB server and AD server are used, you can repeat the group of these sub parameters. Multiple pairs of IMS Tools KB server and AD server can share a same DAI TCP server. For details of configuration options for IBM IMS Tools Base servers, see *IBM IMS Tools Base for z/OS Configuration Guide*.

```
Example 1: Specifying a single set of IMS Tools KB server and AD server with a DAI TCP
server
imsToolsServers:
- dai:
     address: localhost
     port: 5123
     sslConnection: false
  itkb:
     xcfGroupName: FPQSERVR
  ad:
     xcfGroupName: IAVADM00
```

```
Example 2: Specifying two sets of IMS Tools KB server and AD server with a single common
DAI TCP server
imsToolsServers:
- dai:
     address: zos1.example.com
     port: 5123
     sslConnection: true
  itkb:
     xcfGroupName: FPQSRVR1
  ad:
     xcfGroupName: IAVADM01
- dai:
     address: zos1.example.com
     port: 5123
     sslConnection: true
  itkb:
     xcfGroupName: FPQSRVR2
  ad:
     xcfGroupName: IAVADM02
```

```
Example 3: Specifying two sets of IMS Tools KB server and AD server with different DAI
TCP servers
imsToolsServers:
- dai:
     address: zos1.example.com
     port: 5123
     sslConnection: true
  itkb:
     xcfGroupName: FPQSRVR1
  ad:
     xcfGroupName: IAVADM01
- dai:
     address: zos2.example.com
     port: 5123
     sslConnection: true
  itkb:
     xcfGroupName: FPQSRVR2
  ad:
     xcfGroupName: IAVADM02
```

Note: If you want to use the DBD and PSB Map feature or the feature to register IMS subsystems and data sharing groups for IMS Administration Tool, you may need additional setup for the TAS servers of IMS Tools Base DAI and IMS Security. For more details, see IMS Administration Foundation and IMS Tools.

4. Edit the `IZPIMSPM` member. The following parameters are used by IMS Administration Foundation:

   imsTempDatasetHLQ

   Specifies the high-level qualifier (HLQ) for temporary data sets that are created during operations of jobs submitted by the Unified Management Server for IMS Administration Foundation functions. The value must be 1 to 8 characters in length. This is an optional parameter. If this parameter is not specified, the DBA user ID that was specified by the value for the key `components.izp.security.pkcs11.dbaUser` in the UMS configuration member ZWEYAML of the UMS PARMLIB.

5. Set up IMS OM command security for the stared task ID used for the auxiliary address space of the Zowe cross-memory server. For details, see "Security setup for the auxiliary address space of Zowe cross-memory server" on page 81.
6. Set up IMS Connect security if you want to use either of the following features:

   - IMSplex, IMS, and IMS Connect pages
   - IMS command processor
   - IMS SQL processor

   For details of the setup, see the topic "Setting up security for IMS Connect servers" on page 256.
7. Set up resource access permissions for UMS users and the UMS DBA user ID. For details, see "Security setup for IMS Administration Foundation" on page 81.
8. Navigate to the remaining configuration for the Unified Management Server and restart the Zowe server or the Unified Management Server as a component of Zowe. For details, see "Step 2: Installing Unified Management Server" on page 56.

What to do next

If you want to use any IMS Administration Foundation feature, you must first register at least one IMS data sharing group. For details, see "Registering IMS subsystems as a data sharing group" on page 114.

If you have already registered one or more IMS data sharing groups and modified the UMS authentication types, then the UMS super administrator may need to edit each of those registrations and reconfigure the IMS Connect port settings to comply with the selected authentication type. If you have changed the authentication type from STANDARD_JWT to MFA_JWT and you are using or intend to use the IMS command processor, you must specify an appropriate APPL name that can be used for PassTicket generation and validation for the authentication of the selected port.

If you see an error status displayed for a registered IMS data-sharing group, do the following:

1. Go to **Subsystems**>**Discovered** tab, and then select an IMS subsystem type.
2. Click the **Refresh** icon.
3. After the IMS subsystem list is refreshed, go back to the **Registered** tab.
4. Select a registered IMS data sharing group for which an error status is displayed.
5. Open its overflow menu and select the Refresh object discovery menu item.

Note: Repeat steps 4 and 5 for each registered IMS data sharing group for which an error status is displayed. These object discovery requests will be processed one by one. It takes a while for an object discovery to be complete.

Security setup for IMS Administration Foundation

IMS Administration Foundation uses some IMS commands in its backend processes to perform various actions. IMS Administration Foundation uses type-2 commands wherever possible. Type-1 commands such as /DISPLAY ACT are used in the IMS page of the IMS components. All commands, including the type-1 commands, are issued through the IMS CSL Operations Manager; thus the CSL OM command security apply if they are activated.

Some IMS commands are issued with the login user's credentials as a result of the user interface operations. For details of those IMS commands, see "IMS commands issued with the login user credentials" on page 249.

Some IMS commands are issued with the credentials of the DBA user ID of Unified Management Server to discover IMS resources and IMS component configurations. For information on the IMS commands issued with the DBA user credential, see "IMS commands issued from the DBA user ID" on page 247.

Some IMS DBRC commands are issued with the credentials of the DBA user ID of Unified Management Server to discover IMS resources registered to RECON data sets. For information on the DBRC commands issued with the DBA user credential, see "IMS DBRC commands issued from the DBA user ID" on page 247.

Security setup for the auxiliary address space of Zowe cross-memory server

UMS uses some of the IMS-specific services that are provided by programs running in the auxiliary address space of Zowe cross-memory server.

Depending on the IMS Administration Foundation features that you use, the following additional security configurations are required for the auxiliary address space:

- Mandatory security setup for IMS subsystem discovery
- Additional security setup for IMS subsystem discovery for IMS Administration Tool

Mandatory security setup for IMS subsystem discovery

If IMS Administration Foundation is installed on UMS, IMS subsystem discovery uses IMS type-2 commands for collection of detailed information on IMS subsystems. Therefore, the IMS Common Service Layer (CSL) and the Operations Manager (OM) must be configured accordingly.

Important: This step is required if you set up the IMS OM security or IMS type-2 command security for an IMS subsystem that can run on the z/OS system on which the Zowe cross-memory server is or will be running. This requirement is independent of whether you use an IMS Administration Foundation feature or not for the UMS server.

For more information about the subsystem discovery, refer to "Key concepts" on page 105.

The data collector portion of the IMS subsystem discovery will run in an auxiliary address space of Zowe cross-memory server. For more information about the Zowe cross-memory server and its auxiliary address spaces, see the Zowe documentation.

Additional permissions may be required for the started task user ID that is used for the auxiliary address space of Zowe Cross Memory server. The ID that is assigned to the started task for the auxiliary address space must have the following authorities. .

- The ID must be allowed to use IMS type-2 commands that are listed in IMS commands issued from the Zowe auxiliary address space. To define SAF security profiles for those type-2 commands, see the following topics in *IMS System Administration*:
  - CSL SCI security
  - CSL OM command security
- The ID must have the read access authority to the STEPLIB and DFSRESLB data sets allocated to each IMS control region on the sysplex LPARs on which the Zowe cross-memory servers are running. This is required only when IMS modules `CSLSRG00` and `CSLSDR00` are not in the LINKLIST. These modules are required to register and deregister from SCI.
- The ID must have the read access authority to the STEPLIB and IMSDALIB data sets for the IMS DBRC region.

Important: If you did not do this IMS security setup for the started task ID to be used for the auxiliary address space of Zowe cross-memory server and for an IMS subsystem before you start the UMS server, the UMS server will report one or more authorization errors for that IMS subsystem.

Additional security setup for IMS subsystem discovery for IMS Administration Tool

If you want to use the IMS Administration Tool from IMS Administration Foundation, you need to perform an additional security setup for the started task user ID used for the auxiliary address space of Zowe cross-memory server.

If the IMS Administration Tool is used for IMS Administration Foundation, job logs of some of IMS system address spaces for IMS subsystems registered by IMS Administration Foundation are accessed from the auxiliary address space of the Zowe cross-memory server during the IMS data sharing group registration process. The following IMS system address spaces are accessed to get the IMS system configuration information that is required for IMS registration processing by the IMS Administration Tool:

- IMS control regions for IMS subsystems that belong to the data sharing group to be registered
- DBRC region and DLISAS region jobs started by those IMS control regions
- IRLM used for those IMS subsystems

If the JES spool access is protected by ESM resource profiles (such as JESSPOOL profiles of RACF), resource access violations can occur. In this case, you will see one or more access violation messages such as ICH408I in RACF. To avoid resource access violation, you need to permit the started task user ID of the auxiliary address space for the Zowe cross-memory server with READ access to the job logs for the IMS system address spaces of all IMS subsystems that you plan to register with IMS Administration Foundation using IMS Administration Tool.

Security setup for IMS Administration Tool

If you want to use the IMS Administration Tool from IMS Administration Foundation, you need to ensure that the security setup is completed for the IMS Administration Tool and the UMS super administrator who will be registering the IMS data sharing group has access to the services provided by the IMS Administration Tool.

The following security requirements must be met to use the IMS Administration Tool:

- The UMS DBA user and all UMS users who are planning to use the IMS Administration Tool features including RECON ID creation and IMS registration during the IMS data sharing group registration must have READ access to the profile ATYADMIN.ACCESS in the RACF FACILITY class.
- If the profile ATYADMIN.SETUP is created in the RACF FACILITY class, the UMS DBA user and the administration user who will be registering the IMS data sharing group must have READ access to the profile.

For details on the security setup, see Secure the IMS Administration Tool functions in the *IBM IMS Administration Tool for z/OS User Guide and Reference*. For details on the other setup required for IMS Administration Tool, see Product configuration in the *IBM IMS Administration Tool for z/OS User Guide and Reference*.

If IMS Administration Tool is used for IMS Administration Foundation, job logs of some of the IMS system address spaces for IMS subsystems that belong to an IMS data sharing group that is to be registered by IMS Administration Foundation are accessed from the auxiliary address space of the Zowe cross-memory server during the IMS data sharing group registration process. For details on the security requirements, see Security setup for the auxiliary address space of Zowe cross-memory server.

Installing data management experiences

Data management experiences are installed in a similar way as Unified Management Server.

Before you install data management experiences, you must complete Unified Management Server installation.

Steps to install a data management product:

1. Perform the SMP/E installation by following steps in the *Program Directory* for the data management product.
2. Edit a provided sample job and specify where in the z/OS UNIX file system the data management product and Unified Management Server are installed.
3. Submit the job. A script will copy data management product files to z/OS UNIX where UMS is installed and create one or more PARMLIB members that contain parameters that are specific to the data management product or the subsystem. Here, the PARMLIB is the z/OS MVS data set that is where ZWEYAML is expected to be located.
4. Update the parameters and run a data management product post-installation script from z/OS UNIX .

Note: If you are using spool management or archive system, ensure that the jobs submitted by this product are not immediately removed from the spool. Jobs should remain in the spool for the duration of the operation, which may be several minutes.

Installing Db2 DevOps Experience

To work with and manage Db2 for z/OS on your subsystems, you must have Db2 DevOps Experience properly installed and configured.

Before you begin

1. Review the software requirements for Db2 DevOps Experience. For details, see the Software requirements for Db2 DevOps Experience section.
2. The latest PTF for IBM Unified Management Server for z/OS must be installed.
3. You need to retrieve the UMS installation folder path on z/OS UNIX System Services and the *RUNHLQ* of your UMS installation data sets.

   Note: This is not the installation path. It is the path of `components.izp.workspaceDirectory`.

4. The SMP/E install for Db2 DevOps Experience should be in the same SMP/E global zone as UMS.
5. Install the FMIDs: H0IHD10 and H25GD10. These are included with the restricted license for Db2 DevOps Experience, or you can reuse an existing installation of Db2 Administration Tool and Db2 Object Comparison Tool. The HLQs are required for the libraries of FMIDs (H0IHD10 and H25GD10) during installation, see "Step 2: Installing Unified Management Server" on page 56.

Procedure

1. Shut down Zowe.
   a) Go to System Display and Search Facility (SDSF) in 3270.
   b) Stop ZSS cross-memory server.

   ```
   /p <job_name>
   ```

   If Zowe defaults have been chosen, `job_name` should be replaced with `ZWESISTC`.

   c) Stop Zowe server.

   ```
   /p <job_name>
   ```

   If Zowe defaults have been chosen, `job_name` should be replaced with `ZWESLSTC`.
2. To perform the Db2 DevOps Experience SMP/E installation, see the *Db2 DevOps Experience Program Directory*. Note the z/OS UNIX location where Db2 DevOps Experience is installed.
3. To copy modules into libraries for Db2 experiences and to create `DOEDB2P0 JCL` and a `PARMLIB` member `IZPDB2PM`, which are required in later steps, complete the following steps:
   a) Add the experience SMP/E installation location to `ZWEYAML` array element `components.izp.experiences`:

   ```
   experiences:
       - /uss/db2-devops/installation/directory
   ```

   The default location is: `/usr/lpp/IBM/doe/`*`<vrm>`*`/bin`.

   b) Submit `JCLLIB(IZPEXPIN)`. You can run `IZPEXPIN` multiple times if you need to install multiple experiences. It is necessary to include all your installed experiences in the `components.izp.experiences` array.

   c) Confirm that the `JCLLIB(IZPEXPIN)` job populated the `gitDir` and `gitEnv` values in the `IZPD2DPM PARMLIB` member.
4. To set information necessary for later steps, specify the following parameters in the data management product `PARMLIB` member `IZPDB2PM`, which will be created by `IZPEXPIN` if not present in `PARMLIB`:

   cmbatchDsn

   In addition to the above stated parameters, there is a commented out parameter in the platform `IZPD2DPM` parameter member, *cmbatchDsn*, which is related to Db2. During Db2 subsystem registration, UMS will create several batch-related artifacts. The default location for these is `<dbaUser>.IZPBATCH.PROCLIB`. Uncomment the *cmbatchDsn* variable to modify this location. Note that this affects only the newly registered Db2 subsystems.

   The following parameters related to the ISPF message library are required:

   ISPFMessageLibrary

   ISPF message library

   ISPFSkeletonLibrary

   ISPF skeleton library

   ISPFTableLibrary

   ISPF table library

   ISPFLoadLibrary

   ISPF load library

ISPFLPALibrary
ISPF LPA library

defaultPlanName
Name of the default plan that Unified Management Server will always bind to a particular subsystem

The following parameters related to the ISPF message library are optional:

dbaSqlid
Owner of Db2 databases and table spaces. Any value will be converted to uppercase. Default value is '<NONE>'.

- `A SQLID`
  - The specified Run `SQLID` is the owner of databases and table spaces. If the specified Run `SQLID` is different from the current owner, the databases, table spaces, and all dependent objects are dropped and re-created to accomplish the change of owner.
- `<NONE>`
  - No `SET CURRENT SQLID` statements are generated.
- `blank`
  - `SET CURRENT SQLID` statements are generated when necessary.

tempDatasetHLQ
High-level qualifier (HLQ) for temporary data sets that are created while different JCL jobs are run. The length of the `tempDatasetHLQ` parameter must be 1 - 14 characters and the length of each qualifier in the `tempDatasetHLQ` parameter must be 1 - 8 characters. The name of each qualifier must begin with an alphabetic (A - Z) or a special character (#, @, $). The remaining characters in the name can be alphanumeric (A - Z, 0 - 9) or special characters (#, @, $).

5. Grant users the access to the Db2 catalog tables by completing the steps described in "Granting catalog access to Db2 tables" on page 90.
6. Optional: To specify the DDL definition files for the objects, define the DDL templates:
   a) Edit the `IZPD2DPM PARMLIB` member and specify `NICE_NAME: true`.

      Note: If you specify `NICE_NAME: false`, the files will be named according to the current naming format. For example, TB/`1af0b92a-8329-4dd2-986e-1e0ed2a8e1f9.txt.`

   b) Specify the attributes for the templates that you want to define.

   Note: For information about defining the templates, see " Db2 DDL file name definitions and validations" on page 241.
7. Start Zowe.
   a) Start ZSS cross-memory server.

      ```
      /s <job_name>,reusasid=yes
      ```

      If Zowe defaults have been chosen, *job_name* should be replaced with ZWESISTC. You must specify `reusasid=yes` after *<job_name>*.

   b) Start Zowe server.

      ```
      /s <job_name>
      ```

      If Zowe defaults have been chosen, *job_name* should be replaced with ZWESLSTC. Replace *<zowe_instance>* with the absolute path of your Zowe instance directory.
8. In a supported browser, log in to Zowe at the following URL and open IBM Unified Experience for z/OS.

   ```
   https://<hostname>:<port>/ZLUX/plugins/org.zowe.zlux.bootstrap/web/#/
   ```

   where *<hostname>* and *<port>* are the host name (or IP address) and port of the Zowe host computer.

9. Optional: Verify that Db2 DevOps Experience is correctly installed by completing the steps mentioned in "Verifying the installation of Db2 DevOps Experience" on page 94.

What to do next

To use the features of Db2 DevOps Experience, update the subsystem registration by completing the steps described in "Registering Db2 subsystems" on page 113.

If you want to generate the DDL of an explored object, you must configure a Workload Manager (WLM) environment for your subsystem:

1. Create a WLM environment by using the template in `JCLLIB(WLMPROC)` that points to the load libraries in *<HLQ>*`.SIZPLLIB`.
2. Start the WLM by using the template in `JCLLIB(WLMPROC)` that points to the WLM you created in step 1.
3. While registering the subsystem, specify the name of the WLM in the **Workload Manager environment** field on the **Configuration** tab.

   Note: If the subsystem is already registered, edit the subsystem, specify the WLM environment, and click **Update**.

Installing Db2 DevOps Experience PTF

It is recommended to install the program temporary fix after installing Db2 DevOps Experience. Use the following process for installing the PTF.

Procedure

1. Shut down UMS and all Zowe processes.
2. Apply the program temporary fix (PTF) in SMP/E.
3. Rerun `IZPEXPIN`.
4. Restart Zowe.

Results

The `IZPEXPIN` job is resubmitted and successful job completion will confirm the installation of Db2 DevOps Experience PTF.

Configuring UMS for SQL Tuning Services for Db2

Before you can tune the SQL queries for better performance, you must configure Unified Management Server for using SQL Tuning Services.

Before you begin

- Ensure that either Db2 DevOps Experience or Db2 Administration Foundation is installed.
- The instructions mentioned in this topic also apply to the Query Workload Tuner for z/OS that covers advanced tuning services. For more information, see Query Workload Tuner for z/OS documentation.
- Create the user IDs and assign them the required privileges and permissions. To know more about the different user roles, see the Setting up required user IDs and permissions table.
- Ensure that SQL Tuning Services with APAR PH39038 applied to it is up and running. To install SQL Tuning Services and enable SSL on the SQL Tuning Services server, complete the steps that are described in Installing and configuring SQL Tuning Services.

Procedure

To configure the SQL Tuning Services, complete the following steps:

1. By using the `<system_admin_id>` user ID, open the `PARMLIB` member `IZPDB2PM`.

2. Specify values for the following parameters:

   tuningHost
   The address of the host where SQL Tuning Services is running.

   tuningPort
   The port number that you must use when connecting to the host.

   tuningThroughHttps
   The default and preferred value of this parameter is `true`. Set this value to `false` to enable the HTTP connection from Unified Management Server to the SQL Tuning Services server.

   tuningDb2SecurityMechanismId
   The mechanism that you want to use for securing the connection to SQL Tuning Services. The default value of this parameter is 3.

   maxInMemorySize
   This parameter is used to configure the response buffer size for SQL Tuning Services. Before configuring this value, consider the number of concurrent users and environments in your organization. The default value of this parameter is `10 MB`.

   tuningApplicationId
   The linkname of the subsystem that is used to configure the repository database. This value must be consistent with the `appl_id` parameter set during the SQL Tuning Services configuration. This parameter is mandatory when MFA is enabled.
3. Save and close `IZPDB2PM`.
4. To enable secure communication between Unified Management Server and SQL Tuning Services, see "Setting up secure communication for UMS" on page 49.
5. Restart UMS.

   UMS is configured for SQL Tuning Services.

What to do next

- Create tuning profiles
- Tune SQL queries

Configuring UMS for Db2 Analytics Accelerator

Before you can manage Db2 Analytics Accelerator, you must configure Unified Management Server to use Db2 Analytics Accelerator Administration Services.

Before you begin

- Ensure that Db2 Administration Foundation is installed.
- Ensure that Db2 Analytics Accelerator 7.5.9.0 is up and running. To install Db2 Analytics Accelerator Administration Services and enable SSL on the Db2 Analytics Accelerator Administration Services server, complete the steps that are described in Installing and configuring Db2 Analytics Accelerator Administration Services.
- Create the user ID and assign it the required privileges and permissions. See the Setting up the required user ID and permissions table for more information.

Procedure

To configure Db2 Analytics Accelerator Administration Services, complete the following steps:

1. By using the `<system_admin_id>` user ID, open the `PARMLIB` member `IZPDB2PM`.
2. Specify values for the following parameters:

   idaaHost
   The address of the host where Db2 Analytics Accelerator Administration Services is running.

idaaPort

The port number that you must use when connecting to the host.

idaaThroughHttps

The default and preferred value of this parameter is `true`. Set this value to `false` to enable the HTTP connection from Unified Management Server to the Db2 Analytics Accelerator Administration Services server.

idaaDb2SecurityMechanismId

The mechanism that you want to use for securing the connection to Db2 Analytics Accelerator Administration Services. The default value of this parameter is 3.

maxInMemorySize

This parameter is used to configure the response buffer size for Db2 Analytics Accelerator Administration Services. Before configuring this value, consider the number of concurrent users and environments in your organization. The default value of this parameter is `10 MB`. The value of this parameter determines the ability to download the Db2 Analytics Accelerator for the zOS trace file using the Save trace functionality from the Db2 Administration Foundation.

3. Save and close `IZPDB2PM`.
4. To enable secure communication between Unified Management Server and SQL Tuning Services, see "Setting up secure communication for UMS" on page 49.
5. Restart UMS.

   UMS is configured for Db2 Analytics Accelerator Administration Services.

Configuring UMS for Db2 Analytics Accelerator Loader

Before you load table data to one or more accelerators in a subsystem, you must configure Db2 Administration Foundation to use Db2 Analytics Accelerator Loader.

Before you begin

- Ensure that Db2 Administration Foundation for z/OS is installed and APARs PH54968 and PH54452 are applied.
- Ensure Db2 Analytics Accelerator Loader for z/OS version 2.1 is installed and APAR PH54984 is applied.

Notes:

- Db2 Analytics Accelerator version 7.5 or later is supported.
- Db2 Analytics Accelerator groups or aliases are not supported.

Procedure

To configure UMS Zowe for Db2 Analytics Accelerator Loader, complete the following steps:

1. Navigate to the ZWEYAML member of PARMLIB.
2. Enable discovery by setting `components.izp.toolsDiscovery.enabled` to `true`.
3. Specify the locations of the Db2 Analytics Accelerator Loader YAML files in `components.izp.toolsDiscovery.discoverySearchPaths`:

   ```
   - "DSN:hlq.SHLOSAMP(HLODSCVP)"
   ```

   For example,

   ```
   toolsDiscovery:
     enabled: true
     discoverySearchPaths:
     - "DSN:HLQ.SHLOSAMP(HLODSCVP)"
   ```

4. Restart Zowe.

   Zowe is configured for Db2 Analytics Accelerator Loader.

5. Verify whether Tools Discovery has recognized and loaded the YAML file for Db2 Analytics Accelerator Loader by validating the message that appears in the Zowe log.

   For example,

   ```
   <discoverToolsIfEnabled> : Yamls processed during discovery:
   {5639-OLE={2.1.0=<ToolsDiscoveryEntry pathToProductYaml=<hlq.SHLOSAMP>(HLODSCVP),
   productYaml={product_id=5639-OLE, product_name=Db2 Analytics Accelerator Loader,
   product_ver=2, product_rel=1, product_mod=0, RVT=144}, instances=>} : Tools Discovery
   completed.
   ```

   Notes:

   - The user ID under which the Zowe server is running should have READ access to the data set `hlq.SHLOSAMP(HLODSCVP)`. To set up resource access permission, see "Setting up users and teams" on page 46.
   - Zowe must be restarted to apply the changes made to `hlq.SHLOSAMP(HLODSCVP)` data set.

Configuring UMS for External Tooling

Before discovering external tools, the Unified Management Server must be installed and configured properly. UMS discovers external tools by searching a list of YAML file locations that contain the definition or configuration of the tool. Some of the examples of external tools are Db2 Analytics Accelerator Loader and Db2 Administration Tool.

Before you begin

- Ensure Unified Management Server is installed.
- Identity the fields to be changed within the PARMLIB member ZWEYAML. See the Step 2: Installing Unified Management Server for more information.
- Ensure the started task user has the permissions required to access the YAML files for each tool.

Procedure

To configure External Tooling, complete the following steps:

1. Navigate to the ZWEYAML member of PARMLIB.
2. Enable discovery by setting `components.izp.toolsDiscovery.enabled` to `true`.
3. Specify the locations of the YAML files containing external tool definitions in `components.izp.toolsDiscovery.discoverySearchPaths`. A `discoverySearchPaths` entry begins with a prefix to help identify the location of the YAML file. A valid prefix is one of the following: DSN, DIR.

   a. For a UNIX System Services directory, use the following syntax:

   ```
   DIR:/path/to/example/directory
   ```

   b. For a MVS data set, use the following syntax:

   ```
   DSN:qualifier.for.dataset
   ```

   Note: If you are working with a partitioned data set, specify the member name. For example, `DSN:qualifier.for.partitioned.dataset(member)`.
4. Restart Zowe.

   Zowe is configured for External Tooling.

Configuring UMS after upgrading Db2 for z/OS

If you migrate from one version, mode, or function level of Db2 for z/OS to another, you need to make sure that IBM Db2 DevOps Experience for z/OS accounts for any changes in the Db2 catalog. If you update the IBM Db2 Administration Tool and IBM Db2 Object Comparison Tool libraries used by Db2 DevOps Experience, perform the following steps:

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Subsystems**.
3. Click the overflow menu of the required subsystem and make relevant changes.
4. Update the following:
    a. Subsystem or group that has been upgraded
    b. Version
5. Click **Update**.

Results

The subsystem registration job is resubmitted and it rebinds the packages. The job also automatically detects the new function level of Db2, if on Db2 12.

Granting catalog access to Db2 tables

Before you install and configure the data management products, make sure that you grant users the access on following Db2 catalog tables by running the following DCL script:

```
GRANT SELECT ON TABLE SYSIBM.SYSAUXRELS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCHECKS2 TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLDIST TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLDISTSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCOLUMNS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCONSTDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCONTEXT TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCONTEXTAUTHIDS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCONTROLS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCOPY TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSCTXTTRUSTATTRS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDATABASE TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDATATYPES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDEPENDENCIES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMY1 TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMYA TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDUMMYE TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDYNQRY TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSDYNQRYDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSENVIRONMENT TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSFIELDS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSFOREIGNKEYS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXPART TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXSPACESTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSINDEXSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSJARCONTENTS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSJAROBJECTS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSJAVAOPTS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSJAVAPATHS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSKEYCOLUSE TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSKEYS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTARGETS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTARGETSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTGTDIST TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSKEYTGTDISTSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSLEVELUPDATES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSLOBSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSOBDS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSOBJROLEDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKAGE TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKCOPY TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKLIST TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPACKSTMT TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPARMS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPENDINGDDL TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPENDINGOBJECTS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPKSYSTEM TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPLAN TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPLANDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSPLSYSTEM TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSRELS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSROLES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSROUTINES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSROUTINES_OPTS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSROUTINES_SRC TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSEQUENCES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSEQUENCESDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSESSION TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSESSION_EX TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSESSION_STATUS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSTMT TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSTOGROUP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSTRINGS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSSYNONYMS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABCONST TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLEPART TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLESPACE TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLESPACESTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABLES_PROFILES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTABSTATS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSTRIGGERS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSUTIL TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSUTILX TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSVARIABLES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSVIEWDEP TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSVIEWS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSVOLUMES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSXMLRELS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSXMLSTRINGS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSXMLTYPMOD TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.SYSXMLTYPMSCHEMA TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.XSRANNOTATIONINFO TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.XSROBJECTCOMPONENTS TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.XSROBJECTHIERARCHIES TO <Db2 data management product user>;
GRANT SELECT ON TABLE SYSIBM.XSROBJECTS TO <Db2 data management product user>;
```

Running installation verification program

The installation verification program verifies installation of the following data management products:

- Db2 Administration Foundation
- Db2 DevOps Experience

Verifying the installation of Db2 Administration Foundation

You can test the successful installation of Db2 Administration Foundation using steps listed in the following section. The installation verification program uses REST `APIs`, GET, PUT, and POST operations to verify the Db2 Administration Foundation installation verification program sample objects.

Before you begin

1. Ensure that the Unified Management Server and Db2 Administration Foundation are installed.
2. Ensure that the Db2 for z/OS installation verification program sample database is installed in one of the Db2 Subsystems to be used in the Unified Management Server/Db2 Administration Foundation installation verification program.
3. The installation verification program is a Python 3.0 program that confirms Python 3.0 environment availability. To install Python 3.0, see Python 3 Installation & Setup Guide. Ensure that Python 3.0 with the following plug-ins is installed on a workstation connected to the network.
    - `getpass`
    - `urllib` and `urllib3`

- requests
- json
- datetime
- pyyaml

Note: Other modules can be used, and must be available with modern UNIX and macOS platforms as per the Python documentation.

4. Ensure that you have an ID that is established as a Super User in the Unified Management Server/Db2 Administration Foundation installation. Ensure that the ID you create has access to read the selected Db2 catalog.

Procedure

1. **Accessing the installation verification program**: The installation verification program files are installed in the `components.izp.runtimeDirectory/ivp` folder. Use `sftp` or `ftp` in `bin` mode to copy all files into a dedicated directory on your workstation.

   The files contained in this folder are described below.

Table 35. The installation verification program folder contents

File name	Description
/Resources/Common/config.json	File for configuring the following UMS host and port variables: • IZP_UMS_HOST • IZP_UMS_HTTP_PORT
/Resources/adminFoundation/ivpAFDData.json	File for defining the default values of the installation verification program parameters. Ensure that you customize this file before running the installation verification program.
/Resources/db2_subsystems/ssid.NAME.json	If you are using the installation verification program to register a Subsystem, you must edit this template before running the installation verification program to create the Db2 Administration Foundation Subsystem.
/Reports	Stores the last run results.
afxivp.sh	File for running the Db2 Administration Foundation installation verification program.

2. **Setting up the installation verification program**: After the required installation verification program files are available in a directory on your workstation, customize the following files.

   a. Open the `config.json` file in a text editor. Specify values for the following UMS host and port variables:

      - IZP_UMS_HOST
      - IZP_UMS_HTTP_PORT

   b. If you need the installation verification program to register a Subsystem, customize the sample `ssid.NAME.json` file. If you already have a Subsystem registered in Db2 Administration Foundation and you want to use this Subsystem for the installation verification program, this file customization is not required.

   c. Copy the `ssid.NAME.json` file to a location where NAME is replaced with the SSID of your Db2 Administration Foundation Subsystem. To register a Db2 named DSNA, a file named `ssid.DSNA.json` must exist in the same directory as the Python program. This is the directory where the installation verification program files were extracted.

   d. Edit the `ssid.<ssid-name>.json` file attributes in a text editor.

Note: This file is used as an input into the Subsystem registration API call and must reflect the installation.

You must edit the following attributes in the ssid.<ssid-name>.json file.

Table 36. The ssid.<ssid-name>.json file attributes

Attribute name	Description
name	Change the ssid value to match the Db2 SSID or group name and the value used in the <ssid-name> node of the file name.
url	Change the url.com value to match the IP address or domain name used when making a JDBC connection to this Subsystem.
port	Change the port value to match the IP port used when making a JDBC connection to this Subsystem.
location	Change the location value to match the database location used when making a JDBC connection to this Subsystem.
sdsnexit	Change the prefix.SDSNEXIT value to match the SDSNEXIT library name for this Subsystem.
sdsnload	Change the prefix.SDSNLOAD value to match the SDSNLOAD library name for this Subsystem.
runlib	Change the prefix.RUNLIB.LOAD value to match the RUNLIB.LOAD library name for this Subsystem.
cmbatchDsn	Change the izpPrefix.PROCLIB value to match the PROCLIB chosen during installation.

Optionally, you can edit the following attributes in the ssid.<ssid-name>.json file.

Table 37. The ssid.<ssid-name>.json file attributes

Attribute name	Description
storageGroup	• Lists the Db2 Storage Group that is used when creating Db2 Administration Foundation instance objects in this Subsystem. • Replace SYSDEFLT with this Db2 Storage Group name, if needed.
version	Change this value to match the Db2 version for this Subsystem.
installWLM	Required for DDL generation.
plan	Change the default plan name, if needed.
installSchema	Db2 Administration Foundation objects created for tool management uses has a schema of IZPN01. Change this value, if needed.
installNamingRule	Db2 Administration Foundation objects created for tool management uses has a name starting with IZP1. Change this value, if needed.
dsntep2PlanName	If DSNTEP2 is bound with a plan name other than DSNTEP2, change this value.
jobcard	Review this attribute as an acceptable jobcard framework for your system.

The installation verification program runs remotely by using Unified Management Server/Db2 Administration Foundation APIs to invoke multiple back-end functions.

3. **Running the installation verification program**: From the directory where the installation verification program files are located, run the `afxivp.sh` script. This invokes Python 3.0. The script checks whether it is running in a Python 3.0 environment. The script stops if Python 3.0 is not available.
4. You are prompted for the following values:

   - UMS configuration

     The installation verification program gathers the environment details by default. Users can also update details by selecting options. Once the configuration details are set correctly, the values are saved for future use.
   - User ID
   - Password

     The password is not stored after running the script.
   - Db2 Administration Foundation data setup

     Specify a subsystem name. You can use the existing object name or add new objects to be used for the installation verification program.
5. At the end of the installation verification program, a summary report is displayed. You can select to save the summary report in the `/Reports` folder.
6. You can quit the installation verification program by using the following options:

   - q quits and still writes options and report files.
   - `q!` quits without writing the options or report files.

Verifying the installation of Db2 DevOps Experience

You can test the successful installation of Db2 DevOps Experience using steps listed in the following section. The installation verification program uses REST `APIs`, GET, PUT, and POST operations to verify the Db2 DevOps Experience installation verification program sample objects.

Before you begin

1. Ensure that the Unified Management Server and Db2 DevOps Experience are installed.
2. Ensure that the Db2 for z/OS installation verification program sample database is installed in one of the Db2 Subsystems to be used in the Unified Management Server/Db2 DevOps Experience installation verification program.
3. The installation verification program is a Python 3.0 program that confirms Python 3.0 environment availability. To install Python 3.0, see Python 3 Installation & Setup Guide. Ensure that Python 3.0 with the following plug-ins is installed on a workstation connected to the network.

   - `getpass`
   - `urllib` and `urllib3`
   - `requests`
   - `json`
   - `datetime`
   - `pyyaml`

   Note: Other modules can be used, and must be available with modern UNIX and macOS platforms as per the Python documentation.
4. Ensure that you have an ID that is established as a Super User in the Unified Management Server/Db2 DevOps Experience installation. Ensure that the ID you create has access to read the selected Db2 catalog.

Procedure

1. **Accessing the installation verification program**: The installation verification program files are installed in the `components.izp.runtimeDirectory/ivp` folder. Use `sftp` or `ftp` in `bin` mode to copy all files into a dedicated directory on your workstation.

   The files contained in this folder are described below.

   Table 38. The installation verification program folder contents

File name	Description
`/Resources/Common/config.json`	File for configuring the following UMS host and port variables: • `IZP_UMS_HOST` • `IZP_UMS_HTTP_PORT`
`/Resources/devopsExperience/doeivp.json`	File for defining the default values of the installation verification program parameters. Ensure that you customize this file before running the installation verification program.
`/Resources/devopsExperience/ivpSiteRules-db2`	This folder contains the JSON files to create sample site rules mentioned in the row above.
`/Resources/devopsExperience/IVPApp.json, IVPEnv.json, IVPInst.json, IVPTeam.json`	• Templates to create the Db2 DevOps Experience application, environment, instance, and team respectively. • The installation verification Python program customizes these templates to create Db2 DevOps Experience objects. • The customized templates are renamed as `IVP*2.json`.
`/Resources/db2_subsystems/ssid.NAME.json`	If you are using the installation verification program to register a Subsystem, you must edit this template before running the installation verification program to create the Db2 DevOps Experience Subsystem.
`/Reports`	Stores the last run results.
`doeivpSiteRulesdb2.sh`	A shell script to create site rules mentioned in `.json` files under the `Resources/devopsExperience/ivpSiteRules-db2` folder.
`doeivp.sh`	A shell script to start the appropriate python program by using Python 3.0.

2. **Setting up the installation verification program**: After the required installation verification program files are available in a directory on your workstation, customize the following files.

   a. Open the `config.json` file in a text editor. Specify values for the following UMS host and port variables:

      - `IZP_UMS_HOST`
      - `IZP_UMS_HTTP_PORT`

   b. Open the `doeivp.json` file in a text editor. It is recommended to customize this file for your installation because the installation verification program refers to this file for default values. If this file is not updated, several field updates are required while running the script.

      Although the following attributes are not mandatory, it is recommended that you edit them in the `doeivp.json` file.

Table 39. The doeivp.json file attributes

Attribute name	Description
ivpSSID	• Change this value to the Db2 Subsystem Identifier that is used by the installation verification program. For example: DSNA, DB2P. • Ensure that the Db2 Subsystem you select must have the Db2 for z/OS installation verification program database defined. For example: database DSN8DxxA, where xx denotes the Db2 version.
ivpSampDB	Change this value to the Db2 sample database name that is in the ivpSSID Subsystem.
ivpSampCR	Change this value to the Db2 sample database table creator name that is in the ivpSSID Subsystem.

Optionally, you can edit the following attributes in the doeivp.json file.

Table 40. The doeivp.json file attributes

Attribute name	Description
reportFile	• When the installation verification program is running, it creates reporting and options files. This value is used as the prefix for file names. • The same prefix is used to find and prune older files, as directed by the reportLimit attribute.
showSampDBInfo	• When validating Subsystems, Db2 is queried to ensure the required Sample Database objects exist. • Default value: 0. Used for basic reporting. • If you need detailed reporting, set this attribute value to 1.
ivpEnvironment	• This attribute value is used to name the Unified Management Server/Db2 DevOps Experience environment. • If the environment exists, it is used only after querying it for compatibility.
ivpTeam	• This attribute value is used to name the Unified Management Server/Db2 DevOps Experience team. • If the team exists, it is used only after querying it for compatibility.
ivpApplication	• This attribute value is used to name the Unified Management Server/Db2 DevOps Experience application. • If the application exists, it is used only after querying it for compatibility.
ivpInstance	• This attribute value is used as a name prefix to create the Unified Management Server/Db2 DevOps Experience instance. • A random number is appended to this prefix for a unique instance name.

Table 40. The `doeivp.json` file attributes (continued)

Attribute name	Description
`preSleep`	• When polling for the completion of asynchronous API calls, the program sleeps for the specified number of seconds before starting a polling loop. • Default value: `10`
`pollSleep`	• When polling for the completion of asynchronous API calls, the program sleeps for the specified number of seconds between GET calls to check for completion. • Default value: `20`
`loopLimit`	• When polling for the completion of asynchronous API calls, the program performs the iteration check for the specified number before timing out. • Default value: `20`

c. If you need the installation verification program to register a Subsystem, customize the sample `ssid.NAME.json` file. If you already have a Subsystem registered in Db2 DevOps Experience and you want to use this Subsystem for the installation verification program, this file customization is not required.

d. Copy the `ssid.NAME.json` file to a location where `NAME` is replaced with the SSID of your Db2 DevOps Experience Subsystem. To register a Db2 named DSNA, a file named `ssid.DSNA.json` must exist in the same directory as the Python program. This is the directory where the installation verification program files were extracted.

e. Edit the `ssid.<ssid-name>.json` file attributes in a text editor.

Note: This file is used as an input into the Subsystem registration API call and must reflect the installation.

You must edit the following attributes in the `ssid.<ssid-name>.json` file.

Table 41. The `ssid.<ssid-name>.json` file attributes

Attribute name	Description
`name`	Change the `ssid` value to match the Db2 SSID or group name and the value used in the `<ssid-name>` node of the file name.
`url`	Change the `url.com` value to match the IP address or domain name used when making a JDBC connection to this Subsystem.
`port`	Change the `port` value to match the IP port used when making a JDBC connection to this Subsystem.
`location`	Change the `location` value to match the database location used when making a JDBC connection to this Subsystem.
`sdsnexit`	Change the `prefix.SDSNEXIT` value to match the SDSNEXIT library name for this Subsystem.
`sdsnload`	Change the `prefix.SDSNLOAD` value to match the SDSNLOAD library name for this Subsystem.
`runlib`	Change the `prefix.RUNLIB.LOAD` value to match the `RUNLIB.LOAD` library name for this Subsystem.
`cmbatchDsn`	Change the `izpPrefix.PROCLIB` value to match the PROCLIB chosen during installation.

Optionally, you can edit the following attributes in the `ssid.<ssid-name>.json` file.

Table 42. The `ssid.<ssid-name>.json` file attributes

Attribute name	**Description**
`storageGroup`	• Lists the Db2 Storage Group that is used when creating Db2 DevOps Experience instance objects in this Subsystem. • Replace SYSDEFLT with this Db2 Storage Group name, if needed.
`version`	Change this value to match the Db2 version for this Subsystem.
`installWLM`	Required for DDL generation in Db2 Administration Foundation.
`plan`	Change the default plan name, if needed.
`installSchema`	Db2 DevOps Experience objects created for tool management uses has a schema of `IZPN01`. Change this value, if needed.
`installNamingRule`	Db2 DevOps Experience objects created for tool management uses has a name starting with `IZP1`. Change this value, if needed.
`dsntep2PlanName`	If `DSNTEP2` is bound with a plan name other than `DSNTEP2`, change this value.
`jobcard`	Review this attribute as an acceptable `jobcard` framework for your system.

The installation verification program runs remotely by using Unified Management Server/Db2 DevOps Experience APIs to invoke multiple back-end functions.

3. From the directory where the installation verification program files are located, run the `doeivp.sh` script. This invokes Python 3.0. The script checks whether it is running in a Python 3.0 environment. The script stops if Python 3.0 is not available.
4. You are prompted for the following values:

   - UMS configuration

     The installation verification program gathers the environment details by default. Users can also update details by selecting options. Once the configuration details are set correctly, the values are saved for future use.
   - User ID
   - Password

   The password is not stored after running the script.
5. The installation verification program operates in a `Create` or `Destroy` mode. The script defaults to the `Create` mode if no selection is made or an invalid option is selected.
6. The installation verification program refers to the `doeivp.json` file and presents all parameters required for program completion. Most parameters can be individually modified.
7. If the installation verification program was run previously, the last set of parameters can be loaded, or saved parameter files can be listed. You can select from the saved values. The past `Options` files are stored in the `ivpOptions` subdirectory.
8. When selecting parameters, notice that the changes are marked with an `'*'`. If the previous installation verification program options were loaded, you can return to the default values.
9. You can quit the installation verification program by using the following options:

   - q quits and writes options and report files.
   - `q!` quits without writing the options or report files.

Create or Destroy mode

Procedure

1. After the parameters are set and the process is chosen, the options are verified and the installation verification program starts running.
2. The options are written to an `Options` file in the `ivpOptions` subdirectory.
3. **Create mode**

   In the create mode, the script performs the following tasks:

   a. Confirms that the Subsystem is already registered. If not, the script finds the `ssid.<ssid-name>.json` file and registers the Subsystem.

   b. Queries the Subsystem to confirm they the Db2 Sample DB exists with the Sample DB & CR value options. The `DEPT`, `EMP`, `ACT`, and `PROJ` tables must exist.

   c. Confirms that the environment exists. If the environment does not have a Subsystem assigned to it, the script customizes the `IVPEnv.json` file and creates the environment.

   d. Confirms that the team exists. If not, the script customizes the `IVPTeam.json` file and creates the team.

   e. Validates that the current user is a Team Admin assigned to the team. If not, the script makes the current user a Team Admin.

   f. If the application exists, the script ensures that it contains the Sample DB/CR objects. If it does not exist, the script customizes the `IVPApp.json` file and creates the application. The script also polls for the completion of the application.

   g. Customizes the `IVPInst.json` file and provisions an instance for the application with the installation verification program instance name as a prefix and a random number concatenated to the end of the name. If all objects exist, the create mode attempts to create a new instance. The script polls until the instance creation is complete.

      Note: The installation verification program environment is set to have a limit of five instances. An error is returned if a sixth instance is attempted.

   h. The script creates the sample site rules from the JSON files that are in the `ivpSiteRules-db2` directory.

      - Each `JSON` file is reviewed to determine if the payload represents a simple or a complex rule. The appropriate `POST API` is then used.
      - If a site rule name already exists, it is skipped. For more information on site rules, see site rule descriptions.

   i. A report file of messages is written to the installation verification program directory.

   j. The script verifies if the report files count exceeds the set Report File Limit. If so, the older files are removed.

   k. Review the messages. If no errors are shown, your installation verification program was successful. You can launch the Db2 DevOps Experience UI and view the Db2 DevOps Experience objects.

   l. Errors reported during the create mode may indicate problems with the installation or customization, including the security set up. You might encounter errors that are not related to the installation verification program. For example, there might be errors due to preexisting objects with an incompatible configuration. In such cases, resolve the installation or configuration issue. For details on resolving installation verification program issues, see “Resolving installation verification program issues” on page 102.
4. **Destroy mode**

   In the destroy mode, the script performs the following tasks:

a. If the script finds an installation verification program application, all instances for the named application are deprovisioned. The script polls these instances until they are removed. The application is then deleted.

b. The script deletes the specified team. If that team has other Team Members/Admins assigned, you are prompted to validate team deletion.

c. The script deletes the specified environment. If the environment has other Subsystems assigned to it, you are prompted to validate environment deletion.

d. Subsystems are not removed by using the destroy process. To remove Subsystem registration, use the Db2 DevOps Experience UI.

e. Installation verification program site rules are not removed by using the destroy process. To delete the installation verification program site rules, navigate to the site rules page and filter on `IVP`. This lists rules indicated in the script output. You can multi-select and delete these rules.

f. Error reporting during the Destroy mode may be due to changes made to the Db2 DevOps Experience installation verification program objects since their creation. If the message indicates otherwise, there may be an installation or customization issue. Attempt to diagnose or contact your IBM technical representative to open a support case. The report and option files, from the installation verification program, can be used to isolate and resolve the issue. For more information on resolving installation verification program issues, see "Resolving installation verification program issues" on page 102.

Configuring site rules

About this task

Site rules are specific to Db2 DevOps Experience. You can create a set of samples simple or complex site rules for different scenarios. This process uses the sample `doeivp.json` file containing the default values. If invoked from the installation verification program, all needed variables are passed. If you are configuring site rules independently of the installation verification program, the script queries for the ID and password, and then selects the server and port number from the `doeivp.json` file. The default values can be changed when running independently.

Procedure

1. The script locates `JSON` files in a subdirectory named `ivpSiteRules-db2`. These files contain the `JSON` body for the appropriate `POST` site rule APIs.
2. The script reviews these files to determine if they are simple or complex site rules. The appropriate `POST` API is then constructed and invoked.
3. The script prefixes `IVP` to the site rule name, if not already present. The prefix helps in filtering on the site rules in the UI. A suffix (`SSR` or `CSR`) is also added to the name to indicate if it is a simple or complex rule.
4. If the site rule name already exists, the script skips the name. This method allows site rules to be added to the directory and the process to be rerun. The script only adds new site rules. If a rule is updated in the file, delete it from the Db2 DevOps Experience UI and rerun the installation verification program process or the site rule module.

   - Site rules are not assigned to any applications or environments. You can use these rules as reference-only samples that can be customized as needed.
5. The script returns a summary to the terminal/command-line processor and is recorded in a report file. If this process was a part of the installation verification program, the `site rules` section is added in the report. This output includes a summary count of all site rule files and the status of these rules (created, skipped, or in error).

Table 43. Site Rule descriptions

Site rule name	Description
`IVP Must be UTS (CSR)`	• The example of a complex site rule with precedence. • `maxpartitions` qualification is not necessary to support the rule but adds rule context. • Demonstrates how to code continuation characters and line returns for readability. For example, `'\\\nand'` could be coded as `'\ \\n and'`). • Introduces an important pattern (`segsize` and `segsize` > 0). This tests that the `SEGSIZE` keyword is present and is > 0. **Note:** There is a different pattern for keywords that have acceptable defaults, but also have a rule. Refer to the IVP No DCC (CSR) rule.
`IVP No Implicit Databases (CSR)`	• Verifies that `DBNAME` exists for table creation. • This is a derivative of the `IVP Must be UTS (CSR)` rule pattern to check existence of the database. **Note:** A simple site rule cannot perform this task. As a result, complex site rules are used.
`IVP No Implicit Table Spaces (CSR)`	• Same as the IVP No Implicit Databases (CSR) rule except for `TSNAME`. • Used together, a table must have an explicitly named `DBNAME` and `TSNAME`.
`IVP No DCC (CSR)`	• This is a complex version of the `IVP No DCC (SSR)` rule that uses a ternary expression. • It is a pattern for a keyword where the default value is mandated by the site rule. Currently, a Simple site rule does not support this capability. • This rule checks the value of the keyword if it exists. If the keyword is not provided, the `else` path is used.
`IVP IX must start with X (SSR)`	The following three rules are used together to check an index name. • Follows the Db2 Sample DB standard. • Checks that the index name starts with `'X'`. • Alternately, this rule can be incorporated into a complex site rule by using the **`startswith(“X”)`** command.
`IVP IX name includes TB name (CSR)`	• Follows the Db2 Sample DB standard. • This is a complex site rule that extends the above rule to ensure that the table name follows the `'X'` in the index name. The spaces on either side of the + are required. • Indexing starts with 0, therefore name **`[1:len(tbname) + 1]`** starts its search in the 2nd position and continues for the length of the table name.

Table 43. Site Rule descriptions (continued)

Site rule name	Description
`IVP IX name ends with number (CSR)`	• Follows the Db2 Sample DB standard. • This rule also demonstrates escaping the double-quote, when needed.
`IVP Table Name (CSR)`	• This rule demonstrates an example of a single complex site rule to verify the name. • Does not support the naming standard for Db2 Sample DB tables.
`IVP No AUDIT (SSR)`	• A Simple site rule indicating that `AUDIT` must be `NONE`. • Refer to the IVP No DCC (SSR) rule for considerations with a rule of this type.
`IVP No DCC (SSR)`	• A Simple site rule indicating that `DATACAPTURE` must be `NONE`. • Uses the same format as the IVP No Audit (SSR) rule. • Ensure that the `DATACAPTURE` phrase must appear in the DDL and it must be set to `NONE`. • This rule fails if `DATACAPTURE` is not coded or if it is coded with a value other than `NONE`.

Resolving installation verification program issues

About this task

The goal of this script is to identify installation issues. This section helps to isolate issues with the installation verification program.

If you face issues running the installation verification program, the `doeivp.json` or the `ssid.NAME.json` files might not be configured correctly.

- Refer to the doeivp.json file attributes and ssid.<ssid-name>.json file attributes sections for instructions.
- Verify the parameters shown by the installation verification program script before running the script.
- Adjust the terminal window size for readability, if needed.

If you are using the installation verification program script to register a Subsystem, verify the `ssid.NAME.json` file exists as per the ssid.<ssid-name>.json file attributes instructions and the included values are correct. The name of the second node in the data set must be the `DOE SSID` name. A template `ssid.NAME.json` file is provided, but it must be customized and saved as the new file used for the installation verification program.

While running the installation verification program, perform the following steps:

- Note the error messages that are displayed by the script. This helps in identifying the issue or point to a batch job for additional diagnostics. Use your output viewing tool, such as SDSF.
- Start the Db2 DevOps Experience UI and navigate to the affected component (Subsystems, environments, teams) and locate the tile. The UI might indicate that an error has occurred. Investigate the **Status messages** link, if available. This provides the needed information or point to the job output for diagnosis.
- Notice if the Db2 DevOps Experience component is left in an error state.
  - If the error state is related to a Subsystem, use the UI to remove the malfunctioning Subsystem. Select **Remove Subsystem** from the ellipsis menu of the tile.

- For other components, run the installation verification program script with the same parameters as in the destroy mode.
- You can remove the object definition by using the UI.
- Run the installation verification program script in the create mode.

Post Zowe upgrade tasks for UMS

If you have performed an upgrade for Zowe, perform the following tasks for UMS:

Procedure

1. UMS caches information about the Zowe runtime in `{zowe.components.izp.dataset.hlq}.ENVIRON`. In `IZPGENER`, ensure `zowe.yaml` references the new Zowe runtime libraries, then run `IZPGENER`.
2. If Zowe auxiliary started task `{zowe.setup.security.stcs.aux}` was updated, rerun `IZPSTEPL`.
3. UMS installs Zowe ZSS plugins into `{zowe.setup.dataset.authPluginLib}.ZWESAPL`. If the upgraded version of Zowe is in a different data set than the previous version, rerun `IZPIPLUG`.

Chapter 5. Getting started

To get started with IBM Unified Experience for z/OS, you must register subsystems, and then create environments and teams.

Key concepts

To get started using the UMS graphical component called IBM Unified Experience for z/OS, familiarize yourself with some of the key concepts.

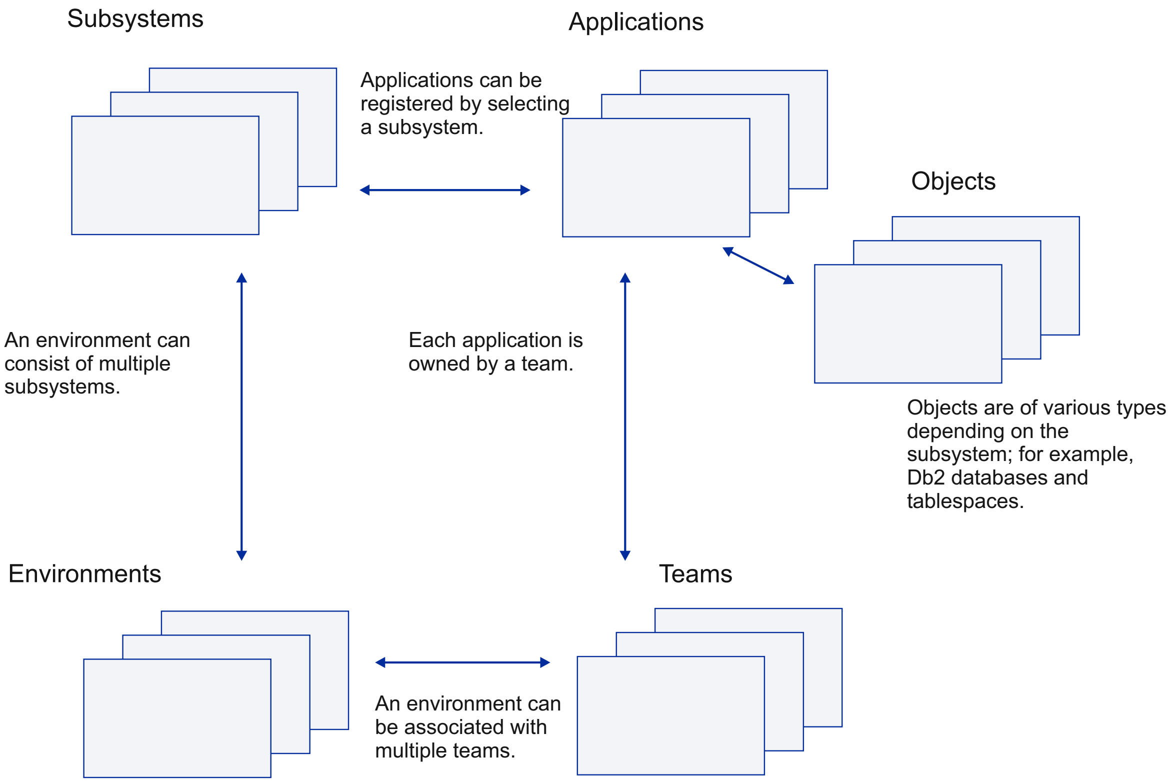

Subsystems
Applications
Applications can be registered by selecting a subsystem.
Objects
An environment can consist of multiple subsystems.
Each application is owned by a team.
Objects are of various types depending on the subsystem; for example, Db2 databases and tablespaces.
Environments
Teams
An environment can be associated with multiple teams.

Figure 6. Basic concepts used in the IBM Unified Experience for z/OS

When you start Unified Management Server, it discovers all of the Db2 and IMS subsystems in your z/OS environment and stores the subsystem information. This process is called *subsystem discovery*.

UMS super administrators (also referred to as *super administrators*) can view the major characteristics of each subsystem, and if a data management product is installed and activated for a subsystem type, they can register some of those subsystems of that subsystem type with IBM Unified Experience for z/OS for later use in the data management product. This is called *subsystem registration*. Only super administrators can register subsystems. To register a subsystem, you select it from the set of discovered subsystems on the **Discovered** tab on the Subsystems page.

A *team* is a group of Unified Management Server users who work together toward a goal such as application development or test environment creation. Each team can be associated with the environments that it can use. Super administrators can create teams and assign environments. If SAF-based security is used, the super administrator should assign permissions to the security administrator to perform team management.

For details, refer to “Setting up users and teams” on page 46.

An *environment* is a collection of subsystems that are used by teams. The super administrator creates environments after subsystems have been registered. After teams have been defined, each environment can be assigned to one or more teams.

If you are using Db2 DevOps Experience, environments are subsystems where members of an associated team can provision application instances. For example, if an application contains Db2 objects, team members must provision the application instance to their team's environment with at least one Db2 subsystem.

In Db2 DevOps Experience, provisioning rules are also associated with each subsystem type that is supported by the environment. For an explanation of provisioning rules, see “Db2 DevOps Experience terms and concepts” on page 156.

Objects are the system resources that are required to run applications. Different subsystem types support different object types; for example, Db2 subsystems support object types such as database, table space, table, and index, and IMS subsystems support object types such as DBD, PSB, and transaction. Objects of a particular subsystem type are defined and modified in a way that is specific to that subsystem type. For example, Db2 objects are defined in the form of DDL statements, and IMS objects are defined in several different forms depending on object types, including assembler macro statements for PSBs and DBDs.

Whenever a new subsystem is registered or when the Unified Management Server is restarted, object discovery is performed. *Object discovery* discovers objects that are defined in the registered subsystems. The discovery status is shown on the **Registered Subsystems** tab on the Subsystems page.

By default, object rediscovery is performed every 4 hours. You can also explicitly initiate object discovery on the registered subsystems by clicking the overflow menu on the subsystem tile and selecting **Refresh object discovery**. The time of the last object discovery is shown on the **Registered subsystems** tab on the **Subsystems** page. Only a single subsystem can be rediscovered at any point in time. All subsequent subsystems are queued for discovery.

Notes:

If you want to increase the `graphql` API timeout limit, perform the following tasks:

1. By using the `<system_admin_id>` user ID, open the `PARMLIB` member `ZWEYAML` and navigate to `components.izp.server.graphQLTimeout` parameter.
2. Specify a new value for the parameter.
3. Restart the UMS server.

The default value is `300` seconds.

An *application* is a logical collection of, or a set of references to, the objects that you create and manage together for the use of an application program or a set of application programs. Those objects can include databases, table spaces, tables, and indices in Db2, and PSBs, DBDs, and definitions for online programs, databases, and transactions in IMS.

A super administrator can register an application and assign it to a team. A team administrator can also register applications for the team.

Note: These conceptual components and associated functions are activated only when the Unified Management Server is used with the Db2 DevOps Experience product.

Roles and responsibilities

In IBM Unified Experience for z/OS, super administrator, team administrator, and team member roles have different responsibilities.

Subsections:

- “Db2 Administration Foundation user roles and responsibilities” on page 107
- “Db2 DevOps Experience user roles and responsibilities” on page 108
- “IMS Administration Foundation user roles and responsibilities” on page 111

The following figure illustrates an overview of user roles and their major responsibilities.

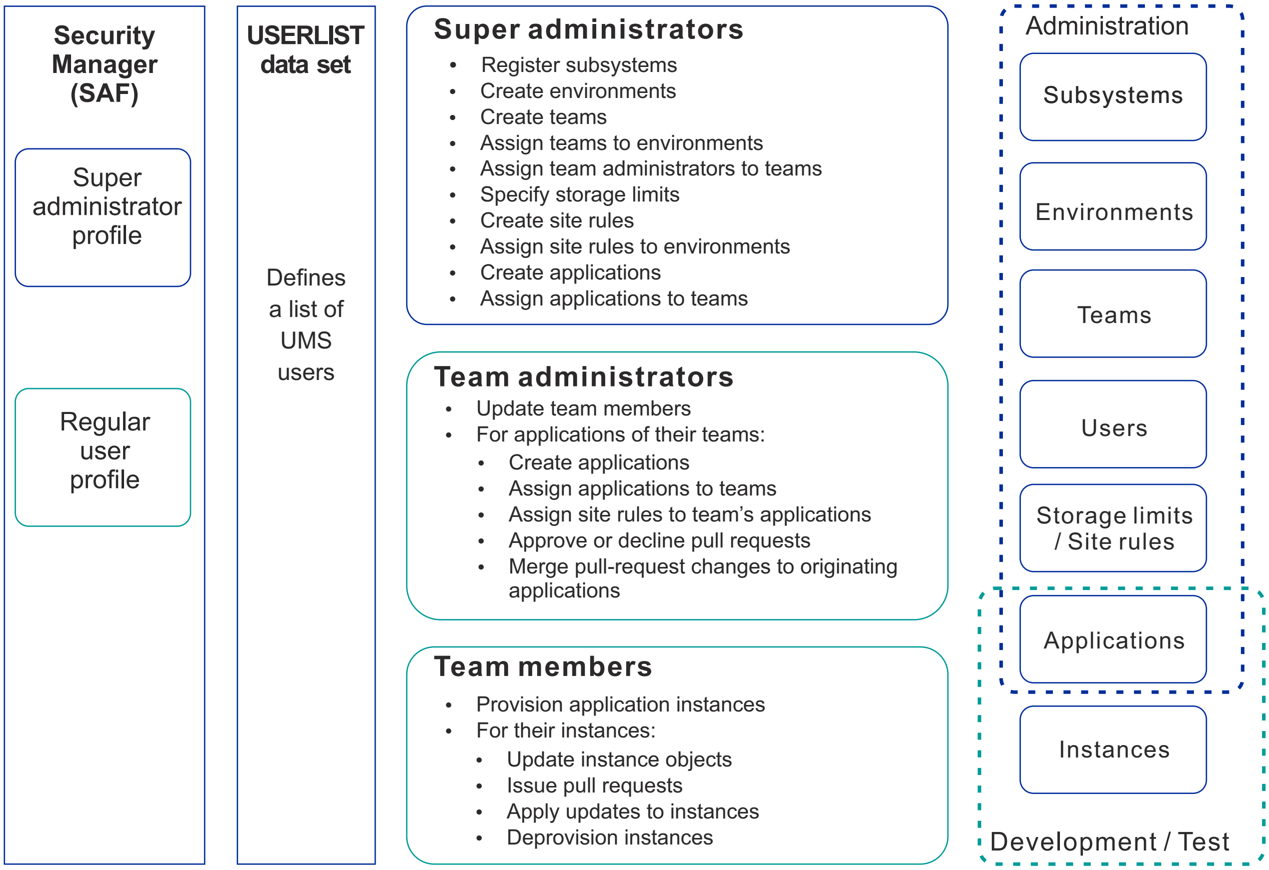

Security Manager (SAF)
Super administrator profile
Regular user profile
USERLIST data set
Defines a list of UMS users
Super administrators
Register subsystems
Create environments
Create teams
Assign teams to environments
Assign team administrators to teams
Specify storage limits
Create site rules
Assign site rules to environments
Create applications
Assign applications to teams
Team administrators
Update team members
For applications of their teams:
Create applications
Assign applications to teams
Assign site rules to team's applications
Approve or decline pull requests
Merge pull-request changes to originating applications
Team members
Provision application instances
For their instances:
Update instance objects
Issue pull requests
Apply updates to instances
Deprovision instances
Administration
Subsystems
Environments
Teams
Users
Storage limits / Site rules
Applications
Instances
Development / Test

Figure 7. User roles and major responsibilities

Note: All of these functions are available only if Unified Management Server is used with the Db2 DevOps Experience product. With only Unified Management Server installed, all you can do is view the subsystems that have been discovered by UMS.

Db2 Administration Foundation user roles and responsibilities

The following table summarizes the user roles and their accessible functions of Unified Management Server running with Db2 Administration Foundation.

Table 44. Db2 Administration Foundation user roles and responsibilities

Category	Task	Super administrator	Team administrator	Team member
Subsystems	View subsystems	Available	Available	Available
	Register subsystems	Available	--	--
	Edit subsystems	Available	--	--
	Remove subsystems	Available	--	--
Users	View users	Available	Available	Available
	Assign users to teams with roles	Available (to all teams)	Available (for their own teams only)	--
	Remove users from teams	Available (for all teams)	Available (for their own teams only)	--

Table 44. Db2 Administration Foundation user roles and responsibilities (continued)

Category	Task	Super administrator	Team administrator	Team member
Objects	Explore objects	Available	Available	Available
	Viewing object details	Available	Available	Available
	Saving a search	Available	Available	Available
	Deleting a search	Available	Available	Available
SQL Processor	Run SQL queries. Authorizations are dependent on user permissions in Db2. You must have permission to run a command in Db2 to run the same command using SQL processor.	Available	Available	Available
Command Processor	Run Db2 commands. Authorizations are dependent on user permissions in Db2. You must have permission to run a command in Db2 to run the same command using Command processor.	Available	Available	Available

Db2 DevOps Experience user roles and responsibilities

The following table summarizes the user roles and their accessible functions of Unified Management Server running with Db2 DevOps Experience.

Table 45. Db2 DevOps Experience user roles and responsibilities

Category	Task	Super administrator	Team administrator	Team member
Subsystems	View subsystems	Available	Available	Available
	Register subsystems	Available	--	--
	Edit subsystems	Available	--	--
	Remove subsystems	Available	--	--
Environments	View environments	Available	Available	Available
	Create environments	Available	--	--
	Edit environments	Available	--	--
	Delete environments	Available	--	--
Teams	View teams	Available	Available	Available
	Create teams	Available	--	--
	Edit teams (assign users and environments)	Available	Available (their own teams only)	--
	Delete teams	Available	--	--

Table 45. Db2 DevOps Experience user roles and responsibilities (continued)

Category	Task	Super administrator	Team administrator	Team member
Users	View users	Available	Available	Available
	Assign users to teams with roles	Available (to all teams)	Available (for their own teams only)	--
	Remove users from teams	Available (for all teams)	Available (for their own teams only)	--
Storage	View by team	Available	Available	Available
	View by environment	Available	Available	Available
	View by user	Available	Available (for non-super administrators only)	Available (for non-super administrators only)
	View by application	Available	Available	Available
	Create storage limit	Available	Available (for their own teams only)	--
Applications	View applications	Available	Available	Available
	View application details	Available	Available	Available
	Register applications	Available	Available (for their own teams)	--
	Edit application settings	Available (can change the owner to any team)	Available (can change the owner to their own teams only)	--
	Delete applications	Available	Available (for their own teams)	--
Site rules	View site rules	Available	Available	Available
	Create site rules	Available	--	--
	Edit site rules	Available	--	--
	Assign site rules to applications	Available	Available (their own teams only)	--
	Assign site rules to environments	Available	--	--
	Delete site rules	Available	--	--

Table 45. Db2 DevOps Experience user roles and responsibilities (continued)

Category	Task	Super administrator	Team administrator	Team member
Instances	View instances	Available (all instances)	Available (all instances)	Available (all instances)
	Provision instances	Available (see notes 5 and 6)	Available (see note 5)	Available (see note 5)
	Change instance owners	Available (see note 4)	Available (see note 4)	Available (see note 4)
	View application details and instance object definitions	Available (all instances)	Available (all instances)	Available (all instances)
	Edit instance object definitions	Available (see note 3)	Available (see note 2)	Available (see note 1)
	Submit pull requests	Available (see note 3)	Available (see note 2)	Available (see note 1)
	Deprovision instances	Available (all instances)	Available (their own team's instances only)	Available (instances that they own)
SQL Processor	Run SQL queries. Authorizations are dependent on user permissions in Db2. You must have permission to run a command in Db2 to run the same command using SQL processor.	Available	Available	Available

Notes:

1. Team members can edit instance objects and submit pull requests only if they are the instance owners, belong to the team that owns the originating application, and the instance was provisioned by that team.
2. Team administrators can edit objects in the instance that a team member has provisioned and submit pull requests only if they are the team administrator of the team that owns the originating application, and the instance was provisioned by that team.
3. Super administrators can edit instance object definitions of their own instances and submit pull requests only if they belong to the team that owns the originating application, and the instance was provisioned by that team.
4. Only the administrator and the members of the team that owns the instance can change the owners of the instance.
5. Only a team that has team members can create an instance.
6. If the super administrator, who creates the instance, is not part of the team under which the instance is being created, the super administrator cannot become the default owner of the instance. Therefore, the super administrator must select at least one instance owner.

The following table summarizes instance-specific roles (regardless of user roles) and their accessible tasks related with pull requests.

Table 46. Pull requests: Tasks and roles

Task	Instance editor	Instance reviewer	Other users
Submit pull requests for instances that you can edit	Available (see note 1)	--	--

Table 46. Pull requests: Tasks and roles (continued)

Task	Instance editor	Instance reviewer	Other users
Add users as additional reviewers	Available	--	--
View all pull requests that you opened	Available	--	--
View all pull requests that you are a reviewer of	--	Available	--
View pull request details	Available	Available	--
Add, edit, and delete comments to any pull requests that you opened	Available	Available	--
View comments from other users	Available	Available	--
Decline pull requests	Available	Available	--
Approve pull requests	--	Available (see note 2)	--
Mark pull requests as "Needs work"	--	Available	--
Merge pull requests	Available	Available	--

Note:

1. Submitting a pull request is allowed only if the instance owner (the team member who provisioned the instance) is a member of the team that owns the originating application.
2. Any reviewers can approve pull requests, but at least one team administrator of the team that owns the originating application of the instance must approve the pull request before it can be merged into the originating application.

IMS Administration Foundation user roles and responsibilities

The following table summarizes the user roles and their accessible functions of Unified Management Server running with IMS Administration Foundation.

Table 47. IMS Administration Foundation user roles and responsibilities

Category	Task	Super administrator	Team administrator	Team member
Subsystems	View subsystems	Available	Available	Available
	Register subsystems	Available	--	--
	Edit subsystems	Available	--	--
	Remove subsystems	Available	--	--

Table 47. IMS Administration Foundation user roles and responsibilities (continued)

Category	Task	Super administrator	Team administrator	Team member
Users	View users	Available	Available	Available
	Assign users to teams with roles	Available (to all teams)	Available (for their own teams only)	--
	Remove users from teams	Available (for all teams)	Available (for their own teams only)	--
Objects	Explore objects	Available	Available	Available
	Viewing object details	Available	Available	Available
	Saving a search	Available	Available	Available
	Deleting a search	Available	Available	Available
IMS SQL Processor	Run SQL queries Authorizations are dependent on user permissions in IMS.	Available	Available	Available
IMS Command Processor	Run IMS Type-1 and Type-2 commands through Operations Manager. Authorizations are dependent on user permissions in IMS. You must have permission to run a command in IMS to run the same command using Command processor.	Available	Available	Available

Managing subsystems

Before you can do anything using the IBM Unified Experience for z/OS, you must install one of data management products and register subsystems to make their objects available. Only super administrators can register subsystems.

The registered subsystems can be used for various different purposes. If you are using Db2 DevOps Experience, you can use the registered subsystems to do the following tasks:

- Create environments and associate them with some of the registered subsystems
- Define groups of subsystem objects as applications
- Provision application instances in a subsystem defined in an environment

To open the **Subsystems** page, click **Manage** and then **Subsystems** on the navigation menu. This page has two tabs: **Registered** and **Discovered**. If subsystems have already been registered, they will be displayed on the **Registered** tab. Before super administrators register any subsystems, this page is empty.

On the **Discovered** tab, you see a list of the subsystems that have been discovered by the Unified Management Server.

If you are not a super administrator: You cannot register subsystems. However, you can view detailed information about the registered subsystems.

Support for Db2 subsystems with encrypted JDBC connections

UMS supports Db2 subsystems with encrypted JDBC connections. UMS discovers the secure ports for any subsystems that support encrypted JDBC connections and will, by default, display and use that port for a subsystem. Users can specify an insecure port if it is preferred for any reason. To allow UMS to register and use a subsystem that has a JDBC connection, the UMS server must have the certificate for the subsystem in its truststore. If the certificate is not imported or SSL/TLS is not correctly configured, the following message is displayed:

```
IZPDB0039E - Could not connect to the subsystem '<subsystem_name>' at url '<hostname>'
with port '<portname>' and location '<location_name>'. Make sure hostname, port number, and
location are correct. For encrypted connections, make sure the certificate for subsystem
'<subsystem_name>' has been imported and SSL/TLS has been correctly configured. (SSL Exception:
'%5$s').
```

To add a certificate to the UMS truststore, run the following commands depending on your truststore:

Using a key ring as a truststore

You must add the certificate authority of Db2 to your key ring. Run the following command for each certificate on a per subsystem basis:

```
RACDCERT CONNECT(CERTAUTH LABEL('<DB2_CA_LABEL>') RING(<RINGNAME>) USAGE(CERTAUTH))
ID(<Zowe_STARTED_TASK_ID>)
SETROPTS RACLIST(DIGTCERT,DIGTRING) REFRESH
```

Using a file-based truststore

For instructions on using a file-based truststore, see "Setting up secure communication for UMS" on page 49.

Registering Db2 subsystems

To register a Db2 subsystem, complete the following steps:

Before you begin

To be able to register Db2 subsystems, you must have at least one Db2 experience properly installed and configured.

Note: To register members of a data sharing group, register the entire group.

Procedure

1. Click the navigation menu and select **Manage**.
2. Click **Subsystems**.
3. Click the **Discovered** tab.
4. Select or search the subsystem that you want to register, and then click **Register subsystem**.
   a) To register multiple subsystems at the same time, select all the subsystems that you want to register, and then click **Register subsystem**.
5. Verify the system details that are displayed on the **System** tab.
   a) If you want to specify another value for the **dbaSqlid** parameter, move the **Subsystem-specific dbaSqlid** slider toward On, and then specify a new value in the **DBA SQL user ID** field.

      Ensure that this ID begins with a letter and has a maximum length of eight characters. To know more about the values that you can specify in the **DBA SQL user ID** field, see Installing Db2 DevOps Experience.

      After the subsystem is registered, this new DBA SQL user ID will be the owner of all the databases and table spaces that are created in this subsystem.

   b) If you want to use an external Db2 security and do not want any Db2 grants issued automatically by the product, move the **Suppress Db2 grants** slider toward `True`. By default, the value is set to `False` for the `defaultDb2GrantSuppression` member in `PARMLIB (IZPDB2PM)`.

c) Optional: If you want to connect to the subsystem by using a specific user ID, specify the details in the **DBA User ID** and **Password** fields. You must also confirm your password once.

   If you do not specify a user ID, the `DBA user ID`, which you set up by completing the steps described in "Item 1: Prepare a DBA user ID and ensure its privileges" on page 54, will be used to connect to the subsystem.

6. Click **Next**.

   The **Configuration** tab is displayed.

7. Enter details on the **Configuration** tab.

   Verify populated fields and make changes as necessary. To copy information of the subsystems that you have already registered, click **Copy configuration**.

8. Click **Register**.

   While registering multiple subsystems at the same time, repeat steps "5" on page 113, "7" on page 114, and "8" on page 114 for each subsystem that you selected.

   Note: If there are errors while registering a subsystem, the multiple registration process continues till all the selected subsystems are registered. However, other validation errors, such as non-existing data sets or inability to connect to a supplied port stop the registration. You must resolve these errors to continue registering the remaining subsystems.

   Important: If you are using spool management or archive system, ensure that the jobs submitted by this product are not immediately removed from the spool. Jobs should remain in the spool for the duration of the operation, which may be several minutes.

Results

After a few moments, the status of the subsystem is displayed on the **Registered** tab on the **Subsystems** page.

The following tips will help you register multiple subsystems:

- If you click **Cancel** while registering a subsystem, that and the subsequent subsystems will not be registered.
- To skip registering the remaining subsystems that you selected, clear the **Register the next subsystem (subsystem name) after this one** check box.

Important: To register a subsystem that is undiscovered, click the **Registered** tab, **Register subsystem**, and then **Define subsystem manually**. To do this, you must have all the system and configuration information of that subsystem, and then complete the steps that follow.

Note: By default, object rediscovery is performed every 4 hours in the **Registered** tab on the **Subsystems** page. You can also explicitly initiate object discovery on the registered subsystems by clicking the overflow menu on the subsystem tile and selecting **Refresh object discovery**.

Registering IMS subsystems as a data sharing group

To work with IMS Administration Foundation, you must register IMS subsystems or data sharing groups. Only UMS super administrators can register IMS subsystems and data sharing groups.

Before you begin

To be able to register IMS subsystems and data sharing groups, you must have IMS Administration Foundation installed and configured properly.

Notes:

- In IMS Administration Foundation, IMS subsystems are registered as IMS data sharing groups and not as individual subsystems. An IMSplex that consists of a single IMS subsystem is also registered as an IMS data sharing group.

- For an IMS data sharing group to be registered for using IMS Administration Foundation, the data sharing group and its associated IMSplex must satisfy a set of prerequisite conditions. For details, see "Software requirements for IMS Administration Foundation" on page 39.

If you are using IBM IMS Administration Tool for z/OS (also referred to as IMS Administration Tool) and you want to register data sharing member IMS systems to IMS Administration Tool along with the registration of the data sharing group to IMS Administration Foundation, you must complete the setup for IMS Administration Tool before starting an IMS data sharing group registration for IMS Administration Foundation. For the setup details, see IMS Administration Foundation and IMS Tools and Installing IMS Administration Foundation.

About this task

UMS automatically discovers IMS data sharing groups, their system components, and their properties. Some of these properties are displayed when you select an IMS data sharing group to register with UMS.

To register an IMS subsystem, you must specify the IMS library information that is not available from the IMSplex components that compose the data sharing group. To use the IMS data sharing group for IMS Administration Foundation, specify the following information:

- A library data set that contains IMS DFSMDA dynamic allocation members for the subject data sharing group. Select a data set from the list of possible data sets that is displayed.

To use the IMS SQL processor or IMS command processor of IMS Administration Foundation, you must select the IMS Connect ports that are to be used for the processors. For each of DRDA and IMS command port types, a list of available IMS Connect servers and their ports is displayed during registration.

- To use the IMS SQL processor feature, you must specify at least one DRDA port of an IMS Connect server that can connect to the data sharing group.
- To use the IMS command processor feature, you must specify at least one transaction port of an IMS Connect server that can connect to the IMSplex to which the data sharing group belongs.

  Note: To use a transaction port for the IMS command, specify the IMS Connect OM Command exit routines HWSCSLO1 and HWSSMPL1 in the **EXIT=** parameter of the TCPIP statement in the IMS Connect configuration member in which the port is defined. See "Software requirements for IMS Administration Foundation" on page 39.

To use any of the following IMS Administration Foundation features, you must install one of prerequisite IMS Tools product and specify a pair of an IMS Tools Knowledge Base (IMS Tools KB) server and a RECON ID:

- Visualizing the database segment structure and segment relationships that are defined in DBDs and PSBs.
- Detecting threshold exceptions for some selected database space statistics and reviewing those exceptions.
- Detecting if a database reorganization or an image copy is needed based on the pre-defined criteria.

If no RECON ID is found in any of IMS Tools KB servers, you can associate the ID after you have created a RECON ID on an IMS Tools KB server. For details on the RECON ID, see "Configuring IMS Tools Knowledge Base and RECON IDs" on page 253. You can create a RECON ID during IMS data sharing group registration if you have installed IMS Administration Tool. For details, see "IMS Administration Foundation and IMS Tools" on page 249.

Tip: It is recommended that you create a set of RECON IDs before you register IMS data sharing groups if you want to use features that require RECON IDs and you do not have IMS Administration Tool installed.

Procedure

To register an IMS data sharing group, complete the following steps:

1. From the navigation menu, select **Manage**.
2. Click **Subsystems**.

3. Click the **Registered** tab to make sure that the IMS data sharing group that you are registering has not been registered yet.
4. Click the **Discovered** tab.

   Note: You can also register an IMS data sharing group from the **Registered** tab. In that case, select a subsystem in the **Select subsystem to register** dialog, and go to step “8” on page 116.
5. Select **IMS** from **Subsystem type** drop-down list.

   In the list to be displayed, you will see all active IMS subsystems, IMSplexes, and IMS data sharing groups that UMS discovered on the sysplex on which the UMS server is running. If multiple data sharing groups are found in an IMSplex, a group name will be created for each data sharing group as the IMSplex name followed by a suffix "GRnn" and the group name will be displayed in the list. If IRLM is used for the data sharing group, its IRLM XCF group name will also be displayed.
6. Find the IMS data sharing group that you want to register.

   You can filter the list by specifying a first few characters of the name in the search field. By default, the data sharing groups that have already been registered are not included in the list. To show them in the list, select **Show registered subsystems**.
7. Select an entry from the list and click **Register subsystem**.

   Alternatively, click **Ready to register** under the **Status** column of the subsystem or data sharing group that you want to register. If a status other than **Ready to register** is displayed, you cannot register that subsystem.

   The **Register subsystem (IMS)** page opens. This page has several steps, and you must confirm the discovered information and fill in all the required fields before you can register the data sharing group.

   If you have installed and setup IMS Administration Tool, which was registered to one of the IMS Tools Knowledge Base servers configured for the subject UMS server, the following additional features will be activated during the IMS data sharing group registration:

   - You can create an IMS Tools RECON ID if one does not exist for the data sharing group. You can rename the RECON ID if it already exists for the data sharing group.
   - You can register IMS subsystems that belong to the data sharing group. These IMS subsystems will be registered to the selected IMS Tools Knowledge Base server by using IMS Administration Tool for future use by various IMS Administration Tool functions.
   - You can register DBDLIBs and PSBLIBs to the selected or created RECON ID for future use by various IMS Administration Tool functions.
8. Verify the IMS data sharing group information that is discovered by UMS.
9. Specify values for all the required fields in each step on the **Register subsystem** page, and then click **Register** to register the data sharing group to UMS.

What to do next

To edit an IMS data sharing group definition, click the overflow menu on the registered subsystem tile for that data sharing group, select **Edit subsystem**, and then click **Update**.

Removing Db2 subsystems

By removing subsystems, they will become unavailable from view and use in the Unified Management Server. Only super administrators can remove subsystems.

Before you begin

To be able to remove a subsystem, it must not be assigned to any environments and the subsystem must be active.

Procedure

To remove a subsystem, complete the following steps:

1. From the navigation menu, select **Manage**.
2. Click **Subsystems**.
3. Click the **Registered** tab.
4. On the subsystem that you want to remove, click the overflow icon (the vertically aligned three-dotted icon on the upper right corner), and select **Remove subsystem**.

Results

The subsystem is removed.

You can forcefully delete a subsystem that is unreachable. To unregister such a subsystem, click the **Unregister** button on the **Remove subsystem** window. You can also remove an unreachable subsystem by specifying **`forceDelete`**`=true` in the `Deregister Db2Subsystem` API. To know how to use the API, see the "Swagger documentation" on page 265.

Removing IMS data sharing groups

By removing an IMS data sharing group, they will become unavailable from view and use in the Unified Management Server. Only super administrators can remove registered IMS data sharing groups.

Before you begin

To be able to remove an IMS data sharing group, it must not be assigned to any environments and at least one IMS subsystem in the data sharing group must be active.

Note: If an IMS data sharing group is associated with an IMS Tools RECON ID, removing the data sharing group does not remove its associated RECON ID from the IMS Tools Knowledge Base repository.

Procedure

To remove an IMS data sharing group, complete the following steps:

1. From the navigation menu, select **Manage**.
2. Click **Subsystems**.
3. Click the **Registered** tab.
4. On the subsystem that you want to remove, click the overflow icon (the vertically aligned three-dotted icon on the upper right corner), and select **Remove subsystem**.

Updating Db2 subsystem registration parameters

Apart from the parameters that are displayed on the **Register subsystem** page, there are other parameters that you can specify while registering a subsystem. Use the `register.JCL` file to modify the existing hardcoded values or specify new values for all the parameters.

Procedure

To modify the subsystem registration parameters, complete the following steps:

1. Open the `register.JCL` file that is located at: `components.izp.workspaceDirectory/conf/jclTemplate/db2` and assign a new value to the parameter that you want to modify.

   For example, you can change the value of the **CHKBPT** parameter from `BUFFERPOOL BP0` to `BP16K0`.
2. Save and close `register.JCL`.
3. Update all the registered subsystems that you registered before updating `register.JCL`.

   The new parameter values are applied to the registered subsystems.

Managing environments

Super administrators can create, edit, or delete environments. An *environment* consists of a set of subsystems that need to be processed as a group. For each environment, you can define a set of rules that are applied when an application instance is provisioned to one of the subsystems in that environment.

To open the **Environments** page, click **Manage** and then **Environments** on the navigation menu. If environments have already been created, they will be displayed on the page. Before super administrators create any environments, this page is empty.

If you are not a super administrator: You cannot create environments. However, you can view detailed information about the environments that have been created. For descriptions of each field that is displayed in the **Environments** page, see "Creating environments" on page 118.

Note: The environment features are experimental for IMS. You can add registered IMS subsystems or data sharing groups and assign them to teams. The environments or teams are not used for IMS functions.

Creating environments

Create and edit environments to group one or more subsystems and to define rules for instance provisioning.

Before you begin

To be able to create an environment, you must have the super administrator role, and the subsystems for that environment must have been registered.

Procedure

To create an environment, complete the following steps:

1. From the navigation menu, select **Manage**.
2. Click **Environments**.

   The **Environment** page opens. This page shows a list of the environments that have been created.
3. Click **Create environment**.

   The **Create environment** page opens.
4. In the **Environment name** field, type the name of the new environment.

   The environment name is case-insensitive; that is, lowercase and uppercase letters are not distinguished.
5. In the **Instance limit** field, click **Set limit**, and specify the maximum number of instances that can be provisioned to this environment.

   To disable the instance limit, click **Remove limit**.
6. Click the type of the subsystem that you want to add to this environment.
7. In the **Add subsystems** page, select subsystems from the list, and click **Add**.

   Important: At least one subsystem must be associated with an environment.

   The subsystems that you added are displayed under the **Subsystems** field in the **Create environment** page.
8. Optional: For each subsystem type that you added to this environment, specify provisioning rules.

   For a description of what provisioning rules are, see "Db2 DevOps Experience terms and concepts" on page 156. If no provisioning rules are specified, default rules will be applied.

   a) Click **Provisioning rules**.

      The **Provisioning rules** dialog for the selected subsystem type opens.

   b) From the **Object type** field, select the object type for which you want to create a rule.

   c) From the **Rule type** field, select a rule type.

   d) In the **Rule** field, enter a rule for the selected object type, and click **Add rule**.

      For a description of the wildcard symbols that you can use, see the tool tip for **Rule type**.

e) To add rules for other object types, repeat steps “8.b” on page 118 through “8.d” on page 118 as necessary.

   You can create a rule for each object type.

f) Click **Update**.

   You will return to the **Create environment** page. The provisioning rules that you specified will be shown in the **Provisioning rules** field.

9. When you have finished adding all subsystems, click **Create**.

   The environment that you just created will be displayed in the **Environments** page.

Editing environments

After an environment is created, super administrators can change the environment settings, to change the environment name, add or remove subsystems, and add, delete, or change rules for instance provisioning.

Before you begin

To be able to edit environment settings, you must have the super administrator role.

Procedure

To create an environment, complete the following steps:

1. From the navigation menu, select **Manage**.
2. Click **Environments**.

   The **Environment** page opens. This page shows a list of the environments that have been created. You can edit or delete existing environments.
3. On the environment that you want to edit, click **Edit**.

   The environment page opens.
4. Change the settings as necessary.

   Environment name
   If you want to change the environment name, overwrite the current name with the new name.

   Instance limit
   If you want to change the instance limit, click **Set limit** and change the maximum number of instances that can be provisioned to this environment. To disable the instance limit, click **Remove limit**.

   Subsystems
   If you want to add subsystems, click the type of the subsystems (for example, Db2) that you want to add to this environment and select subsystems from the list.

   If you want to remove a subsystem from the environment, click **Remove** next to that subsystem name. To be able to remove a subsystem, there should be no instance that is associated with that subsystem. If you try to remove a subsystem that is associated with one or more instances, an error message is displayed.

   Important: At least one subsystem must be associated with an environment.

   Provisioning rules
   If you want to add provisioning rules, click **Provisioning rules** and specify the rules. If you want to change or delete existing provisioning rules, click the **Edit provisioning rules** icon next to the provisioning rule name. For a detailed explanation of how to specify provisioning rules, see “Creating environments” on page 118.
5. When you have finished editing this environment, click **Update**.

Deleting environments

If an environment is no longer needed in IBM Unified Experience for z/OS, super administrators can delete it.

Before you begin

To delete an environment, you must first do the following:

- Remove all the instances that have been provisioned to the environment. To remove those instances, go to the **Instances** page, view all team instances, filter the instances by entering the environment name, and then deprovision each of those instances.
- Remove association to this environment from all teams. You can either edit the teams associated with the environment and remove the environment from each team (see "Editing teams" on page 122) or remove all the teams that are associated with the environment (see "Deleting teams" on page 122).

You can delete an environment even if one or more subsystems are associated with it.

Procedure

1. From the navigation menu, select **Manage**.
2. Click **Environments**.

   The **Environment** page opens. This page shows a list of the environments that have been created.
3. On the environment that you want to delete, click the overflow menu (a vertically aligned three-dotted icon on the upper right corner), and select **Delete environment**.

Managing teams

Super administrators can create, edit, or delete teams. A *team* can consist of team members and team administrators. Team administrators can do various administrative tasks, such as updating team members, managing applications that are owned by the team, and approving pull requests for application instances owned by the team.

For details about what team administrators and team members can do, see "Roles and responsibilities" on page 106.

Important: This section is applicable only when `useSafOnly` is set to `false`. If set to `true`, you can ignore this section.

To open the **Teams** page, click **Manage** and then **Teams** on the navigation menu. If teams have already been created, they will be displayed on the page. Before super administrators create any teams, this page is empty.

If you are not a super administrator: You cannot create teams. However, you can view detailed information about the teams that have been created. For descriptions of each field that is displayed in the **Teams** page, see "Creating teams" on page 120.

Creating teams

Create teams of users who work together on applications. When you create a team, you must give it a name and assign environments to it. Optionally, you can specify a team job prefix to make it easy to identify team jobs that will be submitted by Unified Management Server to manage team's applications and application instances or add users to the team as team administrators and members.

Before you begin

To be able to create teams, you must have the super administrator role.

Procedure

To create a team, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Teams**.
3. In the **Teams** page, click **Create team**.
4. In the **Create team** page, specify the following fields:

   Team name

   Enter the name of the new team. The team name is case-insensitive; that is, lowercase and uppercase letters are not distinguished.

   Job prefix

   Specify a job prefix for this team. All the team jobs that will be submitted by Unified Management Server to manage team's applications and application instances will have job names that start with the specified prefix. This field is optional.

   If the job prefix for the team is not specified, the job name will be generated based on the job prefix that was specified in the Unified Management Server configuration in the PARMLIB data set member ZWEYAML element `components.izp.server.jobPrefix`.

   Team members and administrators

   Important: If `useSafOnly` is set to `true`, this field is disabled. For details, refer to "Managing teams" on page 120.

   Specify team members and team administrators, as follows:

   a. Click **Manage users**.

   b. In the **Manage users** dialog, select IDs from the list. When you select IDs, they will be team members by default. If you want someone to become a team administrator, select **Team administrator** from the **Role** column on that user's entry. A team administrator for a team has all permissions that a team member has for the team.

      Note: Assign at least one administrator to the team. Only a team with an administrator can own an application.

   c. Click **Select**.

   This field is optional.

   Environments

   Optionally, assign one or more environments to this team by clicking **Assign environments**. In the **Assign environments** dialog, select the environments that you want to assign, and click **Assign**. The settings of the selected environment are displayed in the **Create team** page.

   You can optionally limit the number of instances that this team and each user of this team can provision to the environments. Environment limits are the absolute maximum. Team limits cannot be greater than environment limits. User limits cannot be greater than team limits. When no limit is set, the implied limit is the "parent" limit. For example, with an environment limit of 100 and no team limit set, one team can provision a maximum of 100 instances. To set a limit, click **Set limit** and specify a value.
5. Click **Create**.

   In the **Teams** page, you will see the team that you just created.

What to do next

After a team has been created, create an application and make the team its owner. Members of the team can then provision instances of the application to their environment.

Editing teams

Super administrators and team administrators can edit team definitions.

Procedure

To edit a team, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Teams**.
3. In the **Teams** page, click the overflow menu of the team that you want to edit, and click **Edit team**.
4. In the **Edit team** page, change the fields as necessary.

   Important: While updating the users in your team, ensure the team has at least one administrator. You cannot remove the last administrator of a team if the team owns an application. Only a team with an administrator can provision an instance and own an application.

   For details of each field, see "Creating teams" on page 120.
5. Click **Save**.

Deleting teams

Super administrators can delete teams when they are no longer needed.

Before you begin

Ensure the following before deleting a team:

- The team does not own any application instances
- The team does not own any applications

Procedure

To delete a team, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Teams**.
3. In the **Teams** page, click the overflow menu of the team that you want to delete, and select **Delete team**.

   The team disappears from the **Teams** page.

Managing users

Super administrators can assign users to teams or remove them from teams. Team administrators can assign users to, or remove users from, their own teams only. When assigning users to teams, you can specify who will be the team administrators. If the user you want to assign to a team is not in the Users page, you must first add that user to UMS.

For details about what team administrators and team members can do, see "Adding and deleting user records" on page 70.

Important: This section is applicable only when `useSafOnly` is set to `false`. If set to `true`, you can ignore this section.

The **Users** page displays a list of the TSO IDs who can be added to teams. This list shows the number of teams that each user belongs to as well as the names of those teams.

If you are not a super administrator or a team administrator: You cannot assign users to teams. However, you can view information about which users belong to which teams in the **Users** page.

Assigning users to teams

After creating a team, assign users to that team and specify who will be the team administrators.

Procedure

To assign a user to one or more teams, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Users**.
3. In the **Users** page, select an ID from the list and click **Assign teams**.

   To filter the list, enter a search string in the **Search users** field.
4. In the **Assign teams** page, repeat the following steps until you have selected all the teams that you want to assign to this user.
   a) Select the name of the team that you want to assign to this user.

      When you select a team, that user is assigned to the team as a team member by default.

   b) Optional: If you want to assign this user as a team administrator, click **Team administrator** to change the role.
5. Click **Select**.
6. Repeat steps "3" on page 123 through "5" on page 123 to assign teams to other users.

Removing users from teams

Super administrators can remove users from any teams. Team administrators can remove users from their own teams only.

Procedure

To remove a user from one or more teams, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Users**.
3. In the **Users** page, select an ID from the list and click **Assign teams**.

   To filter the list, enter a search string in the **Search users** field.
4. In the **Assign teams** page, clear the check boxes for all the teams that you want to remove from this user.

   Important: You cannot remove the last administrator of a team if the team owns an application.
5. Click **Select**.

Removing users from all teams

Super administrators can remove users from all teams. Team administrators can remove users from their own teams only.

Before you begin

Ensure the following before removing a user from all the teams:

- The user is not the only administrator for that team.
- The user is not the only administrator for the team that owns an application or an instance.

Procedure

To remove a user from all teams, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Users**.

3. In the **Users** page, select an ID from the list and click **Remove from all teams**.

   To filter the list, enter a search string in the **Search users** field.
4. When a confirmation message is displayed, click **Remove**.

   The selected user is removed from all teams.

Chapter 6. Running SQL queries and commands

The SQL processor is a built-in editor that you can use to run and tune SQL queries on the subsystem objects. For example, you can query the tables and views in the registered Db2 subsystems. For Db2, you can also use SQL Tuning Services to receive recommendations for improving the performance of the SQL query.

The command processor is a built-in general-purpose editor that you can use to run commands for Db2 and IMS. For example, you can display all the locks and claimers on a database by running a Db2 command. You can also bind, rebind, and free a plan by running a DSN command for Db2.

Working with SQL queries for Db2

Use the SQL processor to run and tune SQL queries.

Using SQL processor for Db2

You can create and send SQL queries and call procedures directly to databases in the registered subsystems. By using the SQL processor, you can, for example, confirm whether an application instance has been successfully provisioned or the changes to an object in a provisioned application instance have been correctly applied to the instance.

Procedure

To query a database in a subsystem, complete the following steps:

1. From the navigation menu, select **Run**, and then click **SQL**.

   The **SQL** window opens.
2. Select a subsystem from the **Connection** drop-down list.
3. Perform either of the following actions:

   a) To run a new query, on the **New statements** tab, type or paste the query into the editor.

   b) If you want to reuse an SQL statement that you previously submitted, open the **Recent statements** tab, and click the SQL statement that you want to reuse.

   The selected statement is automatically displayed on the **New statements** tab. You can use the SQL statement as it is or change it as necessary.
4. Click **Run**.

   The **Add variable value** window opens with the parameters that require variable values.

   Notes:

   - To move the **Add the required values for the variables** section to the bottom, click ⬓ or to the right, click ◨.
   - When there are multiple queries listed in the **Add variable value** window, use → to move to the next query and ← to move to the previous query.
5. Enter the appropriate variable value in the **value** field or select the **Is null** check box if the value is NULL. The status of the query on the right-corner changes to **Visited**.

   Note: If the required value is not entered, the status remains as **Not visited** and the **Run** button will be disabled.
6. Click **Run**.

What to do next

You can tune the SQL statements or export the query results. See Tune SQL queries or "Exporting SQL queries " on page 128.

Using SQL Tuning Services for Db2

SQL Tuning Services provides a comprehensive set of tools for analyzing and tuning Db2 for z/OS SQL statements as well as the features that are required to set up a tuning environment. SQL Tuning Services includes Visual Explain, which allows you to tune a query under the current execution path. It includes Statistics Advisor, which provides recommendations for improving the performance of a query based on RUNSTATS output. And it includes the Capture Query Environment function, which is useful for re-creating an environment on a different Db2 subsystem or to help IBM Support understand problems that you are experiencing with an SQL query. All of these features provide the capabilities you need to improve query performance and reduce execution time and system resource usage. is an efficient tool for improving the performance of an SQL query that is not performing at a measurable standard. By tuning an SQL query, you can improve its response time and reduce the resources that are needed to run the query.

Before you begin

Configure the UMS for using SQL Tuning Services by completing the steps that are described in "Configuring UMS for SQL Tuning Services for Db2" on page 86.

About this task

To tune an SQL query, you must specify the tuning profile, the tuning action that you want to use while tuning the SQL query, and the tuning parameters.

Procedure

To tune an SQL query, complete the following steps:

1. From the navigation menu, select **Run**, and then click **SQL**.

   The **SQL** window opens.
2. Select a subsystem from the **Connection** drop-down list.
3. Perform either of the following actions:
   a) To tune a new query, on the **New statements** tab, type or paste the query into the editor.
   b) If you want to reuse an SQL statement that you previously submitted, open the **Recent statements** tab, and click the SQL statement that you want to reuse.

      The selected statement is automatically displayed on the **New statements** tab. You can use the SQL statement as it is or change it as necessary.

   Note: You can tune only one SQL query at a time.
4. After selecting or typing the SQL query that you want to tune, click **Tune**.

   The **Query tuning** window opens.
5. Select the tuning profile that you want to use for tuning the query, and then click **Next**.

   Note: The **Select tuning profile** drop-down lists only the profiles that are shared with you or that are available in the registered subsystem that you selected in step "2" on page 126.
6. On the **Select tuning actions** tab, select the tuning action, and then click **Next**.

   The **Parameters** tab is displayed. This tab shows the parameters that are needed to tune the SQL query.
7. To change the default values of the parameters, on the **Parameters** tab, move the **Re-explain** slider toward **On**.

   The default value of the **Re-explain** slider is **On**.
   a) Specify the job name in the **Job name** field.

      You can change the default job name that is displayed in this field.

b) Specify values for the parameters that are listed on the **General** and the **Parameters** tabs.

   Note: The fields display the current values of the parameters. If you do not specify a value for the parameters, the default values are used.

8. To tune the query, click **Next**.

   The **Output** tab displays the job name, the tuning action, and the status of the job that was submitted for tuning the SQL query.
9. To view the query tuning results, click **View results**.
10. Optional: To capture the details of the environment in which the query was tuned, click **Capture query environments** on the **View results** page.

What to do next

To see the status of all the SQL query tuning jobs, from the navigation menu, select **View jobs**, and then click **Tuning**.

Working with SQL queries for IMS

You can create and send IMS SQL queries to IMS databases by selecting an IMS ODBM datastore alias registered for an IMS data sharing group and specifying a PSB that includes PCBs for the databases.

About this task

The SQL processor is a built-in editor that the user can use to run the following types of IMS SQL data management statements against IMS PSBs:

- SELECT statement
- INSERT statement
- UPDATE statement
- DELETE statement

Data definition statements CREATE, ALTER, and DROP are not supported by the IMS SQL processor.

For considerations when you write SQL queries against IMS databases, see the 'Writing SQL queries to access an IMS database with the IMS Universal JDBC driver' topic in *IMS Application Programming* guide.

Note: All the SQL statements in a **Run** session will be committed automatically if all of the statements succeeded. If there was any statement that resulted in an error, all the SQL statements for the **Run** session will be rolled back.

Procedure

To query IMS databases in an IMS subsystem, complete the following steps:

1. From the navigation menu, select **Run** and then click **SQL**.
2. Select the subsystem type of IMS.
3. Select an IMS data sharing group name and a connection.
4. Specify an IMS PSB that is defined and activated on the selected IMS subsystem.
5. Do either of the following:
   a. In the **New statements** tab, type or paste your query into the editor.
   b. If you want to reuse a SQL statement that you previously submitted, go to the **Recent statements** tab, and click the one that you want to reuse. The selected statement will be automatically entered in the **New statements** tab. You can use it as it is or change it as necessary.
6. Click **Run**.

   The results of the query will be displayed in the lower area of the page.

What to do next

You can export the query results. See “Exporting SQL queries ” on page 128.

Exporting SQL queries

You can export the results of an SQL query from the **SQL Processor** page.

Before you begin

Before exporting SQL queries, see “Using SQL processor for Db2” on page 125 or “Working with SQL queries for IMS” on page 127.

Procedure

To export the results of an SQL query, complete the following steps:

1. Run one or more SQL statements by using the appropriate SQL processor.
   a) For Db2, see “Using SQL processor for Db2” on page 125.
   b) For IMS, see “Working with SQL queries for IMS” on page 127.
2. Click ⇨.
3. Select a separator from the **Select CSV separators** drop-down list.
4. If more than one query was run, then select a result set to export from the **Result sets** drop-down.
5. Click **Export**.

Running Db2 and DSN commands

You can create and run the Db2 and DSN commands in the subsystems.

Procedure

To query a database in a subsystem, complete the following steps:

1. From the navigation menu, select **Run**, and then click **Commands**.

   The **Commands** page opens.
2. Select a subsystem.
3. Type the command in the Db2 command input field and click **Run**.

   The results are displayed in the lower area of the page. The results of all the commands that you run are displayed one below the other with the latest results displayed at the top. If you want to see the results of only the current command, click **Clear and run** after you enter and type the Db2 command.

Running IMS commands

You can input and run IMS type-1 and type-2 commands for IMSplexes.

About this task

At least one IMS data sharing group of the target IMSplex must be registered to the Unified Management Server and the Operations Manager must be set up and used for the IMSplex.

Procedure

To query a database in a subsystem, complete the following steps:

1. From the navigation menu, select **Run**, and then click **Commands**.

   The **Commands** page opens.
2. Select **IMS** from the subsystem type list on the top right.

The view is switched to the IMS command processor view.

3. Type the command in the **IMS command** input field.
4. Select and **IMSplex**.

   A list of valid command targets is displayed in the Route list dropdown. The default route is * (Route all).
5. Select a list of route targets and click **Run**.

   Each command result is displayed in a separate tab in the lower area of the page. The result of a type-1 command is displayed as text lines. The result of a type-2 command is displayed as a table.

Chapter 7. Using Db2 Administration Foundation

Db2 Administration Foundation is a separate product that you can install and activate on top of Unified Management Server. It has its own set of features and functionality that you can use to perform tasks for z/OS subsystems.

Important: To use Db2 Administration Foundation features on your subsystems, you must have Db2 Administration Foundation installed and configured properly. If it is not yet installed, see "Installing Db2 Administration Foundation" on page 74.

For explanations of some of the basic concepts and key terms used in this product, see Chapter 5, "Getting started," on page 105.

Benefits of Db2 Administration Foundation

IBM Db2 Administration Foundation for z/OS leverages the Open Source Zowe Virtual Desktop platform of IBM Unified Experience to provide a rich set of capabilities for administering and working with your Db2 for z/OS environment.

Db2 Administration Foundation delivers the following key capabilities in a modern browser-based interface:

- An intuitive and easy-to-use graphical user interface for exploring and querying the Db2 catalog. This interface includes a series of dashboards that provide high-level information and details about Db2 objects, the relationships between objects, and the data associated with objects. The interface is designed to make it easy for both new and experienced users to work with the Db2 catalog.
- An SQL editor that allows you to edit, save, run, and tune SQL statements, and create workloads against registered Db2 subsystems. Separate tabs are provided for SQL statements that you have recently worked with and SQL statements that have been analyzed.
- The ability to capture SQL from a variety of sources including the statement cache, plans, packages, stabilized SQL, and input files, and to subsequently explore the SQL that you capture.
- A command interface from which you can run Db2 and DSN commands.
- The ability to analyze the performance of and tune SQL statements that access Db2 for z/OS data. Some of the tuning capabilities are available at no charge. A more robust set of tuning capabilities is available when you purchase Query Workload Tuner for z/OS 6.1, which enables you to increase SQL performance by refining the use of SQL syntax and formatting, indexes, access paths, and statistics for both individual SQL statements and SQL statements that comprise a workload. Adding Query Workload Tuner 6.1 to your Db2 Administration Foundation enables you to significantly increase the efficiency and performance of Db2 for z/OS SQL applications.

  See Overview of IBM SQL Tuning Services for details about the no-charge tuning features and the additional tuning features that are available with Query Workload Tuner 6.1.
- The ability to work with accelerators and accelerated queries, which is provided by the Db2 Analytics Accelerator Administration Services component of IBM Db2 Analytics Accelerator for z/OS. A series of dashboards provides you with an overview of your accelerator environment and allows you to update tables, view accelerated queries, and monitor accelerators based on various characteristics including system utilization, data slices, current tasks, finished tasks, and incremental updates.

Exploring Db2 objects

The **Explore objects** page provides a unified platform for searching objects across multiple subsystems. You can use this page as a dashboard to navigate to the detailed pages of the objects.

Procedure

1. On the navigation menu, click **Explore** and then **Objects**.

The **Explore objects** page opens.

2. Specify the search criteria, such as the object name, schema, object type, and subsystem.

   Note: You can specify some or all criteria. The object name and the schema can include alphanumeric (A-Z, a-z, 0-9) characters, special characters, and (spaces). The following wild card characters are supported: `! ? % * _`

   For example:

   - If you specify **Name matches** `!ABC`, the results will exclude all the objects that begin with ABC.
   - If you specify **Name matches** `A9_A`, the results will include all the objects that begin with A9 and end with A, such as A91A, A9CA, and so on.
   - If you specify **Name starts with** `A9%`, the results will include all the objects that contain A9, such as A91A, A91234, and so on.

   Important: You can use `!` to exclude objects only when you use it with the search criterion **matches**.

   Note:

   - To see the details of a DSG member, specify the member name in the **Name** field, select `Subsystem` from the **Object type** drop-down list, and then click the member name that is displayed in the result table. For example, if `dsg_member` is a DSG member, then to see the details of `dsg_member`, specify `dsg_member` in the **Name** field, select `Subsystem` from the **Object type** drop-down list, and then click `dsg_member` that is listed in the result table.

     To see the names of the DSG members click **No. of members** on the DSG subsystem card on the **Registered** tab of the **Subsystems** page.
   - To load tables to accelerators, select `Table` from the **Object type** drop-down list and a Db2 subsystem. For more information, see Loading tables to accelerator.

     The check box to select the tables from the result list is displayed only if Db2 Analytics Accelerator Loader is integrated with Db2 Administration Foundation. See Configuring Db2 Analytics Accelerator Loader.
   - To view the template of Db2 command, select any of the following in the **Object type** drop-down list: `Database`, `Function (Db2)`, `Index`, `Stored procedure`, `Subsystem (Db2)`, `Table`, and `Table space`. For more information refer to “Running Db2 commands using template” on page 134.

3. To see the search result, click **Apply**.

   Note: To save this search for using it later, click **Save as**. To view the list of searches that have been previously saved, click **Load search**.

Results

The objects meeting the search criteria are displayed in the result table. If the search results take more than 300 seconds to be displayed on the **Explore objects** page, cancel this search and try again.

Important: If object discovery for the subsystem is not yet complete, inconsistent values might be displayed in the result table. For example, `unknown` or `-1`.

Note: By default, the **Explore objects** page performs object rediscovery every 4 hours.

What to do next

View object details by clicking the object that is listed in the result table.

Loading a search

If you save the search criteria that you frequently use, you can load the results faster. The **Load search** feature displays a list of all the saved searches. You can load any search from this list and also share it with other Db2 Administration Foundation users in your organization.

Procedure

1. Click **Load search**.

   The **Load search** dialog opens and displays a list of the saved searches.
2. Select the search you want to load, and then click **Load and apply**.

   The search result is displayed in a tabular format on the **Explore objects** page.

   - To only review the search criteria without loading the results, click **Load**.
   - To delete a search that you previously saved, click **Load search**, select the search you want to delete, and then click the **Delete** icon. You can access but cannot delete search filters created by other users.

Results

The **Explore objects** page displays the search criteria and its results.

Updating a search

You can update an existing search criteria to improve your search results. The **Update** feature lets you load and update an existing search criteria. You can provide an updated description if needed.

Procedure

1. Click **Load search**.

   The **Load search** dialog opens and displays a list of the saved searches.
2. Select the search you want to load, and then click **Load and apply**.

   The search result is displayed in a tabular format on the **Explore objects** page.
3. Update search filters as needed.
4. Click **Save**.
5. Specify an updated description for this search filter if needed.
6. Click **Update**.

Results

The **Explore objects** page displays the search filter successfully updated notification.

Loading tables to accelerators

You can load tables to accelerators and include advanced options. The advanced options selection provides additional keywords to your load. The **Load data** page supports only **Table** object type, and objects that are from the same Db2 subsystem.

Procedure

1. Select the check box of the required tables displayed in the result table.

   Note: The check box for **Table** object type is displayed only when Db2 Analytics Accelerator Loader is configured, see Configuring Db2 Analytics Accelerator Loader.
2. On the right corner of the result table, click **Take action** and then **Load tables to accelerators**.

Note: If you are performing this for a single object in the list, click ⋮ and then **Load table to accelerators**.

Important: If multiple object types other than `Table` are selected, a warning message will be displayed with the option to either cancel the operation or continue. If you click **Continue**, object types other than Table will be ignored while performing the Load table to accelerators action.

3. Select the required field values and check boxes for the following:

   Name
   Select the check box of accelerators to which the tables should be loaded.

   Enable acceleration
   Controls whether query acceleration is enabled for the table after a successful load.

   Detect data changes
   Indicates whether to load only those tables and partitions that have changed in Db2 since the last load.

   Add table to accelerator
   Indicates whether to add missing tables to the accelerator before starting the load job. Valid values are

   - **Add** - Add missing tables. This value generates the control card ACCEL_ADD_TABLES into the JCL.
   - **Remove and add** - Add missing tables; remove and re-add existing tables. This value generates the control card ACCEL_REMOVE_AND_ADD_TABLES into the JCL.

   Table lock mode
   Specifies the protection level while tables on an accelerator are being loaded. Valid values are

   - None - No locking at all.
   - Table - Protects just the table that is currently being loaded.
   - Tableset - Protects all tables to be loaded against changes during the load operation.
   - Partitions - Protects the table space partition containing that part of the table that is currently being loaded.
   - Row - Protects just the row or page that is being loaded against updates.

   Load tasks
   Specify the number of parallel load tasks to use. Valid values are 1 to 30.
4. Click **Submit** to load the tables to the specified accelerators attached to the selected Db2 subsystem.

Results

A message is displayed that the job has been submitted successfully along with the job name and job ID.

What to do next

Copy the job name and job ID that are displayed to check the status of the load job. Validate the job output to confirm that the accelerator is loaded successfully.

Running Db2 commands using template

You can view or edit the template of the Db2 command for one or more objects by using the **Run Db2 command** feature.

Procedure

1. Select the check box of the required objects displayed in the result table.

Note: The **Take action** drop-down is displayed on the upper-right corner of the result table only when the check box of the **Object type**, such as `Database`, `Function (Db2)`, `Index`, `Stored procedure`, `Subsystem (Db2)`, `Table`, and `Table space` are selected.

2. On the right corner of the result table, click **Take action** and then **Run Db2 command**.

   Note: If you are performing this for a single object in the list, click ⋮ and then **Run Db2 command**.

   Important: If any object types other than those specified in step 1 are selected, then they will be ignored while performing the **Run Db2 command** action.

3. Select the required field values for the following:

   Select command
   Select the command and the valid values are `Display`, `Start`, `Stop`, and `Access`.

   Select command details
   Select the details for the command selected in the **Select command** field.

   Show template
   Toggle the button to view the template in edit mode. To edit the template, click **Edit** and make the required changes.

   Notes:

   - To compare the changes that you made to the existing template, toggle the **Diff view** to `On`.
   - To apply the changes made to the template, click **Apply changes**.
   - To revert the changes made to the template, click **Revert changes**.
   - To copy the template, click ⧉.

   Important: You can toggle the button to **Hide template** to move from edit to view template mode.

4. Click **Run**.

Results

The **Commands** page is displayed with the result of the selected command and the changes that were implemented.

What to do next

Copy the command using the ⧉ for future reference.

Running DSN commands using template

You can view or edit the template of the DSN command for one or more objects by using the **Run DSN command** feature.

Procedure

1. Select the check box of the required objects displayed in the result table.

   Note: The **Take action** drop-down is displayed on the upper-right corner of the result table only when the check box of the **Object type**, such as `Package(Db2)`, `Stored procedure`, and `Trigger(Db2)` are selected.

2. On the right corner of the result table, click **Take action** and then **Run DSN command**.

   Note: If you are performing this for a single object in the list, click ⋮ and then **Run DSN command**.

   Important: If any object types other than those specified in step 1 are selected, then they will be ignored while performing the **Run DSN command** action.

3. Select the required field values for the following:

Select command
Select the command and the valid values are `Free`, `Bind`, and `Rebind`.

Select command details
Select the details for the command selected in the **Select command** field.

Show template
Toggle the button to view the template in edit mode. To edit the template, click **Edit** and make the required changes.

Notes:

- To compare the changes that you made to the existing template, toggle the **Diff view** to `On`.
- To apply the changes made to the template, click **Apply changes**.
- To revert the changes made to the template, click **Revert changes**.
- To copy the template, click ⧉.

Important: You can toggle the button to **Hide template** to move from edit to view template mode.

4. Click **Run**.

Results

The **Commands** page is displayed with the result of the selected command and the changes that were implemented.

What to do next

Copy the command using the ⧉ for future reference.

Working with objects

The **Object details** page provides detailed information of the objects. You can use this information to monitor the status of these objects.

You can drill down to the object details by using the following options:

- Click the name of the object listed in the search result on the **Explore objects** page. The **Object details** page opens. This page contains various tabs that display related information for that object.
- Click the **Object hierarchy** button. The **Relationship** tab on the **Object details** page opens and displays the relationship between objects. To see more objects in the relationship hierarchy, expand the caret and click the object names. The relevant tab opens and displays detailed information.

You can customize the object details by using the **Filter** icon.

Notes:

- To see the different versions of a package, select a version from the **Version-Bind time** drop-down list that is available on the **Object details** page of a package.
- To perform an action on a Db2 accelerator, select the required action from the **Accelerator actions** drop-down list that is available on the **Object details** page of a Db2 accelerator.

  If you want to submit a request to collect system performance data, perform the following:

  1. Select the **Collect system performance data** option from the **Accelerator action** drop-down.
  2. Select the **Select to provide the timeout** check box to provide the timeout in minutes to capture the system data.
  3. Select the **Select to provide the top queries** check box to provide the number of queries that should be collected for each category.
  4. Click **Submit**. The request is submitted successfully.

The following table describes the tabs that are available on the **Object details** page:

Table 48. Tabs on the Object details page

Tab	Description
Overview	Summarizes the important details of the object. This tab displays the following tiles: Key information, Object hierarchy, Structure, and Creation details. To open the relevant tab, click **Details** or click the tile. To start or stop members of the Db2 accelerators that are part of a data sharing group, click **Manage Members** on the **Key information** tile. To start or stop paired Db2 accelerators that are part of a subsystem, click the subsystem on the **Object hierarchy** tile. On the **Paired accelerators** tile, do the following: • To start the accelerator, click ▶ and then click **Submit**. • To stop the accelerator, click ■ and then click **Submit**.
Structure	Shows information of the associated objects, such as tables, table spaces, plans, and so on in a tabular format. You can customize the displayed information by sorting and filtering the data. Select an associated object from the **Select object detail** drop-down list. The detail of that associated object is displayed in a tabular format.
DDL	Generates the data definition of the selected object. You can copy this definition and use for other tasks, such as running SQL commands. Only for Db2 Administration Foundation, you can edit and run the DDL statement of the object in the SQL processor by clicking the **Run in SQL processor** button. **Note:** • The DDL of an object is generated only if the WLM environment is configured for the subsystem. To configure the WLM environment, complete the steps described in Configuring WLM for Db2 Administration Foundation. • The DDL of implicit objects is not generated.
Bind	Binds the plans and packages with all the required properties. You can bind, rebind, and free plans and packages. But you can only rebind the packages that have the type F, N, T, and 1. You can update the statement of the package or the plan by clicking **Edit**. Before applying or reverting your changes, compare them with the current information by using the **Diff view** slider: • To see your changes, as you edit, immediately below the current information, click the **Unified** icon. • To see the current statement and your changes in a split view, click the **Side by side** icon. The left pane of the split view displays the current information. The right pane is editable, and the differences are shown as you edit the statement.

Table 48. Tabs on the Object details page (continued)

Tab	Description
Relationship	Displays the relationship of the selected object with other associated objects, including the subsystem of the object. • To see the differed objects that are associated with the selected object, expand the subsystem. • To see details of any object, click the name of the object. The **star** icon next to the object indicates that you are viewing the **Relationship** tab of that object. **Important:** • Only 10 children are displayed in the object's hierarchy. To see all the children of the object, click the **Structure** tab. • For the table object with type T, the **Referential integrity** tile is displayed next to the **Object hierarchy** tile. Use the drop-down list to see the constraints associated with the selected table in the form of parents, children, and siblings.
Storage	Displays the storage details of the selected index or table space of a subsystem, such as current and maximum size, partition number, and storage graph. Additionally, it shows the status of the objects as follows: • Needs Action - Above 90% • Needs Attention - Between 75% to 90% • Doing Good - Below 75% • Total **Note:** You can modify these values if required. For details on enabling the Storage tab, see "Configuring Storage tab" on page 76.
Statistics	Displays the statistics data of the selected index or table space of a subsystem, which is fetched from the catalog table (SYSTABLESPACESTATS). It shows the status of the objects as follows: • Needs action • Needs attention • Doing well • Total The statistics data of the object is represented in percentage under the following categories: • Image Copy - Updated Pages • Reorg - Data Modification • Reorg – Space (specific to tablespace) or Reorg - Insert Append (specific to index) • Reorg – Disorganization • Runstats - Data Modification Additionally, you can view the statistics graph and log for each category. For details on enabling the Statistics tab, see "Configuring Statistics tab" on page 76.

Table 48. Tabs on the Object details page (continued)

Tab	Description
Status	Displays the following information about the locks and claims on the selected object: • The type of lock on the object • The user who has claimed this object • Users who have requested a claim for this object To see the related information, select **Claims** or **Locks** from the drop-down list.
Data	Displays the data that is available inside a table or a view.
SQL	Displays the list of SQL statements that are available for a package. The following functions are available for the SQL statements: **Copy** To copy the SQL statement, click the **Copy to clipboard** icon. For example, you can run this SQL statement by pasting it in the editor of the SQL processor. **Tune** To tune the SQL query, click the overflow menu, and then click **Tune**. The **Query tuning** window opens. To tune the query, complete the steps that are described in "Using SQL Tuning Services for Db2" on page 126. **View SQL query** To view the full SQL query, click the overflow menu, and then click **View SQL query**. The **New statements** tab opens. You can edit, copy, tune, and run the SQL statement.
Monitor	Displays the health and performance of a Db2 accelerator on different tabs. • To see a visual representation of the resources that are being used by the Db2 accelerator, click the **System utilizations** tab. • To see the space that a subsystem uses on the top 2 out of 5 full and empty database partitions, click the **Data slices** tab. • To see all the tasks that are allocated to the Db2 accelerator, click the **Current tasks** tab. • To see a list of all the tasks that the Db2 accelerator has completed, click the **Finished tasks** tab. • To see the graphical representation of incremental updates, click the **Incremental updates** tab.
Tables	Displays the information about the tables that are being accelerated. You can see information, such as the schema name, status of the acceleration, the time when the table was last loaded, and so on. You can customize the search results by using the **Column selector**. • To add a table to the Db2 accelerator, click **Add**. • To perform an action on a table, click the overflow menu, and then click the action that you want to complete.

Table 48. Tabs on the Object details page (continued)

Tab	Description
Queries	Displays the information about the queries that are being accelerated. You can also filter the queries by specifying the filtering criteria. You can customize the search results by using the **Column selector** and the **Column filter**. • To perform an action on a query, click the overflow menu, and then click the action that you want to complete. **Note:** To save the filtering and ordering criteria for using them later, click **Save as**. To view the list of searches that have been previously saved, click **Load search**.

Exploring SQL

You can use Db2 Administration Foundation to capture SQL statements on demand from the Db2 dynamic statement cache, packages, and plans to stabilize, tune, and free stabilized SQL statements.

Before you begin

- Ensure that SQL Tuning Services is installed and configured, that the SQL Tuning Services repository database has been created, and that the SQL Tuning Services user ID that you are using has permission to access the repository database. SQL Tuning Services is available in the following two products:

  IBM Database Services Expansion Pack

  This offering is available at no additional cost to licensed IBM Db2 Accessories Suite for z/OS users. It includes a subset of the available SQL Tuning Services features including Visual Explain, Statistics Advisor, and Capture Query Environment.

  IBM Db2 Query Workload Tuner for z/OS

  This offering can be purchased separately or is included in the IBM Db2 Performance Solution Pack for z/OS. It includes all of the features that are included with IBM Database Services Expansion Pack plus a more robust set of SQL analysis and tuning features.
- Ensure that a valid tuning profile has been created.
- Ensure that EXPLAIN tables have been created.

About this task

This task illustrates the general process for using the SQL exploration features in Db2 Administration Foundation.

Procedure

1. From the navigation menu, click **Explore** > **SQL**.
2. From the upper-right corner of the **Explore SQL** page, select a tuning profile and the source that you want to capture SQL from.
   - Statement cache
   - Plan
   - Package
   - Stabilized SQL
   - Input file
3. Specify the SQLID, which is the current user’s login ID that is authorized for this process.
4. Optional: Specify the number of results to display in the output. The default number is 100.

5. Optional: Specify filtering criteria and ordering criteria.

   Selections that you make are displayed in the **Specified criteria** section of the window. To remove a criteria, click the **X** on the right side of the criteria.
6. Click **Capture**. The results are displayed in a table at the bottom of the page.

   To search columns for specific values, use the search field at the top of the result table. To customize the results, use the column selector icon on the right side of the search field.
7. Click the three vertical dots at the end of the query's row to see the actions that are available for that query:

   - **View SQL query** opens the query in the SQL window. You can edit, run, tune, and create a workload with the statement from here.
   - **Tune** opens the **Query tuning** dialog. From here, you can initiate tuning actions for the statement.
   - **Capture Query Environment** captures details about the environment in which the current SQL statement is running or finished running.
   - **Create workload** enables you to create a new workload by selecting one or more SQL statements.
   - **Add to workload** enables you to add one or more SQL statements to an existing workload.
   - **Stabilize** opens the **Stabilization group selection** dialog. Specify a group name and click **Stabilize** to stabilize the statement. After the stabilization process finishes, close the pop-up and click **Capture** again to refresh the results and verify that the STABILIZED column contains a **Y** for this statement.
   - **Free stabilized** frees a statement that has been stabilized.

     Note that the **Stabilize** and **Free stabilized** actions are available only for statements that were captured from the statement cache.

Capturing SQL from a z/OS data set

Db2 Administration Foundation makes it easy to identify and display SQL statements that exist in a z/OS sequential data set or partitioned data set member and then work with those statements by using other Db2 Administration Foundation features.

Procedure

1. From the **Explore** menu, select **SQL** to display the **Explore SQL** page.
2. From the **Select source** menu on the right side of the SQL page, select **z/OS data set**.
3. Enter a data set name for a sequential data set or a data set name and member name for a partitioned data set, and a statement delimiter, and select the profile that you want to use. Optionally specify the additional criteria that are available on this page.
4. Click **Capture**. The results are displayed in the bottom of the window. You can modify the columns that are displayed in the results by clicking the column selector on the right side of the results pane.

What to do next

From the results pane, you can invoke Db2 Administration Foundation features for an individual SQL statement or for a collection of SQL statements.

- To work with an individual SQL statement, click the three vertical dots at the end of the statement row. The following actions are available:
  - View SQL query
  - Tune
  - Create workload
  - Add to workload
  - Capture query environment

- To work with a collection of SQL statements, select the check boxes for the statements that you want to work with, then click the **Take actions** menu. The following actions are available for a collection of statements:
  - View SQL query
  - Create workload
  - Add to workload

Creating and managing tuning profiles for Db2

A tuning profile contains information that is relevant for connecting to a subsystem, such as the subsystem name, hostname, and port. A tuning profile is required to use the SQL tuning capabilities of Db2 Administration Foundation

Procedure

Before you create a tuning profile, make sure that the following conditions have been met:

- Ensure that you have configured UMS for SQL Tuning Services by completing the steps in Configuring UMS for SQL Tuning Services.
- Ensure that the required SQL Tuning Services user IDs have been created with the appropriate permissions. For more information, see Setting up required user IDs and permissions.

Creating tuning profiles

To create a tuning profile, complete the following steps:

Procedure

1. From the navigation menu, select **Manage**, and then click **Profiles**.

   The **Profiles** window opens. The **Tuning profiles** tab displays a list of available tuning profiles.
2. On the **Tuning profiles** tab, click **Create**.

   The **Create tuning profile** window opens.
3. Specify a name for the profile in the **Profile name** field and select a subsystem from the **Registered subsystems** drop-down list.

   Important: A subsystem is listed in the **Registered subsystems** drop-down list only if it has been successfully registered.
4. Click **Create**.

   The tuning profile is created and listed on the **Profiles** page.

Viewing tuning profiles

Before tuning an SQL query, you must select the tuning profile that is appropriate for the SQL query. You can select the appropriate tuning profile by viewing its details.

Procedure

1. From the navigation menu, select **Manage**, and then click **Profiles**.

   The **Profiles** window opens. The **Tuning profiles** tab displays a list of available tuning profiles.
2. Select the tuning profile whose details you want to see, click the overflow menu, and then select **View details**.

   The **View tuning profile** window opens and displays details of the tuning profile.

Sharing tuning profiles

By using a shared tuning profile, multiple SQL queries can be tuned by using the same tuning profile. Owners of the profiles and users with administrator access to the SQL Tuning Services server can share the tuning profiles.

Before you begin

Ensure that you have the permissions that are required for sharing a profile. For information about the different user roles, see the Setting up required user IDs and permissions table.

Procedure

To share a tuning profile, complete the following steps:

1. From the navigation menu, select **Manage**, and then click **Profiles**.

   The **Profiles** window opens. The **Tuning profiles** tab displays a list of available tuning profiles.
2. Select the tuning profile that you want to share, and then click **Share**.

   The **Share tuning profile** window opens.
3. Select the users with whom you want to share this profile, and then click **Share**.

   The tuning profile is shared with the other users.

Deleting tuning profiles

You can delete a tuning profile if you no longer need it. Owners of the profiles and the users with administrator access to the SQL Tuning Services server can share the tuning profiles.

Before you begin

Before you delete a tuning profile, ensure that the following conditions apply:

- The tuning profile is not shared with other users.
- An SQL query tuning job that is using this profile is not active.

Procedure

1. From the navigation menu, select **Manage**, and then click **Profiles**.

   The **Profiles** window opens. The **Tuning profiles** tab displays a list of available profiles.
2. Select the profile that you want to delete, and then click **Delete**.

   The **Tuning profiles** tab is refreshed, and it displays the latest list of available profiles.

Tuning individual SQL statements

This topic describes how to use the tuning features that can be integrated into Db2 Administration Foundation to tune a single SQL query.

Before you begin

- Ensure that SQL Tuning Services is installed and configured, that the SQL Tuning Services repository database has been created, and that the SQL Tuning Services user ID that you are using has permission to access the repository database. SQL Tuning Services is available in the following two products:

  IBM Database Services Expansion Pack

  This offering is available at no additional cost to licensed IBM Db2 Accessories Suite for z/OS users. It includes a subset of the available SQL Tuning Services features including Visual Explain, Statistics Advisor, and Capture Query Environment.

IBM Db2 Query Workload Tuner for z/OS
This offering can be purchased separately or is included in the IBM Db2 Performance Solution Pack for z/OS. It includes all the features that are included with IBM Database Services Expansion Pack plus a more robust set of SQL analysis and tuning features.

- Ensure that a valid tuning profile has been created.
- Ensure that EXPLAIN tables have been created.

About this task

This task illustrates the general process for using the tuning features in Db2 Administration Foundation. It does not provide details for running each individual tuning feature because the process is similar for all tuning features. This task focuses on Visual Explain, Statistics Advisor, and Capture Query Environment.

Procedure

1. From the navigation menu, click **Run** > **SQL**.
2. From the SQL page, select a subsystem from the drop-down menu.
3. Paste an SQL statement into the **New statements** tab, or type a simple SELECT statement:

   ```
   SELECT * FROM SYSIBM.SYSTABLES;
   ```

4. Click **Tune** to open the **Query tuning** page, and specify the following information:

   a. Select a tuning profile, then click **Next**. Only the profiles that are owned by you or that have been shared with you and that are available in the registered subsystem that you selected in step 2 are available.

   b. Select the tuning actions that you want to run. The actions that are available depend on which SQL Tuning Services offering you are using: IBM Database Services Expansion Pack or Db2 Query Workload Tuner.

      For the purposes of this task, select **Statistics Advisor** and **Visual Explain**, then click **Next**.

   c. Specify tuning parameters, then click **Next** to run the tuning actions.

   The tuning jobs that you selected are displayed in the **Output** tab. Each Job ID is assigned a corresponding job name that ends with a suffix that indicates the type of tuning action (SA for Statistics Advisor, VE for Visual Explain, and so on). When the jobs finish running, you can view the results.

5. Click **View results** for the Visual Explain job to display the graphical representation of the access paths used by the SQL statement. From the Visual Explain results, you can display additional information about the SQL statement by using the actions in the upper-left corner of the Visual Explain output:

   a. Click **SQL Statement** to review the SQL that has been explained.

   b. Click **Warnings** to display any warnings associated with the statement.

   c. Click **Environment & Explain Options** to display additional details about the SQL statement.

6. Click **Related tuning actions** in the upper right corner of the Visual Explain results, then click **View results** for the Statistics Advisor job to display recommendations for collecting additional statistics.
7. Click **Capture query environment** to start a job that collects relevant information about the query and saves it in a downloadable file. This information is a useful resource to provide to IBM Software Support if you need help diagnosing a problem with an SQL statement.

What to do next

- To further familiarize yourself with how to tune SQL statements, you can repeat the steps in this task for the other SQL tuning features that are available to you. If you have a valid Db2 Query Workload Tuner license, you can explore the enhanced tuning features such as Index Advisor, Query Rewrite Advisor, SQL Annotator, and so on.

- To see the status of all the SQL query tuning jobs, from the navigation menu, select **View jobs**, and then click **Tuning**.

Creating and managing SQL workloads

Db2 Administration Foundation provides two options for creating SQL workloads and for managing them after they've been created.

Procedure

Use either of the following methods to create a new workload:

Option 1: Manually entering SQL statements

Use this option to specify the SQL statements for a workload by typing or pasting them into the Db2 Administration Foundation interface.

a. From the UMS navigation menu, click **Run** > **SQL**.

b. Type at least two SQL statements in the **New statements** tab, then click **Create workload**. Note that an SQL workload can consist of a single SQL statement, but for the purposes of this task, we are using the following two SELECT statements:

```
SELECT * FROM SYSIBM.SYSTABLES;
SELECT * FROM SYSIBM.SYSPACKAGES;
```

c. In the **Create workload** dialog that is displayed, assign a name for the workload and select a connection profile from the drop-down menu. Then click **Submit**.

The workload is created and is displayed in the **Workload manager** dashboard along with the SQL statements that comprise that particular workload and information about the statements. From the **Workload manager** dashboard, you can interact with the SQL statements individually by clicking the three vertical dots at the right side of each row, or you can select the **SQL_TEXT** check box to interact with the statements at the workload level.

Option 2: Capturing SQL

Use this method to capture the SQL statements that you want to include in a workload.

a. From the UMS navigation menu, click **Explore > SQL**.

b. Select a connection profile and a source from the **Select source** drop-down menu. You can capture SQL from the following sources:

- Statement cache
- Plan
- Package
- Stabilized SQL
- Input file
- z/OS data set

c. Click **Capture**. The results are displayed at the bottom of the page.

d. Select the queries that you want to include in the workload, then either click the three vertical dots at the end of a row or expand the **Take actions** drop-down menu and select **Create workload**.

e. In the **Create workload** dialog that is displayed, assign a name for the workload and select a connection profile from the drop-down menu. Then click **Submit**.

The workload is created and is displayed in the **Workload manager** dashboard.

What to do next

- To add an SQL statement to an existing workload, select a workload, click the three vertical dots at the end of that statement’s row, then select **Add to workload**. You’ll be prompted to specify the workload that you want to add the statement to.

- To delete an SQL statement from an existing workload, display that workload in the **Workload manager** dashboard, select the statement that you want to delete, then click **Delete**.

Tuning SQL workloads

This topic describes how to use the tuning features that can be integrated into Db2 Administration Foundation to tune the SQL statements that comprise an SQL workload.

Before you begin

- Ensure that IBM Db2 Query Workload Tuner for z/OS is installed and configured. The workload-level tuning features require a Db2 Query Workload Tuner license.
- Ensure that the SQL Tuning Services repository database has been created and that the SQL Tuning Services user ID that you are using has permission to access the repository database.
- Ensure that a valid tuning profile has been created.
- Ensure that EXPLAIN tables have been created.

About this task

This task illustrates the general process for using the SQL workload tuning features in Db2 Administration Foundation. It does not provide detailed instructions for running every possible workload tuning scenario but does demonstrate the frequently used workload tuning features that are available in the Db2 Administration Foundation user interface.

Procedure

1. From the navigation menu, click **Run** > **SQL**.
2. Type at least two SQL statements in the **New statements** tab, then click **Create workload**.

   Note that an SQL workload can consist of a single SQL statement, but for the purposes of this task the workload should consist of at least two SELECT statements; for example:

   ```
   SELECT * FROM SYSIBM.SYSTABLES;
   SELECT * FROM SYSIBM.SYSPACKAGES;
   ```

3. In the **Create workload** dialog that is displayed, assign a name for the workload and select a connection profile from the drop-down menu. Then click **Submit**.

   The workload is created and is displayed in the **Workload manager** dashboard along with the SQL statements that comprise that particular workload and information about the statements. From the **Workload manager** dashboard, you can interact with the SQL statements individually by clicking the three vertical dots at the right side of each row, or you can select the **SQL_TEXT** check box to interact with the statements at the workload level.
4. Now, click **Workload manager** in the breadcrumb links at the upper-left corner of the dashboard (`Dashboard / SQL / `**`Workload manager`**` /`*`workload_name`*) and select a connection profile when prompted. After you select a connection profile, all the workloads that have been created are displayed.
5. Select the workload that you just created, click the three vertical dots at the end of the row, and select **Tune workload**. The **Query workload tuning** window opens.
6. Select the tuning actions that you want to run:
   - Workload access path advisor
   - Workload index advisor
   - Workload query rewrite advisor
   - Workload statistics advisor
7. Click **Next** and specify the parameters for the tuning actions.

8. Click **Next** again to run the tuning actions. The results are disputed in the **Output** section of the window.
9. Click the right end of each row to view the details for each tuning job.
10. To display a list of all the tuning jobs that have been run, click **View jobs** > **Tuning** from the UMS navigation menu. The **Tuning jobs** window opens.
11. To compare the access paths that are being used by two workloads, select two jobs that have the job type of **Workload Explain**, then click **Compare access paths**. The access paths are compared, and the results are displayed in the **Workload manager** dashboard. Click **View details** to display a graphical representation of the comparison.

Evaluating index recommendations before deploying them

You can use the Virtual Index Advisor and Workload Virtual Index Advisor features of Db2 Administration Foundation to virtually apply the recommendations that are provided by Index Advisor and Workload Index Advisor, respectively, so that you can evaluate the projected benefits before you apply the recommendations in your actual Db2 environment.

Before you begin

Make sure that a completed index advisor job is available for the statement or statements that you are working with:

- To virtually apply the indexes for a single SQL statement requires a completed Index Advisor job for that statement.
- To virtually apply the indexes for a workload requires a completed Workload Index Advisor job for that workload.

The completed index advisor job must have at least one candidate index to apply.

About this task

By using Virtual Index Advisor and Workload Virtual Index Advisor, you can virtually test indexes to determine if the performance of a single SQL statement or the SQL statements that comprise a workload can be improved by creating or dropping the indexes that are recommended by the respective index advisors.

The ability to apply these changes virtually before applying them to your system enables you to iteratively evaluate and adjust the indexes that are used in a single query or in a workload to achieve the highest possible performance.

Procedure

1. Select the appropriate completed index advisor job from the **Tuning jobs** page:

   - If you are working with a single query, select a completed Index Advisor job.
   - If you are working with a workload, select a completed Workload Index Advisor job.

   The **Result** page opens for the job that you selected.

   Note: The remainder of this task focuses on working with Workload Virtual Index Advisor; however, the steps for using Virtual Index Advisor for a single query are nearly identical.
2. Click **Take actions** > **Virtual index advisor** to open the Workload virtual index advisor window.
3. Evaluate the indexes that are displayed in the **Recommendations** tab and **Existing indexes** tab, and select the CREATE and DROP actions that you want to include in the evaluation.

   Click the **Selections** tab to see a cumulative view of your selections.
4. Optional: Edit any of the indexes that are displayed on the **Recommendations** tab by clicking the three vertical dots at the end of the row for that index and click **Edit index**.

5. Optional: Add any additional indexes that you want to apply virtually by clicking **Add index** and specifying the characteristics for the new index.
6. When you are ready to apply the changes virtually, click **Run Workload Virtual Index Advisor**. The job is displayed in the **Tuning jobs** window. When it completes, select its job ID to see the projected effects of applying the changes which are displayed in the **Result** page.

   The following information is provided for both the indexes to be created and indexes to be dropped:

   - The estimated change in performance expressed as a percentage
   - The estimated disk space used
   - The number of new indexes that are used in the plan
   - The number of new indexes that are not used in the plan (available only for workloads)
   - The number of dropped indexes
7. If the projected results are acceptable, you can apply the index recommendations to your Db2 environment by using either of these methods:

   - Copy individual index DDL statements from the **Result** page and then click **Run** > **SQL** to open the SQL processor. Paste the DDL in the **New statements** tab and click **Run**.
   - Run all of the DDL statements at once by clicking **Run script** on the **Result** page, which automatically opens the SQL processor and copies all the DDL statements into the **New statements** tab. Click **Run** to submit them.

Analyzing the effects of applying index changes

Use the Index Impact Analysis feature of Db2 Administration Foundation to generate a report that shows the impact of applying recommended index changes on a single SQL statement or on SQL statements in a workload.

Before you begin

Make sure that a completed index advisor job is available for the type of analysis that you want to do:

- Analyzing the impact of index changes for a single SQL statement requires a completed Index Advisor job.
- Analyzing the impact of index changes for the SQL statement in a workload requires a completed Workload Index Advisor job.

The completed index advisor job must have at least one candidate index to evaluate.

Procedure

1. Select the appropriate completed index advisor job from the **Tuning jobs** page:

   - If you are working with a single query, select a completed Index Advisor job.
   - If you are working with a workload, select a completed Workload Index Advisor job.

   The **Result** page opens for the job that you selected.
2. Click **Take actions** > **Index impact analysis** to open the Index Impact Analysis window for the type that you're working with.
3. Select the scope for the analysis:

   - Select **Analyze impact on statements in the following source** to analyze statements in the dynamic statement cache, statements in static bound packages, or both by selecting the appropriate check boxes. You can also specify whether the analysis is run on all packages or only on the most recent packages by using the **Package scope** dropdown menu.
   - Select **Analyze impact on statements in one or more explained workloads** to analyze the impact on statements in one or more workloads. If you select this option, you must select at least one workload that is displayed in the **Workload name** column.

4. Click **Submit** to generate and run an Index Impact Analysis job. The job is displayed in the **Tuning jobs** window. When it completes, select its job ID to see the results, which are displayed in two tabs:
   - The **Impacted packages** tab shows which packages are affected by applying the index recommendations, the specific impact, and the reason for the impact.
   - The **Impacted statements** tab shows the impacted statement for a single query or the impacted statements for a workload, the estimated performance gain, and the schema qualifier.
5. After you review the results of the analysis, you can apply the index recommendations to your Db2 environment by using either of these methods:
   - Copy individual index DDL statements from the **Result** page and then click **Run** > **SQL** to open the SQL processor. Paste the DDL in the **New statements** tab and click **Run**.
   - Run all of the DDL statements at once by clicking **Run script** on the **Result** page, which automatically opens the SQL processor and copies all the DDL statements into the **New statements** tab. Click **Run** to submit them.

Capturing a query's environment

You can capture the details about the environment in which you are running an SQL query and save these details to a file that you can provide to IBM Support when you are trying to resolve a performance problem with an SQL query or that you can use to re-create an environment on another subsystem. You can capture the environment for a single query or for the queries that comprise an SQL workload.

Before you begin

Before you can capture a query's environment, you must first run the query.

About this task

This task provides instructions for capturing the environment for a single SQL query and for the SQL queries that comprise an SQL workload.

Capturing the environment for a single query

Procedure

1. Use either of the following options to invoke Capture query environment for a single query:
   - From the **Tuning jobs** window, click the three vertical dots at the end of a query and select **Capture query environment**.
   - Click **View results** from a Visual Explain job, and click **Capture query environment**.

   After you submit the job, the **Capture query environment jobs** window is displayed.
2. Click **Download** to download the file that contains the details about the environment for the query.

Capturing the environment for the queries in a workload

Procedure

1. Use either of the following options to invoke Capture query environment from the **Workload manager** dashboard:
   - Click the three vertical dots at the end of a workload's row and select **Capture query environment**.
   - Select a workload by clicking its check box, and then click the **Take actions** menu and select **Capture query environment**. You can select only one workload at a time.

   After you submit the job, the **Capture query environment jobs** window is displayed.
2. Click **Download** to download the file that contains the details about the environment for the queries in the workload.

Refining SQL workloads

You can reduce the number of SQL statements that comprise an existing SQL workload by using the Workload Refine feature of Db2 Administration Foundation so that the refined workload results in better query performance.

Procedure

1. From the **Workload Manager** page, select the workload that you want to refine either by clicking the three vertical dots at the end of the row for that workload or by selecting the workload and clicking **Refine** from the action bar.

   You can select only one workload at a time. The workload that you select must be in EXPLAINED status.

   The **Workload Refine** page is displayed.
2. Specify a new name for the child workload if the default name is not acceptable, and use the dropdown menus to specify the appropriate filtering criteria based on your needs.

   The original workload that the child workload is based on remains unchanged.
3. After you specify the criteria that you want to apply, click **Refine**.

   If all of the criteria are satisfied, the SQL statements that comprise the refined child workload are displayed.

Chapter 8. Using IMS Administration Foundation

IMS Administration Foundation is a separate feature that you can activate on top of the Unified Management Server by installing the IBM IMS Tools Base product. IMS Administration Foundation has its own set of functionalities that you can use to perform tasks for IMS.

Important: To use IMS Administration Foundation, you must have IBM IMS Tools Base for z/OS 1.7 installed by SMP/E and post-SMP/E installation and configuration for IMS Administration Foundation have been made properly for the Unified Management Server.

For explanations of some of the basic concepts and key terms used in this product, see Chapter 5, “Getting started,” on page 105.

Benefits of IMS Administration Foundation

IMS Administration Foundation is a browser-based graphical user interface, leveraging modern IMS features and the new Open Source Zowe Virtual Desktop platform. The following basic capabilities are available in IMS Administration Foundation:

- Graphical query and exploration of IMSplexes, IMS subsystems, and IMS data sharing groups with intuitive experience for new users.
- A set of pages that show status and configuration details of an IMSplex, an IMS subsystem, an IMS data sharing group, and an IMS Connect server.
- A set of pages that show a list of online system resources of an IMSplex and a set of pages that show a list of online system resources of an IMSplex and an IMS subsystem and tab-based details page of status and properties of each resource.
- A set of pages that show a list of DBDs and a list of PSBs of an IMS data sharing group and an overview of a DBD or a PSB.
- Click-and-shoot relationship navigation of IMS online resources, DBDs, PSBs, and DBRC-defined groups.
- Basic IMS SQL editor to allow for SQL statements to be issued against IMS ODBM aliases.
- IMS OM command interface for IMS type-1 and type-2 commands equipped with command history.
- Object search for DBD, PSB, DB, PGM, TRAN, RTC, and group.

To use these features, target IMSplexes need to be configured with an Operations Manager, a Resource Manager, one or more CSL-enabled IMS Connect servers with appropriate ports, and an IMS catalog.

When appropriate IBM IMS Tools are installed and the Unified Management Server is configured correctly, some or all of the following extended functions are available:

- Associating an IMS data sharing group registration with an IMS Tools Base RECON ID environment.
- A set of tab-based detail pages of a DBD or a PSB of an IMS data sharing group.
- A graphical segment tree view for a DBD or each DB PCB in a PSB of an IMS data sharing group.
- A summarized page of exceptions.
- Database space usage statistics and exceptions pages for a non-partitioned full-function database, a DEDB area, or a HALDB partition.
- A reports page for a non-partitioned full-function database, a DEDB area, or a HALDB partition.

For details of the dependencies of IMS Administration Foundation user interface features on IBM IMS Tools products, see “User interface features and IBM IMS Tools” on page 152.

For details of IBM IMS Tools products that can be used with IMS Administration Foundation, see “IMS Administration Foundation and IMS Tools” on page 249.

User interface features and IBM IMS Tools

To understand IBM IMS Tools and IMS Administration Foundation functions, refer to the following table:

For the requirements of Available features, see "Software requirements for IMS Administration Foundation" on page 39 and "IMS Administration Foundation and IMS Tools" on page 249.

Table 49. IBM IMS Tools and IMS Administration Foundation functions

User interface feature	No additional IMS Tools	Tools Base TCP server and IMS Tools KB server		
		Without any Solution Pack	With DB or FP Solution Pack, or DB Utility Solution	With Recovery Solution Pack
Dashboard	Available	Available	Available	Available
IMSplex	Available	Available	Available	Available
IMS data sharing group	Available	Available	Available	Available
IMS	Available	Available	Available	Available
IMS Connect	Available	Available	Available	Available
IMS ODBM	Available	Available	Available	Available
List of DBs, PGMs, TRANs, and RTCs and individual details	Available	Available	Available	Available
List of DBDs, PSBs, Groups, and individual details	Available	Available	Available	Available
DBD reports	Not available	Available	Available	Available
DBD and PSB maps	Not available	Available with standalone LIU license	Available	Available with standalone LIU license
DBD statistics	Not available	Not available	DB stats available	Not available
DBD and CA group exceptions	Not available	Not available	Reorg alerts available	Recovery alerts available
Explore objects	Available	Available	Available	Available
IMS SQL Processor	Available	Available	Available	Available
IMS Command Processor	Available	Available	Available	Available

Exploring IMS objects

The **Explore objects** page provides a unified platform for searching objects across multiple subsystems. It provides direct navigation to the individual IMS object page from a search result.

Procedure

1. On the IMS administration dashboard, click **Explore objects**. Alternatively, click **Explore** and then **Objects** on the navigation menu, then select **IMS** from the list of subsystem types on the top right of the page.

   The **Explore objects** page opens.

2. Specify the search criteria, such as the object name, object type, or subsystem.

   Note: You can specify some or all criteria. The object name can include alphanumeric (A-Z, a-z, 0-9) characters, special characters, and (spaces). The following wild card characters are supported: `? % * _`

   For example:

   - If you specify **Name matches** A9_A, the results will include all the objects that begin with A9 and end with A, such as A91A, A9CA, and so on.
   - If you specify **Name starts with** A9%, the results will include all the objects that contain A9, such as A91A, A91234, and so on.

   Note: If you don't specify any object types or any subsystem, all object types or all subsystems will be searched.
3. To see the search result, click **Apply**.

   Note: To save this search for using it later, click **Save as**. To view the list of searches that have been previously saved, click **Load search**.

Results

The objects meeting the search criteria are displayed in the result table.

What to do next

- View details of a resource or an object by clicking the object that is listed in the result table. You will see the search results in the tree view on the left after you have clicked the resource or the object page.
- View details of the IMS or IMS data sharing group to which a resource or an object belongs.
- "Loading a search" on page 153

  If you save the search criteria that you frequently use, you can load the results faster. The Load search feature displays a list of all the saved searches. You can load any search from this list and also share it with other IMS Administration Foundation users in your organization.
- "Updating a search" on page 154

  You can update an existing search criteria to improve your search results. The Update feature lets you load and update an existing search criteria. You can provide an updated description if needed.

Loading a search

If you save the search criteria that you frequently use, you can load the results faster. The **Load search** feature displays a list of all the saved searches. You can load any search from this list and also share it with other IMS Administration Foundation users in your organization.

Procedure

1. Click **Load search**.

   The **Load search** dialog opens and displays a list of the saved searches.
2. Select the search you want to load, and then click **Load and apply**.

   The search result is displayed in a tabular format on the **Explore objects** page.

   - To only review the search criteria without loading the results, click **Load**.
   - To delete a search that you previously saved, click **Load search**, select the search you want to delete, and then click the **Delete** icon. You can access but cannot delete search filters created by other users.

Results

The **Explore objects** page displays the search criteria and its results.

Updating a search

You can update an existing search criteria to improve your search results. The **Update** feature lets you load and update an existing search criteria. You can provide an updated description if needed.

Procedure

1. Click **Load search**.

   The **Load search** dialog opens and displays a list of the saved searches.
2. Select the search you want to load, and then click **Load and apply**.

   The search result is displayed in a tabular format on the **Explore objects** page.
3. Update search filters as needed.
4. Click **Save**.
5. Specify an updated description for this search filter if needed.
6. Click **Update**.

Results

The **Explore objects** page displays the search filter updated successfully notification.

Chapter 9. Using Db2 DevOps Experience

You can use Db2 DevOps Experience as an application object management tool that can be built into the entire DevOps pipeline as the application test environment deployment pipeline through the REST APIs that are provided as part of Db2 DevOps Experience.

Important: To be able to use Db2 DevOps Experience features on your subsystems, you must have a specific DevOps Experience product installed and configured properly. If it is not yet installed, see “Installing data management experiences” on page 83.

Restriction: The current version of Db2 DevOps Experience does not support the following object types:

- Simple table spaces
- Index-controlled partitioning if the Db2 subsystem parameter **`PREVENT_NEW_IXCTRL_PART`** is set to YES.

For explanations of some of the basic concepts and key terms used in this product, see also Chapter 5, “Getting started,” on page 105.

Benefits of Db2 DevOps Experience

Db2 DevOps Experience can address the needs of various users' roles.

You might be facing different challenges depending on your role:

- For application developers, turnaround time to get test environments is always long. Sometimes their requests are misunderstood, and rework is required. Test environment setup is not self-service.
- System and database administrators often receive various types of application test requests from developers. They often find themselves unsure of what needs to be configured and set up to prepare the systems to run their tests.
- IT operators sometimes find resources not properly allocated compared to the resource plan. It takes a lot of time and effort to configure the security settings for those newly defined resources.
- CIOs might find that their organizational agility and efficiency are decreasing due to too much manual effort to set up application testing environments. They face many challenges from other executives in their organizations to move the applications off of the mainframe.

Db2 DevOps Experience, together with IBM Unified Management Server for z/OS, can solve your pain points, in following ways:

- Each application developer can self-provision a copy of their application into their own isolated testing environment, make changes, test, and check in the changes, with minimal to no interaction with administrators. They can deprovision the provisioned objects when they are no longer needed.
- System and database administrators can review, approve, and promote developer’s object changes with confidence. They can also set rules and standards and have oversight to changes if necessary.
- IT operators can manage resource allocation and security control with minimal effort. They can configure and account for the amount of system resources required to support application development.
- CIOs can empower their development teams and shrink their development schedules by providing self-provisioning capabilities, leading to better organizational agility and efficiency.

Db2 DevOps Experience terms and concepts

Before starting to use Db2 DevOps Experience, get familiarized with the terms and concepts used with Db2 DevOps Experience.

applications

An *application* is a logical collection of, or a set of references to, the objects that you create and manage together for the use of an application program or a set of application programs. Those objects can include databases, table spaces, tables, and indices in Db2, and PSBs, DBDs, and definitions for online programs, databases, and transactions in IMS.

Note: In the current releases of Db2 DevOps Experience, an application is associated with a specific subsystem.

application instances

A set of application objects that has been provisioned into the associated environment's subsystem.

When an instance has pending upstream changes to be pulled into an instance from an application, then that instance is not at the latest level with respect to the application. This is also called as back leveled instance.

deprovision

To delete a provisioned application instance from the target system. The user who provisioned the application can deprovision it. In addition, team administrators can deprovision instances of applications owned by the team, regardless of who provisioned them.

environments

An *environment* is a collection of subsystems that are used by teams. The super administrator creates environments after subsystems have been registered. After teams have been defined, each environment can be assigned to one or more teams.

provision

To copy resources, or objects, of an application that is defined in the original environment to a subsystem in the environment assigned to your team so that you can use those resources for development and testing without affecting the originating application.

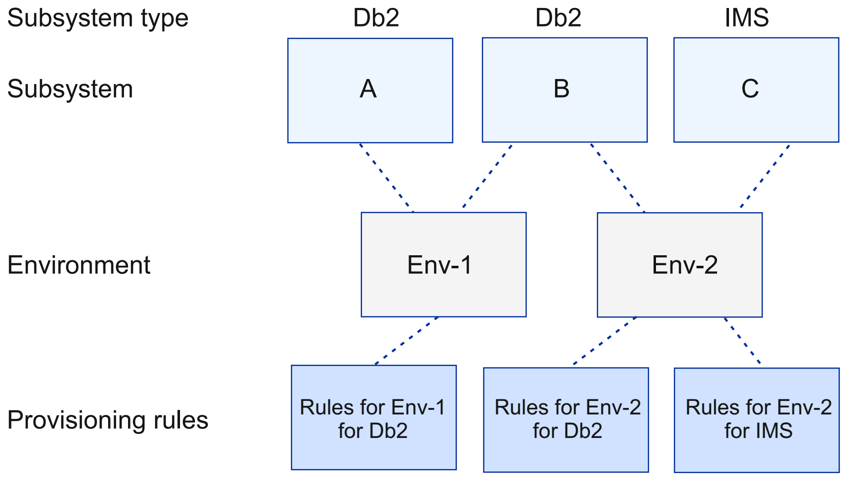
With Db2 DevOps Experience, a team administrator or a team member can provision instances of applications onto environments that are assigned to the team to build their own environments for developing and testing application programs for which they are responsible. When they provision an instance of an application, the referenced objects are copied to the environment's subsystems.

Team members can change definitions of the provisioned objects if the originating application is owned by the team. The changes made on the instance can be reviewed by the team's team administrators and optional reviewers in the team through a pull request process. The changes can be merged into the original objects defined in the instance's originating application only after a team administrator has approved the pull request.

provisioning rules

A set of rules that define how application objects are to be named when the user provisions an application instance. You can define one set of provisioning rules for each subsystem type in an environment. If you do not define any provisioning rules for an environment, Db2 DevOps Experience will automatically determine, during provisioning, the names of application objects based on the default provisioning rule for each object type to avoid naming conflicts.

For example, suppose there are two Db2 subsystems A and B and an IMS subsystem C, and environment Env-1 is associated with A and B and Env-2 is associated with B and C. You can create only one set of provisioning rules for Env-1 because all the associated subsystems are of the same subsystem type, and you can create two sets of provisioning rules for Env-2, one set for each subsystem type.



pull requests

To request a review of the changes made to a provisioned application instance so that interested parties can review those changes and discuss potential impacts before merging the changes to the originating application. For any changes that have been made to an application instance, at least one team administrator of the team that owns the originating application must approve of those changes before they get merged into the originating application.

site rules

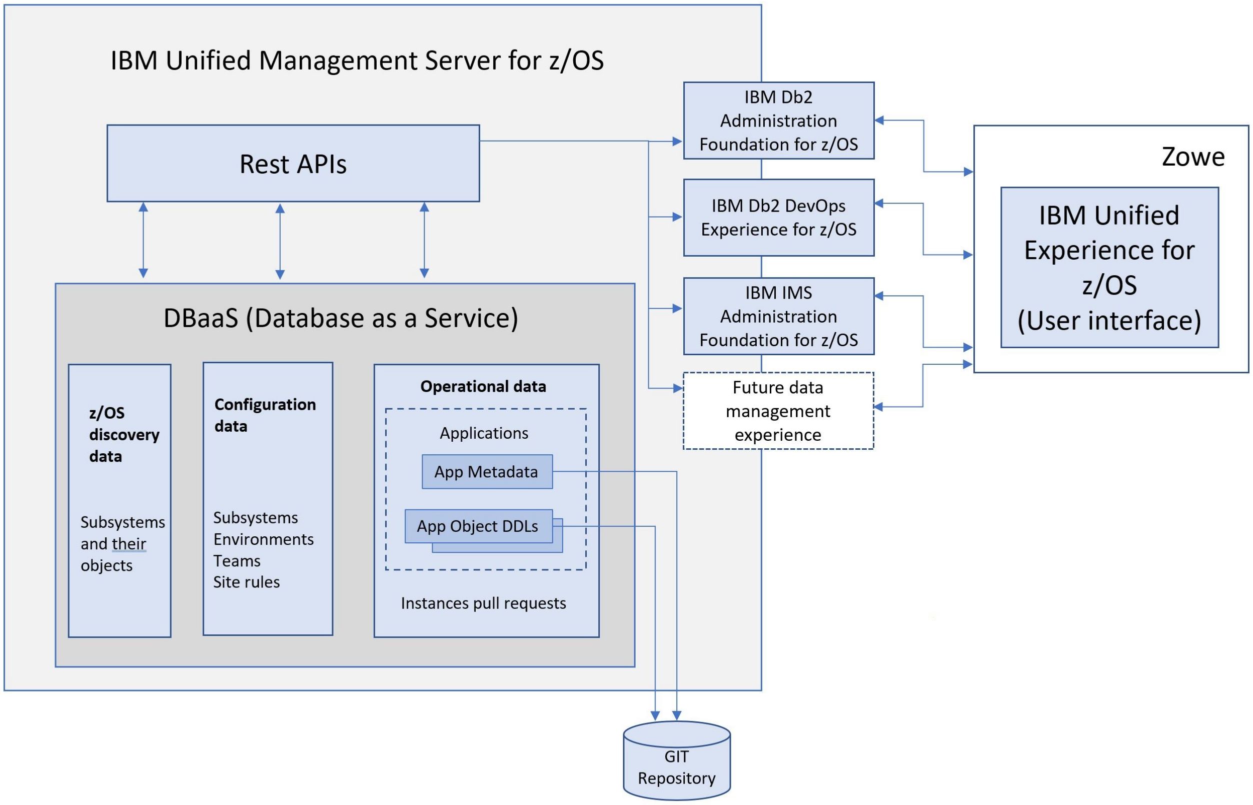
Rules that constrain how developers can change object definitions in provisioned application instances. Super administrators can create as many site rules as they want. Site rules can be associated with applications and environments.

teams

A *team* is a group of Unified Management Server users who work together toward a goal such as application development or test environment creation. Each team can be associated with the environments that it can use. Super administrators can create teams and assign environments. If SAF-based security is used, the super administrator should assign permissions to the security administrator to perform team management.

For details, refer to "Setting up users and teams" on page 46.

The following figure illustrates a high-level architecture of how Db2 DevOps Experience, together with Unified Management Server, manages and maintains various information.


IBM Unified Management Server for z/OS
Rest APIs
DBaaS (Database as a Service)
z/OS discovery data
Subsystems and their objects
Configuration data
Subsystems
Environments
Teams
Site rules
Operational data
Applications
App Metadata
App Object DDLs
Instances pull requests
IBM Db2 Administration Foundation for z/OS
IBM Db2 DevOps Experience for z/OS
IBM IMS Administration Foundation for z/OS
Future data management experience
Zowe
IBM Unified Experience for z/OS (User interface)
GIT Repository

In this figure, *application metadata* is a list of objects that make an application, along with their properties and relationships. Although application metadata cannot be viewed from IBM Unified Experience for z/OS web interface, it can be viewed from REST APIs.

Application object DDL statements define and manage application objects.

Db2 DevOps Experience workflow

It is important that you understand the basic workflow of Db2 DevOps Experience.

Summary of workflow

The following figure illustrates the general workflow of Db2 DevOps Experience. The workflow consists of three major tasks, each of which consisting of several sub-tasks. Typical user roles are also shown for each sub-task. For a detailed information about user roles and their tasks, see "Roles and responsibilities" on page 106.

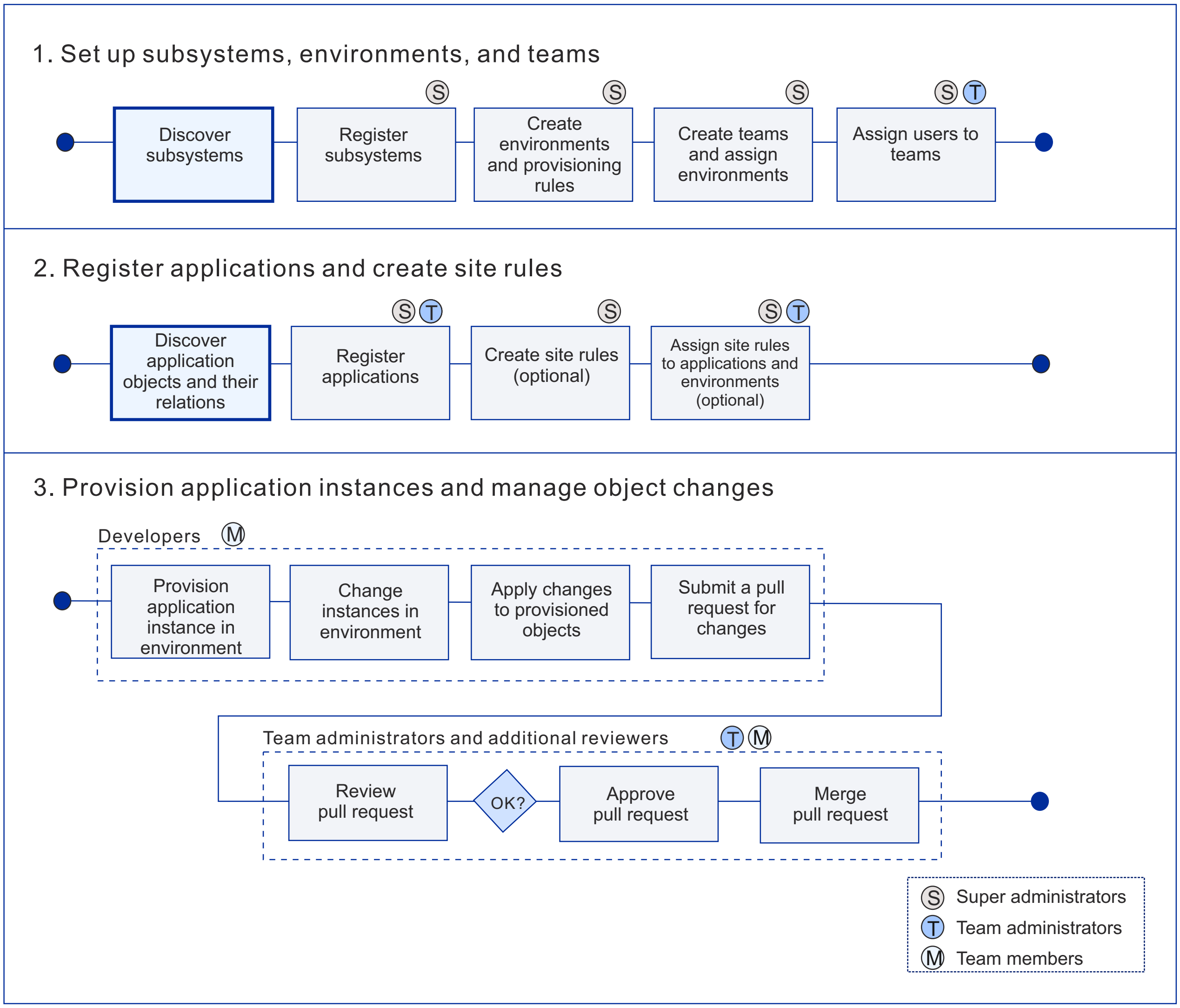

1. Set up subsystems, environments, and teams
Discover subsystems
S
Register subsystems
S
Create environments and provisioning rules
S
Create teams and assign environments
S T
Assign users to teams
2. Register applications and create site rules
Discover application objects and their relations
S T
Register applications
S
Create site rules (optional)
S T
Assign site rules to applications and environments (optional)
3. Provision application instances and manage object changes
Developers M
Provision application instance in environment
Change instances in environment
Apply changes to provisioned objects
Submit a pull request for changes
Team administrators and additional reviewers T M
Review pull request
OK?
Approve pull request
Merge pull request
S Super administrators
T Team administrators
M Team members

Figure 8. General workflow of the tasks and activities of users and administrators in Db2 DevOps Experience

Step 1: Setting up subsystems, environments, and teams

Super administrators need to set up subsystem environments for application development teams.

To set up subsystem environments, the super administrator needs to do the following tasks:

Step 1.1: Discover subsystems

This step is initially done by the system. When the Unified Management Server is started, it automatically discovers all Db2 and IMS subsystems in your z/OS environment.

After this, super administrators can manually run subsystem discovery by refreshing the **Subsystems** interface.

Step 1.2: Register subsystems

You must register subsystems to make their objects available in Db2 DevOps Experience.

To register a subsystem, you select a subsystem from the list of subsystems discovered by the Unified Management Server and register it.

You must have the super administrator role to register subsystems.

Step 1.3: Create environments and provisioning rules

You must create environments for each stage of the product development lifecycle. When you create an environment, you add subsystems to the environment so that teams can provision subsystem objects as application instances and work on them.

You can optionally assign provisioning rules to the environment for each subsystem type that is registered with the environment. For details about provisioning rules, see "Db2 DevOps Experience terms and concepts" on page 156.

You must have the super administrator role to create environments.

Step 1.4: Create teams

You create teams so that team members can develop applications together. When you create a team, you can associate the environments that the team can work on.

You must have the super administrator role to create teams.

Step 1.5: Assign users to teams

You can add users to teams as members or team administrators. To be added to a team, the user must belong to the Unified Management Server user group.

In addition to super administrators, team administrators can also add or remove team members for their own teams. A typical scenario might be:

1. Jane, the super administrator, creates a team
2. Jane assigns environments to the team
3. Jane assigns Tom as a team administrator of the team
4. Tom assigns other users to the team, either as members or team administrators

Step 2: Registering applications and creating site rules

Super administrators or team administrators need to set up application definitions so that developers and testers can provision the applications to work on their own application instances. Team administrators can register and delete their own team's applications and change application settings.

Step 2.1: Discover application objects and their relations

This step is initially done by the system. Object discovery is performed automatically whenever a new subsystem is registered or when the Unified Management Server is restarted. The discovery status will be shown on the **Registered Subsystems** tab in the **Subsystems** page.

You can also initiate object discovery on registered subsystems. Only a single subsystem can be rediscovered at any point in time.

Step 2.2: Register applications

You define a group of objects as an application so that developers and testers can provision application instances in their own environments and work on the objects included in those instances.

To register applications, you must have the super administrator or team administrator role.

Step 2.3: Create site rules (optional)

You can optionally create site rules to constrain how developers can change object definitions in provisioned application instances. You can create as many site rules as you want.

Step 2.4: Assign site rules to applications and environments (optional)

In order for a defined site rule to be applied to a set of application instances, you need to associate the site rule with applications or environments. If a site rule is assigned to an application, the rule applies to any application instances provisioned from that application. If a site rule is assigned to an environment, the rule applies to any application instances provisioned to that environment.

Step 3: Provisioning application instances and managing object changes

Developers and testers can provision application instances into their team's environments. They can then change the definitions of the application objects included in the application instances in their own local environments for application program development and testing. After they make changes and submit pull requests, team administrators and additional reviewers of the team that owns the originating application can review their changes and approve or decline those requests.

In the following, steps 3.1 through 3.4 are done by developers and testers, and steps 3.5 through 3.7 are done by team administrators and additional reviewers of the team that owns the originating application.

Step 3.1: Provision an application instance in a team's environment

You can provision a new application instance by specifying an instance name and selecting the subsystem type, the name of the application that you want to create an instance from, one of your teams, and an environment that is assigned to the team. The subsystem on which the instance is provisioned will be selected automatically by the system if more than one subsystem of the same type is assigned to the selected environment. After an application instance is provisioned to the selected environment, you can use the built-in SQL processor to run an SQL query against an application object in the instance to verify successful provisioning.

Step 3.2: Change the instances in that environment

After an application instance has been provisioned, you can use that instance on your work. To modify an application object included in the instance, an object definition source editor is provided for you to make changes as you like. Any syntax errors are indicated with icons for immediate feedback. If your change does not comply with the site rules that are associated with the application or the environment, one or more error or warning messages will be displayed.

Step 3.3: Apply the changes to the provisioned objects and test them

When you have finished editing an object definition, click **Apply object changes** to apply the changes to the provisioned object. You can modify more than one object and apply all the changes in a single operation by clicking **Apply all changes**. You can optionally use the built-in SQL processor to run SQL queries against the changed application objects to verify the changes.

Step 3.4: Submit a pull request for the changes

When all the work related to the application instance is completed, you can request that your changes be merged to the originating application. To submit a request, open the instance details page and click **Create pull request**. The pull requests that you submit are displayed in the **Open pull requests** widget on your Unified Experience **Dashboard**.

Your pull request must be reviewed and approved before it is merged to the originating application. All team administrators of the team that owns the originating application are automatically assigned as default pull request reviewers. The pull request submitter can add additional reviewers from a list of the team members of the application-owning team.

Step 3.5: Review the pull request

When a pull request is submitted, the request is displayed in the **Pull requests to review** widget of the reviewers' Unified Experience **Dashboard**.

They can review the changes to decide whether to approve or decline it. If there is any problem, they can decline the request so that the pull request submitter can make corrections or additional changes based on the comments provided by the reviewers. The pull request submitter can withdraw the pull request by using the **Decline** function.

Pull requests must be approved by at least one team administrator. Because team administrators cannot approve their own pull requests, at least two team administrators must be assigned to approve a pull request if team administrators themselves join the application development.

Step 3.6: Approve the pull request

If a pull request reviewer is good with the changes, the request can be approved. When at least one team administrator approves the pull request, the **Merge** button gets enabled, and the changes in the pull request will become ready to be merged to the originating application.

Step 3.7: Merge the pull request

A team administrator or an additional reviewer can click **Merge** to merge all the changes that the developer made to the application instance into the originating application.

Managing applications

An application is a set of objects that are grouped together, enabling you to manage and provision them as a single unit. Super administrators can register, change the settings of, or delete applications.

Application objects are logical, meaning that they are only references to the objects. When users provision instances of an application, the referenced objects are copied to create the instances.

By registering applications, you can make their objects available to data management experiences so that you can provision instances of those applications.

Selecting objects

To register an application, you need to compile a list of the objects that you want to include.

Home application

Home application defines the primary ownership and determine the editability of an object. An object must be defined in a home application before it can be referenced in a reference application. An application can both be a home and reference application to different objects. For example, an object referenced in multiple applications can be a home object in the current application.

The home application assignment determines object editability. Objects in an application are edited by using a provisioned instance. The instances must be provisioned by the application owning team.

Selected objects

These are the objects that you want to add to your application. Some objects that you select will be defined only in your application. Your application will be the home application for these objects.

Some objects might be defined in another application but are being referenced in your application because you have selected them. Your application will not be home for these objects. Though the objects have one home application, they can be used in multiple applications.

Required objects

These objects are required by the selected objects. The data management experience automatically adds these objects to your application. For example, if in Db2 DevOps Experience you select a table object for a Db2 application, the Db2 database, table space, and indexes for this table object are automatically added. The primary purpose of automatically pulling in required objects is to create consistent instances that perform similarly to these objects in integrated environments. Aliases, indexes, and triggers are always pulled with the table on which they are defined.

Related objects
These objects are related to the selected or required objects and are not essential to your application. You can choose to include or exclude none, some, or all these objects.

Required objects of the related objects
These objects are required by the related objects. If you select a related object, DevOps Experience automatically adds the required objects to your application. Your application will become the home application for these required objects. If you do not want this application to become their home application, you must create another home application for these required objects.

The flexibility to include and exclude specific objects while creating an application enables you to create applications that suit your needs.

Registering Db2 applications

Register Db2 applications so that developers can provision Db2 application instances.

Before you begin

You need a team to register an application. Ensure that you have created a team with at least one team administrator.

Note: The chunk size decides the number of objects included in a single JCL during DDL generation. The default value for the `CHUNK_SIZE` parameter is 200. You can configure the `CHUNK_SIZE` parameter by specifying a definition for the `javaArgs` element in `PARMLIB(ZWEYAML)`. Every JCL runs in parallel. Proceed with caution while configuring the `CHUNK_SIZE` parameter to avoid extra load on the server. Refer to the following example:

```
components:
  izp:
    server:
      javaArgs:
        --Dcom.<servername>.appManifest.getddl.chunkSize=50
```

It is recommended to set this parameter value to less than 255.

Procedure

To register a Db2 application, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Applications**.
3. Click **Register application**.
4. To select the type of the subsystem in which you want to register an application, click **Subsystem Type**.
5. To select the subsystem that contains the application objects, click **Select source subsystem**, and then click **Next**.
6. Take either or both of the following steps:

   a) To find objects, click **Search object by type** and add the objects that you want in the application:

      i) Select the type of object you are searching for. For example, **Database**.

      Note: For stored procedure, the user can select a stored procedure with a specific version from the version column. For function, the user can select a function with a specific name from the name column.

      ii) Enter search terms in the fields, and then click **Search**.

      Note: You can substitute wildcard characters for characters in your search terms. Use a percent sign (%) to represent zero or more characters. Use an underscore (_) to represent a single character. The search is case-sensitive.

While objects are being searched for, you can always cancel the search by clicking **Cancel search**.

iii) To include objects in the application, expand results and select objects. For example, **Table**. Selected objects display in the **Selected objects** panel. To remove the objects that you selected, click the **Remove** icon.

iv) Repeat these steps until you have added all objects that you want in the application.

b) Select any associated applications from the **Associated applications** menu.

7. Click **Next**.

   Important:

   - The **Home** icon next to the object in the **Selected objects** panel indicates that your application is not the home application for these objects. To see which application is the home for the object, click the **Home** icon.
   - When there is no icon next to an object in the **Selected objects** panel, it indicates that the object is being defined in your application and your application is the home application for the object.
   - The **Link** icon next to the object in the **Selected objects** panel indicates that other objects require this object. To see the list of these required objects, click the **Link** icon. These required objects are automatically added when you select an object.
8. Optional: To add the related objects, click the **+** icon next to the object. This object is moved to the **Selected objects** pane.

   Note: For more information about related and required objects, see “Selecting objects” on page 162.
9. Type a name for your new application in the **Name** field.

   The application name is case-insensitive; that is, lowercase and uppercase letters are not distinguished.
10. Click the **Color** swatch to select a color that will identify this application when it appears in the **Applications** page.
11. Briefly describe the application in the **Description** field.
12. Select a team from the **Owner team** menu. The owning team is responsible for the application and owning team administrators approve pull requests for the application.

    Important:

    - Select a team that has an administrator. Only a team with a team administrator can own an application.
    - When you select objects, the system automatically adds objects that are required for the selected objects. For example, if you select a table, an index is automatically created for that table. Your application will become the home application for these automatically created objects. If you do not want this application to become the home application of these objects, add them to another application before referencing them in this application.
13. To save and register the application, click **Register**.

Results

The **Applications** page displays the application that you registered.

Viewing application details

You can view detailed information about the objects that are defined in an application.

Procedure

To view an application, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Applications**.

The **Applications** page opens. This page shows a list of the applications that have been registered.

3. On the application that you want to view, click **Details**. Alternatively, you can click the overflow menu and select **View application**.

   The **Application details** page opens and displays a list of the objects that are defined in this application.

   - The ⌂ icon is displayed next to the objects whose home application is the application that you are viewing.
   - The ⌂ icon and the object's home application are displayed next to the objects whose home application is different than the application that you are viewing.

What to do next

You can view the following information on this page:

- To search and switch to a different object view in the same application, use the **Search objects** option.
- To see the DDL of the object, click the **View** icon next to the object name.
- To see the DDL of the objects in the application, click **Show DDL**. The **Object definition page** opens. You can switch between application objects that are shown in the left pane. The right pane shows the DDL of the selected object. You can copy the DDL by clicking **Copy to clipboard**. By default, the DDL of the first object of the application is shown in the right pane.
- To see the list of applications in which a home object of this application is being referenced, click the **References** link next to the home object.
- To go back to the **Application** page, click **Application** in the breadcrumb, which is located at the top of the page.
- To see the objects in the associated applications that you added while registering the application, click the **Associated applications** tab.

Changing application settings

You can change the application settings to modify the application name, color, description, and application team.

Before you begin

- Only the UMS super administrator or an application team administrator can change application settings.
- To change application settings, an application must be in one of the following states:
  - Ready
  - DDL generation error
  - Source generation error
  - Apply error
  - Delete error
  - Error
- If an application has an active pull request, the **Owned by** field is disabled.
- If an instance is in one of the following states, the **Owned by** field is disabled.
  - Loading
  - Instantiating
  - Deleting
  - Synchronizing
  - Updating

- If you are logged in as a UMS super administrator, the **Owned by** field will list all teams with at least one team administrator.
- If you are logged in as a UMS team administrator, the **Owned by** field will list all teams where the logged in user is a team administrator.
- If you select to update instances, the access is revoked for the old team members and granted to the new team members.
- If a change in application ownership is in progress, you cannot update instances.

Procedure

To change an application's settings, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Applications**.

   The **Applications** page opens. This page shows a list of the applications that have been registered.
3. On the application that you want to change, click the overflow menu (the vertically aligned three-dotted icon on the upper right corner), and select **Change settings**.
4. Edit the following as needed:
   a) Name
   b) Color
   c) Description
   d) Owned by

      Select a new team for this application.
   e) Toggle the **Update instances currently owned by team <old team name> to <new team name>** option if needed. This option is available if the selected old and new teams have the same environments.

      Review the current and expected status of instances.
5. Click **Save**.

Results

The application settings are modified for the current application.

Deleting applications

Super administrators and the administrators of the team that created the application can delete the application from IBM Unified Experience for z/OS.

Before you begin

Ensure the following before deleting an application:

- An instance is not provisioned from this application.

  Note: An instance cannot be deprovisioned when there are active pull requests. These pull requests must first be resolved (declined or approved/merged).
- This application is not associated with other applications.
- The objects, for whom this application is the home application, are not referenced in other applications. The objects must be moved to another home application before proceeding with the delete task.

Procedure

To delete an application, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Applications**.

The **Applications** page opens. This page shows a list of the applications that have been registered.

3. On the application that you want to delete, click the overflow menu (the vertically aligned three-dotted icon on the upper right corner), and select **Delete application**.
4. Complete the steps that follow.

Results

The application is removed from the subsystem.

Moving objects

This feature enables you to move the home objects from one application to another. *Home objects* are the objects that are defined in your application. Your application is the home application for these objects. You will need to move objects if you want to delete an application whose home objects are referenced in other applications. You can also move objects if you do not want them to be the home objects of an application.

Before you begin

Ensure the following before moving the home objects from one application to another:

- The application to which you want to move the objects is in the ready state and is in the same subsystem as the current application.
- You are either the super administrator or the team administrator of both the applications.

Procedure

To move the objects from one application to another, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Applications**.

   The **Applications** page opens. This page shows a list of the applications that have been registered.
3. On the application whose objects you want to move, click **Details**.

   The **Application details** page opens. The **Application objects** tab on this page shows a list of the objects that are defined in this application.
4. Select the objects that you want to move:

   - To move all the referenced objects, click **Select all reference objects**.
   - To move all the home objects, click **Select all home objects**.

   Tip: You can also manually select or clear each object.
5. To select the application where you want to move the selected objects, click **Move selected objects**.

   The **Move home objects** dialog opens.
6. On the **Move home objects** dialog, select the application where you want to move the selected objects, and then click **Move**.

Results

The selected objects from the current application are moved to the selected application.

Managing site rules

Super administrators can create one or more site rules to guide how team members should change object definitions in provisioned application instances. Each rule can be associated with one or more applications and environments. Also, site rules can be associated with all existing applications at once and you can configure them to be associated with newly created application by default.

The purpose of site rules is to allow super administrators to define the site standards such as naming rules and attribute values. You can define a site rule for a particular object type of a particular subsystem type.

Assigning a site rule to an environment means that the rule is applied to all objects of the specified object type in application instances that have been provisioned to a subsystem defined in that environment.

Assigning a site rule to an application means that the rule is applied to all objects of the specified object type in application instances that have been provisioned from that application.

When a user who is allowed to edit object definitions in a provisioned application instance tries to change the definitions, the changes are evaluated against the following:

- Site rules that are associated with the originating application of the provisioned instance
- Site rules that are associated with the environment to which the application instance was provisioned

For example, if site rule SR-1 for a Db2 object type is assigned to environment ENV-A and site rule SR-2 for a Db2 object type is assigned to Db2 applications APPL-1 and APPL-2, and an instance of application APPL-1 is provisioned to a Db2 subsystem in environment ENV-A, any changes to the corresponding Db2 objects in the instance will be evaluated against both site rules SR-1 and SR-2.

Note: You can assign a site rule for a particular subsystem type to 1) applications of any subsystem types and 2) environments that consist of subsystems of any subsystem types. However, the site rule is not applied to an application instance that has been provisioned from an application of a different subsystem type or to an environment that does not have any subsystem of the same subsystem type. For example, a Db2 site rule can be assigned to IMS applications or the environments that consist of only IMS subsystems, but the rule is not applied to application instances that are provisioned from those applications or to those environments.

If the changes violate associated site rules, notifications will be displayed; however, site rule violations can be ignored. In Db2 DevOps Experience, if site rule violations occur while adding a new object, they cannot be ignored. If site rule violations occur while editing the DDL, you can ignore the violations.

Creating Db2 site rules

Db2 site rules specify how developers should change object definitions in application instances that were provisioned from Db2 applications.

About this task

You can create simple and complex site rules. Simple rules are built using three segments: object, attribute, and verification type. For example, you can create a site rule that specifies that table (object) names (attribute) must start with (verification type) the string "TAB". When applied to an application or environment, developers creating or editing a table in a provisioned application instance are notified if they violate the rule.

Complex site rules are more flexible and powerful. You define them using a small domain-specific programming language that is composed of a subset of the Python syntax. Every valid complex site rule is a Python expression that is evaluated and resolved to either True or False according to normal Python rules. For example, an empty string is False, and the number 7 is True. Expressions can use the following features:

- Python ternary expressions (x if y else z)
- Boolean and/or/not
- Parentheses
- <
- >
- <=
- >=
- ==

- !=
- +
- -
- String subscription (stride not supported)
- The functions: `len`, `startswith`, and `endswith`

Supported data types are integers, strings (delimited by double or single quotation marks), and Boolean operators.

Complex site rules have one non-standard Python feature: a predefined variable for every object attribute. For example, to specify that “table names must start with the letter *T*", you would select **Table** from **Object** to create the following site rule:

```
name.startswith(“T”)
```

The predefined string variable *name* holds the value of the table name. The rest is standard Python syntax. Here is another valid way to write that rule:

```
name[:1] == “T”
```

For a list of objects and their attribute variables, see “Complex Db2 site rule variables” on page 240.

Rules can be combined to make standard Python expressions. For example, here is one way to create the rule “if a table space name is between 4 and 6 characters, tables in that table space must end their names with the last four characters of their table space”:

```
name.endswith(TSNAME[-4:]) if 4 <= len(TSNAME) <= 6 else True
```

Note: Python ternary expression is different from ternary expressions in C/C++/Java and is different from a Python if/else block.

Important: Values must be uppercase. The rule `log=="LOGGED"` works; the rule `log=="logged"` does not work.

The objects and attributes supported by complex site rules are the same as those supported by simple site rules.

To create and apply a site rule, complete the following steps:

Procedure

1. On the navigation menu, click **Manage**, and then **Site rules**.

   The **Site rules** page opens. This page shows a list of the site rules that have been created.
2. Click **Create rule (Db2)**.
3. Complete either of the following steps:
   - To create a simple rule, follow these steps:
     a. In the **Rule name** field, enter the name of the rule. The rule name is case-insensitive; that is, lowercase and uppercase letters are not distinguished.
     b. From the **Object** field, select the type of object for which you want to specify a rule. For example, select **Database** to specify a rule on database objects.
     c. From the **Attribute** field, select the attribute to which you want this rule to be applied. The attributes are different for each object.
     d. From the **Constraint** field, select a constraint.
     e. In the **Value** field, either type values or select an item from the drop-down list, depending on the condition that you specified in previous steps. The rule that you specified will be shown at the bottom of this dialog.
     f. Click **Create**.

- To create a complex rule, follow these steps:
    a. Click the **Simple rule** switch to turn it into **Complex rule**.
    b. In the **Rule name** field, enter the name of this rule. The rule name is case-insensitive; that is, lowercase and uppercase letters are not distinguished.
    c. From the **Object** field, select an object type.
    d. Enter the rule in the editor pane. Make sure that values are uppercase.
    e. Click **Create**.

4. To apply a rule, select it and click **Assign to applications** or **Assign to environments**.
5. Find the applications or environments to apply the rule to, and then click **Update**.
6. To edit, delete, or duplicate a rule, or to assign it to different applications or environments, click the overflow menu for the rule and select an appropriate menu item.

Editing site rules

Team administrators and the user who created the site rule can edit the site rule.

Procedure

To edit a site rule, complete the following steps:

1. On the navigation menu, click **Manage**, and then **Site rules**.

   The **Site rules** page opens. This page shows a list of the site rules that have been created.
2. Select the site rule that you want to edit, click the overflow menu of that rule, and click **Edit rule**.

   The **Edit site rule** page opens.
3. Edit the site rule, and then click **Update**.

   The updated site rule is displayed on the **Site rules** page.

Duplicating site rules

Duplicate a site rule to quickly create a new one that is similar to an already existing site rule.

Procedure

To duplicate a site rule, complete the following steps:

1. On the navigation menu, click **Manage**, and then **Site rules**.

   The **Site rules** page opens. This page shows a list of the site rules that have been created.
2. Select the site rule that you want to duplicate, click the overflow menu of that rule, and click **Duplicate rule**.

   The **Duplicate rule** page opens. A new name is assigned to the duplicate site rule. You can either retain or edit the current information as necessary.
3. Edit the information, and then click **Create**.

   The site rule is displayed on the **Site rules** page.

Assigning site rules to applications

After creating a site rule, super administrators can assign it to applications. Team administrators can assign site rules to applications that are owned by their teams. Assigning a site rule to an application means that the rule is applied to all objects of the specified object type in application instances that have been provisioned from that application. If you are creating an associated application from a parent application, the site rule applied to the parent application will be applied to your associated application.

Procedure

To assign one or more site rules to one or more applications, do the following:

1. On the navigation menu, click **Manage**, and then **Site rules**.

   The **Site rules** page opens. This page shows a list of the site rules that have been created.
2. Complete either of the following steps:

   - To assign a single site rule to one or more applications, click the overflow menu of that rule, and select **Assign to applications**.
   - To assign multiple site rules to one or more applications, select the check boxes of those rules, and select **Assign to applications** from a list of batch action menu items, which is displayed above the table.

   The **Assign site rules to applications** dialog opens.
3. From the table, select the applications that you want to associate the selected site rules with.

   You can assign as many applications as you want. If you are a super administrator and want to assign these site rules to all current and future applications, select **Assign to all current and future applications**.

   If an application is associated with multiple site rules that specify different rules for the same object type, attribute, and constraint, the number of those site rules in conflict will be shown under the **Conflicts** column. If there are conflicting site rules, you must remove them.
4. Click **Update**.

   The names of the applications that you assigned will be displayed in the **Applications** column of the selected site rules.

Assigning site rules to environments

After creating a site rule, super administrators can assign it to environments. Assigning a site rule to an environment means that the rule is applied to all objects of the specified object type in application instances that have been provisioned to a subsystem defined in that environment.

Procedure

To assign one or more site rules to one or more environments, do the following:

1. On the navigation menu, click **Manage**, and then **Site rules**.

   The **Site rules** page opens. This page shows a list of the site rules that have been created.
2. Complete either of the following steps:

   - To assign a single site rule to one or more environments, click the overflow menu of that rule, and select **Assign to environments**.
   - To assign multiple site rules to one or more environments, select the check boxes of those rules, and select **Assign to environments** from a list of batch action menu items, which is displayed above the table.

   The **Assign site rules to environments** dialog opens.
3. From the table, select the environments that you want to associate the selected site rules with.

   You can assign as many environments as you want.

   If an environment is associated with multiple site rules that specify different rules for the same object type, attribute, and constraint, the number of those site rules in conflict will be shown under the **Conflicts** column.
4. Click **Update**.

   The names of the environments that you assigned will be displayed in the **Environments** column of the selected site rules.

Deleting site rules

If a site rule is no longer needed, super administrators can delete it.

Procedure

To delete one or more site rules, do the following:

1. On the navigation menu, click **Manage**, and then **Site rules**.

   The **Site rules** page opens. This page shows a list of the site rules that have been created.
2. Complete either of the following steps:
   - To delete a single site rule, click the overflow menu of that rule, and select **Delete rule**.
   - To delete multiple site rules, select the check boxes of those rules, and select **Delete rules** from a list of batch action menu items, which is displayed above the table.
3. When a confirmation message is displayed, click **Delete**.

   The selected site rules are deleted.

Setting storage limits

When developers provision applications, those applications occupy storage. To manage this storage, you can set limits by team, environment, user, and application. The limits are soft, meaning when they are exceeded IBM Unified Experience for z/OS displays alerts but does not prevent continued activity.

Procedure

To set limits, complete the following steps:

1. Click the navigation menu and select **Manage**.
2. Click **Storage**.
3. Click the tab that represents how you want to limit storage. For example, to limit storage by environment, click **By Environment**.
4. Find and select the environment that you want to limit, and then click **Set limit**.
5. Specify the storage limit for that environment. For example, specify 15 GB to specify that provisioned instances of applications in that environment that occupy more than 15 GB in the Db2 database will generate warnings.

Configuring Db2 privileges

To provision instances using Db2 DevOps Experience, you need to grant Db2 privileges. You can configure these privileges using the `IZPD2DPM` `PARMLIB` member. By default, these privileges are granted.

Procedure

To grant Db2 privileges, perform the following steps:

1. Edit the `IZPD2DPM` `PARMLIB` member.
2. Configure the following parameters:

Table 50. Db2 DevOps Experience privileges

Privilege type	Description	Default value
`SYSTEM_PRIVILEGE`	`BINDADD` and `CREATEALIAS`	True
`DEV_PRIVILEGE`	`CREATEIN`, `ALTERIN`, `DROPIN` and `COLLECTION`	True
`DBADM_PRIVILEGE`	Database administration authority (DBADM)	True

3. Restart UMS.

Configuring pull request privileges

The instance owner and instance reviewer have the privilege to view pull requests in the Db2 DevOps Experience by default. To enable the super administrator and team members to view pull requests, you need to configure the `IZPD2DPM` `PARMLIB` member. By default, this value is set to false.

Procedure

To configure Db2 pull request, perform the following steps:

1. Edit the `IZPD2DPM` `PARMLIB` member.
2. Set the `enableExpandedPRViewing` parameter to true in the API.
3. Restart Zowe.

Results

The super administrator and team members can view all the pull requests, pull request details, comments from other users, and pull request differences. For details, refer to "Roles and responsibilities" on page 106.

Managing instances

An application instance is a set of application objects that has been provisioned into the associated environment's subsystem.

To open the **Instances** page, click **Explore**, and then **Instances** on the navigation menu. If instances have already been provisioned, they will be displayed on the page. You can filter the instances by switching between **Your instances**, **All team instances**, and a specific team name.

Provisioning instances

You can provision an instance of an application to develop and test your program code that uses the objects in the application. A provisioned application instance is a copy of the objects in the original application. You can safely make changes to the objects in the provisioned instance without affecting the original objects of an application.

Before you begin

Before you can provision an instance of an application, a super administrator must register subsystems that contain the application objects, create a team and an environment, assign you to the team, create the application, and assign a team to the application.

You can provision instances of any application. However, in order to edit objects in the provisioned instance and to create pull requests for that instance, you must be a member of the team that owns the application.

Procedure

To provision an instance, complete the following steps:

1. On the navigation menu, select **Manage**.
2. Click **Application**.
3. On the application that you want to provision, click **Details**.
4. In the application details page, click **Provision instance**.

   Note: Instead of these steps, you can alternatively open the **Provision instance** dialog, as follows:

   a. On the navigation menu, select **Explore**.

b. Click **Instances**.

c. Click **Provision instance**.

5. In the **Provision instance** dialog, specify the following fields:

Instance name
Type the name of the new instance. The instance name is case-insensitive; that is, lowercase and uppercase letters are not distinguished.

Subsystem type
If you opened this dialog from the application details page, the subsystem type of the selected application is preset in this field. Otherwise, select a subsystem type to filter the applications that will be shown in the pull-down list of the **Application name** field.

Application name
If you opened this dialog from the application details page, the name of the selected application is preset in this field. Otherwise, select the application from which you want to provision an instance.

If the application is a Db2 application and a limit is set on the total size of all the instances for the application and the limit has already reached, you cannot provision that application.

Include data from source application
Select this check box if you want to include data from the source application.

Team
Select the team under which the instance is to be provisioned. If you are a member of multiple teams, you must specify which team you are provisioning the instance for.

If you are provisioning an instance from a Db2 application, storage limits can be placed on teams.

Environment
Select the environment to which you want to provision the instance.

If an instance limit is set for the environment and the limit has already reached, you cannot provision an instance to this environment.

Additional instance owners
This drop-down menu lists all the members of the team that you specified in the **Team** field. Select the users that you want to add as owners of this instance. You will be the default owner of this instance.

6. Click **Provision**.

The provisioning process might take a while. After the instance is provisioned, it will be displayed on the **Instances** page. Your instances are also displayed on the **Dashboard** page.

7. On the instance that has just been provisioned, click **Details**.

You will see the objects that are included in the provisioned instance.

What to do next

- Optionally, you can use the built-in SQL processor to run an SQL query against an application object in the instance to verify successful provisioning. For details on how to use the SQL processor, see "Using SQL processor for Db2" on page 125.
- After you have provisioned the instance, you can rename it by using the `Update instance fields` API. You can use the same API to add or remove the owners of this instance. To know how to use the API, see the "Swagger documentation" on page 265.

Editing instance objects

Anyone with the privileges to edit objects in a particular instance can make changes to the object definitions. To check the relationship between the user role and the instances that they can edit, see "Roles and responsibilities" on page 106.

Procedure

To edit application instance objects, complete the following steps:

1. On the navigation menu, select **Explore**.
2. Click **Instances**.
3. Find the instance that you want to edit and click **Details**.

   Note: If, for some reason, the instance was not provisioned properly, the **Details** button is not available.

   The **Instance details** page opens. The **Instance objects** tab lists the objects that have been provisioned from the original application. To see how the schemas, databases, and collections in the instance are mapped to the schemas, databases, and collections in the application, click the **Name mapping** tab.

   The ❌ icon next to the object name indicates a violation and the ⚠ icon indicates a warning. To resolve the violations and warnings, edit the object DDL.

   Important: You can create a pull request only after resolving the violations.

   If the selected instance includes objects from associated applications, the list also includes the names of those objects. For each of those objects, the name of the associated application is shown.

   Important: On the **Instance details** page, you can edit only those objects, whose home application is the original application from which the instance was provisioned. You can only view the objects whose home application is not the original application.

   In Db2 DevOps Experience, adding Db2 objects to Db2 application instances is supported. For details, see “Adding Db2 instance objects” on page 177.
4. When the instance is not at the latest level with respect to the application, you cannot create a pull request. However, you can edit the Db2 DDLs, apply changes, or apply all changes. To create a pull request, you must get the instance at the latest level. Click **Update from source** on the instance object details page, instance tile page, or the editor page. An impact report opens listing objects that will be created, altered, or dropped. Click **Update from source**.

   Notes:

   - A user, who can edit an instance, can perform the **Update from source** task. An instance owner who does not belong to the application owner team can also perform this action.
   - When you reassign a Db2 object to another Db2 application, the object's home application and the application from which the instance was provisioned are no longer same. If you had previously edited this object, after updating this instance from the source, this object becomes non-editable in this instance and the status of the instance changes to `invalid_objects`. The team administrator or the super administrator must use the `Reassign object` API to change the object's home application back to the application from which the instance was provisioned. When you update this instance from the source, the object becomes editable and the status of the instance changes to `ready`. To know how to use the API, see “Swagger documentation” on page 265.
5. On the instance object page, click the **Show DDL** link.

   The object definition page opens. You can switch between instance objects that are shown in the left pane. If the selected object is shown as editable, you can modify the object definitions in the right pane.

   The ❌ icon next to the object name indicates a violation and the ⚠ icon indicates a warning. To resolve the violations and warnings, edit the object DDL.

   After getting the updates from the source, if there are conflicts, you can resolve the conflicts in the following ways:

At the object level:
: You can accept incoming changes from the source or accept the existing change in the current source. This conflict resolution is applied to the whole object.

At the conflict level:
: You can review the changes and decide whether to accept the incoming changes or keep the existing current changes or accept both changes at each conflict level. Alternatively, you can also resolve the conflict manually.

Note: A user who can edit an instance can perform the resolve conflict task.

6. Edit the object definitions.

   While editing an object, site rules are checked against object changes if rules have been created and assigned to the application that this instance was provisioned from or the environment into which this instance was provisioned. If the changes violate site rules, error messages are shown.

   Important:

   - If multiple users simultaneously edit the object definitions, changes made by one user might get overwritten with the changes made by another user. This might result in inconsistent object definitions.
   - If you rename an object, a new object is created and the current object stays in the subsystem. To avoid duplicate objects from being created, Db2 DevOps Experience does not support renaming objects.

7. When you are finished editing an object, take one of the following steps:

   - To apply changes to the current object, click **Apply object changes**.
   - To cancel your changes, click **Discard object changes**.
   - To edit another object in this instance, switch to that object by clicking the object name in the left pane and edit it. To apply all of your changes to multiple objects, click **Apply all changes**. To discard all changes to multiple objects, click **Discard all**.

   If you clicked **Apply object changes** or **Apply all changes**, a confirmation dialog opens. If your changes can cause problems, a message is displayed in the dialog, and you will have to resolve the errors before applying changes.

8. When a confirmation message is displayed, you can review the impact report and click **Apply DDL changes**.

   Your changes to the objects will be applied to the instance.

9. Applying changes might fail due to unavailability of the resource when some other utilities are already using the object. As a result, the instance goes into an apply error state.

   To proceed, choose one of the following options:

   - **Resume**: Resumes the instance from the previous failed point.
   - **Apply**: Starts the whole process again. There is a chance of data loss when the whole process is started again.
   - **Discard**: Discards the current changes. There is a chance of data loss when you discard apply changes.

     Notes: If you click the **Edit** button in the apply error notification, you need to do the following:

     – To discard the changes of single or multiple objects, click **Discard object changes** or **Discard all**, respectively.

       A pop-up appears on the screen with the message ‘Edit - DDL might now be out of sync with the catalog’.

     – To apply the changes in single or multiple objects, click **Apply current changes** or **Apply all changes**, respectively.

       Or

     – To cancel the changes, click **Cancel**.

What to do next

If you completed all object changes to an instance, you are now ready to create and submit a pull request to have your changes to the instance reviewed by the team administrators and other reviewers that you specify.

Adding Db2 instance objects

In Db2 DevOps Experience, the administrator of the application-owning team, or the instance owner who is also a member of the application-owning team, can edit an instance and add objects.

Procedure

To add Db2 application instance objects, complete the following steps:

1. On the navigation menu, select **Explore**, and then click **Instances**.

   The **Instances** page opens. Click **Details** to view the instance details for a specific instance.
2. To add objects to this instance, click **Show DDL**.

   The **Object definitions** page opens.

   The **Db2 objects** pane lists the objects that have been provisioned from the original application. This pane also lists the objects that have been provisioned from the associated applications of the original application and objects that are newly added to an instance (if any).
3. Click **Add**. Scroll to the end of the page, if needed.

   The **Add an object** dialog opens.
4. Select the target application for your object from the **Select application** drop-down list. For standalone applications, the drop-down list will have a single value and will be protected.
5. Select how you want to add the new instance object:

   DDL

   Specify the DDL of the new object that you want to add. The requested fields will differ based on the object type and template.

   Note: If the super administrator has selected the **Do not use templates** check box for any object type in the **DDL templates** page, then the legacy workflow will be used.

   a. **Object type**

      Select the type of the object that you want to add.

      Select template

      Select a template for the object creation that you want to add. For details on object creation, refer to "Adding Db2 object templates" on page 183.

   b. Enter a name in the **Object Name** field, and then select the values for other global variable fields, which appear based on the values that you select in the **Object type** and **Select template** fields. This can also span multiple pages, depending on the object type.

      Object name

      Enter a descriptive name for this object.

      Note: The object name is case-sensitive, with uppercase being the default. If you want to retain the letter case of your choice for the object name or include any special characters (such as # in the object name), then enclose the object name in double quotation marks. If the object name with a special character is not enclosed in double quotation marks, a syntax error is displayed, and the letters of the object name are changed to uppercase letters.

      Object qualifier

      Select an object qualifier from the drop-down list, which is in source-instance qualifier format.

Note: If you want to create a new source qualifier, click the **Create New** button. Specify a new source qualifier for your object, and then click **Validate**. The instance qualifier is generated based on the provisioning rules.

Database name
Select a name for the database from the drop-down list, which is in source-instance database format.

Note: If you want to create a new source database, click the **Create New** button. Specify a new source name for your database, and then click **Validate**. The instance database is generated based on the provisioning rules.

Tablespace name
Select a name for the tablespace.

Note: If you want to create a new tablespace, click the **Create New** button. Specify a new name for your tablespace, and then click **Validate**. The **Create New** button is enabled only when you specify the value in the **Database name** field.

Note: The global and local variable fields vary depending on the **Object Type** that you select. For example, some of the global variable fields are version and specific name, and local variable fields are tbQualifier, tbName, correlationNameForOld, seqQualifier, etc.

An existing target object with an existing instance qualifier
Specify an existing object that resides in the target subsystem.

An existing object from source
Specify an object that resides in the source subsystem.

6. Click **Next** after entering the global and local variables on multiple pages, which vary depending on the **Object Type**.

   - The system validates both variables for the selected template before landing on the **Edit DDL** page.

     Note: If **Do not use templates** check box is selected for any object type under the **DDL templates**, or if the template is not available, then the legacy workflow will be used.

   - If you have selected an option other than **DDL** in step 5, the list of objects that are required for the new object is displayed.

7. Review this list of required objects and click **Add new object**.

   The new object will be added to the instance.

Managing pull requests

Pull requests are changes made to a provisioned application instance. The reviewers review the changes to be merged into the originating application. At least one administrator of the team that owns the application must approve the pull request before it can be merged into the originating application.

All pull requests, except the ones with the status as `Merged` or `Declined` are displayed on the **Dashboard**. To see the current status of a pull request, refresh the **Dashboard**. The status of the `Merging` and `Updating` pull requests is automatically refreshed on the **Dashboard**.

The following users can review, approve, and merge pull requests according to their access rights and privileges:

Submitter
A *submitter* is the instance owner or the administrator of the team that owns the application. The submitter creates the pull request and submits it for review and approval. The submitter can add additional reviewers from the members of that team.

Additional reviewer
An *additional reviewer* is a team member, who is added as a reviewer by the submitter. The additional reviewers must be members of the team that owns the application. The additional reviewers can review and approve the pull requests. The instance owner or the administrator of the team that owns the application can merge these pull requests only after the team administrator has approved them.

Reviewer

The additional reviewers and the team administrators are collectively called *reviewers*. The default reviewers for a pull request are the administrator of the team that owns the application.

Submitting pull requests

When all the work related to an application instance is completed, you can request that your changes be merged to the originating application by submitting a pull request.

Before you begin

Note the following restrictions on submitting pull requests:

Restrictions:

- The instance owner can create only one pull request for an instance at a time.
- The instance owner cannot make additional changes to the object definitions while there is an active pull request for that instance.

Procedure

To submit a pull request, complete the following steps:

1. On the navigation menu, select **Explore**.
2. Click **Instances**.
3. On the **Instances** page, click **Details** on the instance for which you want to submit a pull request.
4. On the **Instance details** page, click **Create pull request**.
5. On the **Pull request** page, specify the following fields:

   Title

   Enter a name that identifies this pull request.

   Comment

   Enter a comment for this pull request as necessary. This field is optional.

   Additional reviewers

   In addition to the team administrators of the application owner team, who are by default pull request reviewers, you can optionally add other reviewers from the application owner team. This field is optional.
6. On the lower part of the page, click each object name and verify their definition changes on the **Diff** tab.
7. To see the site rule violations in the changes that you made, click the **Triggered site rules** tab, and then click the name of the object.

   The DDL of the object is displayed on the page. The ⊗ icon indicates a violation and the ⚠ icon indicates a warning.
8. Resolve the violations and the warnings.
9. Optional: Click the **Preview changes** button to review an impact report of this pull request for application or associated applications.
10. If you are good with all the changes, go back to the **Pull request** page, and click **Submit pull request**.

    The pull request is submitted to the reviewers. Your open pull requests are also displayed on the **Dashboard** page.
11. Optional: To view your pull request, click **Go to pull request** on the **Instance details** page.

What to do next

Your pull request must be reviewed and approved by at least one team administrator of the team that owns the originating application of the provisioned instance.

Reviewing and approving pull requests

For the object changes in an application instance to be reflected in the application or associated applications, the reviewers must first review and approve the pull request for the changes. At least one administrator of the team that owns the application must approve the pull request before it can be merged into the application or associated applications.

About this task

When a pull request is submitted, the reviewers are notified of the request. They can review the changes to decide whether to approve or decline the pull request. If there is a problem, the reviewer can either decline or return the pull request to the submitter. If the reviewer is good with the changes, the reviewer approves the pull request. The team administrator reviews and approves the changes to be merged into the application.

Procedure

To review and approve application instance objects, complete the following steps:

1. Open the **Dashboard** page by clicking the **Dashboard** link on the upper left corner of every page.
2. From **Pull request to review**, click the one that you want to review.

   The page for the selected pull request opens.
3. Click the **Diff** tab, and review what has been changed in the source code.
4. To see the site rule violations in the changes that you made, click the **Triggered site rules** tab, and then click the name of the object.

   The DDL of the object is displayed on the page. The ❌ icon indicates a violation and the ⚠ icon indicates a warning.

   Important: You can approve a pull request only after resolving the violations.
5. Click the **Comments** tab, enter your review comments, and click **Add comment**.
6. Optional: Click the **Preview changes** button to review an impact report for this pull request for current application or associated applications.
7. Depending on how you want to handle this pull request, click one of the following buttons:

Button	Action
Sync	The submitter or the team administrator can synchronize the pull request with the application or associated applications. This will get all the changes from the application or associated applications into the instance. This option is available only when the instance is not at the latest level with respect to the application or associated applications. **Note:** Synchronizing the pull request might create merge conflicts. In such a case, you must decline the pull request, resolve the conflicts, and create a new pull request.
Approve	The reviewer can approve a pull request. Approving a pull request lets the submitter know that the changes have been reviewed and the work can be merged with the originating application or associated applications.
Needs Work	If the pull request cannot be merged to the originating application or associated applications in its current state, the reviewer can return the pull request accompanied by a comment that describes the changes that the author must make before the reviewer can approve the request. To make the changes, the submitter must decline the pull request, which is already marked as **Needs work**, and then make the required changes. After the changes are done, the submitter must create and submit a new pull request.

Button	Action
Decline	If the pull request cannot be merged in its current state, the reviewer can decline the pull request. **Note:** Declining a pull request cannot be undone. If the reviewers decline a pull request, the submitter of the request must open a new pull request to have the code for the same instance merged to the originating application or associated applications.

Merging pull requests

A pull request becomes ready-to-merge state when at least one team administrator has approved it. Either the user who edited instance objects or their team administrators can merge all the changes that have been made to the application instance into the originating application. When a pull request is merged for an application, all other instances of that application are not at the latest level. If an application has an associated application and a pull request is merged to the associated application, all instances for the associated application as well as the parent application are marked as not at the latest level. You cannot merge a pull request when the instance is not at the latest level. To merge the pull request, you must synchronize the pull request.

Procedure

To merge a pull request, complete the following steps:

1. Open **Dashboard**.
2. From **Pull request to review**, click the one that you want to review.

   The page for the selected pull request opens.
3. Click the **Diff** tab, and review what has been changed in the source code.
4. To see the site rule violations in the changes that you made, click the **Triggered site rules** tab, and then click the name of the object.

   The DDL of the object is displayed on the page. The ❌ icon indicates a violation and the ⚠ icon indicates a warning.

   Important: You can merge a pull request only after resolving the violations.
5. Click the **Comments** tab, enter your review comments, and click **Add comment**.
6. Depending on how you want to handle this pull request, click one of the following buttons:

Button	Action
Sync	The submitter or the team administrator can synchronize the pull request with the application or associated applications. This will get all the changes from the application or associated applications into the instance. This option is available only when the instance is not at the latest level with respect to the application. **Note:** You cannot merge a pull request when there are merge conflicts. In such a case, you must decline the pull request, resolve the conflicts, and create a new pull request.
Merge	The submitter or the team administrator can merge this instance into the originating application or associated applications. The results of the merge process will be displayed in a message. A pull request can be merged only after the team administrator has approved the pull request and the latest changes from the application are synced into the instance from the application or associated applications.

Button	Action
Decline	If the pull request cannot be merged in its current state, the reviewer can decline the pull request. **Note:** Declining a pull request cannot be undone. If the reviewers decline a pull request, the submitter of the request must open a new pull request to have the code for the same instance merged to the originating application.
Resume Merge	The submitter or the team administrator can resume the merge from the previous failed point. While merging a pull request, applying object changes might fail due to unavailability of the resource when some other utilities are already using the object. As a result, the instance goes into an apply error state. To resume from the previous failed point, leverage the functionality of resume merge. If you click **Resume Merge**, you might lose data.
Revert Merge	The submitter or the team administrator can revert a pull request or instance changes from the application or associated applications. While merging a pull request, applying object changes might fail due to unavailability of the resource when some other utilities are already using the object under consideration. As a result, the instance goes into an apply error state. You can revert the pull request or instance changes from the application or associated applications. The changes will not go in the application or associated applications but stay in the instance itself. If you click **Revert Merge**, you might lose data.

7. If the merge process ends successfully, you can view the updated status of the pull request in the **Dashboard** page.

   - If you are the pull request submitter, click the **View your pull requests** link in the **Open pull requests** widget.
   - If you are the team administrator, click the **View all pull requests to review** link in the **Pull requests to review** widget.

   The status of the pull request will be shown as "MERGED".

Deprovisioning instances

You can deprovision an instance of an application if it becomes no longer needed.

Before you begin

To be able to deprovision an instance of an application, there should be no outstanding pull requests for that instance.

Procedure

To deprovision an instance, complete the following steps:

1. On the navigation menu, select **Explore**.
2. Click **Instances**.

   The **Instances** page opens.
3. On the instance that you want to deprovision, click the overflow menu and select **Deprovision**.

   A confirmation message is displayed.
4. Click **Deprovision**.

The instance is removed from the **Instances** page and moved to the `policy\archive\instances` folder.

Managing Db2 object templates

A super administrator can create one or more Db2 templates and their links to different object types using the DDL template option. However, all the other users can use the DDL template option to view the templates in read-only mode.

The templates that a super administrator creates will populate in the Select template drop-down for object creation. These templates are categorized into two sources: Site and Product.

- The custom templates that are created by the user are listed as Site source. You can either edit or delete these templates.
- The default templates that are available with the product are listed as sample templates for Product source. You can only view these templates and not modify them.

Adding Db2 object templates

In Db2 DevOps Experience, a super administrator can add or edit object templates for different object types. An object template can contain only one DDL Statement, which is CREATE or CREATE/REPLACE. However, all other users can only view these templates and select them while adding objects to an instance.

Before you begin

- The default templates are shipped in the `product` directory and will be used unless optionally overridden or ignored in the `site` directory. The following object template type directories are available under the Product source:

Table 51. Default Db2 object template directories

Directory name	Description
AL	Alias
DB	Database
IX	Index
SA	Sequence Alias
SEQ	Sequence
SP	Stored Procedure
SY	Synonym
TB	Table
TG	Trigger
TS	Table Space
UDF	User-Defined Function
UDT	User-Defined Type
VW	View

- If the name of the user created template (custom) and the default template that is shared with the product have the same file name, then the custom template takes precedence.
- A super administrator can prevent any users from using templates for a specific object type by selecting the **Do not use templates** check box for object type on the DDL templates page.

- Any users other than the super administrator can only view the templates.

Procedure

To add a template for Db2 object types, complete the following steps:

1. On the navigation menu, click **Manage**, and then **DDL templates**.

   The DDL templates page opens. This page shows you the list of object types.
2. Click **Add new template** to create a new template for an object type.
3. Enter a name for the object template in the **Name** field.
4. In the **Template content** pane, paste your template content or write your template DDL.

   To add new variables to the template, place the cursor on the template pane and do any of the following:

   a) Click **Insert user variable**.

   b) Enter a value for the user variable and then click **Insert**. The user variable is added to the template.

   Or

   a) Click **Insert global variable**.

   b) Select the global variables from the drop-down and then click **Insert**. The global variable is added to the template.
5. Click **Add**.

Results

After validating the template details, a new custom template is added to the object type.

Editing Db2 object templates

A super administrator can edit object templates and their links to different object types. However, all the other users can only view the templates in read-only mode.

Before you begin

- A super administrator can only edit custom templates that are listed for Site source. The default templates that are available with the product and are listed as sample templates for Product source are not editable.

Procedure

To edit an existing template of Db2 object type, complete the following steps:

1. On the navigation menu, click **Manage**, and then **DDL templates**.

   The DDL templates page opens. This page shows you the list of object types and the number of templates that are available for a specific object type.
2. Expand an object type to view the templates and sub-templates that are available in an object type.

   Notes: The two types of source templates are Site and Product.

   - The custom templates that are created by the user are listed as Site source. You can either edit or delete these templates.
   - The default templates that are available with the product are listed as sample templates for Product source. You can only view these templates and cannot modify them.
3. Click ⋮ and then select **Edit** or click a site template. The **Edit template** page is displayed.
4. Modify the template fields that include **Name**, **Template content**, and **Link to templates**.

Note: The **Insert user variable** and **Insert global variable** buttons are enabled only when you place the cursor in the **Template content** pane. You can use these options to edit both global and user variables.

5. To duplicate the template, perform the following:
   a) Click the **Duplicate template** button.

      Note: This button is available only for custom (site) templates.

   b) Enter a new name for the template in the **Template name** field.

   c) Click **Add**. A message is displayed that the template is duplicated successfully. This includes the template as well as their links to different object types.
6. Click **Update**.

Results

The custom template of an object type is edited or duplicated successfully.

Deleting Db2 object templates

A super administrator can delete object templates and their links to different object types. However, all the other users can only view the templates in read-only mode.

Before you begin

- A super administrator can only delete custom templates that are listed for Site source. The default templates that are available with the product and are listed as sample templates for Product source cannot be deleted.

Procedure

To delete a template of Db2 object type, complete the following steps:

1. On the navigation menu, click **Manage**, and then **DDL templates**.

   The DDL templates page opens. This page shows you the list of object types and the number of templates that are available for a specific object type.
2. Expand an object type to view the templates and sub-templates that are available in an object type.
3. Click ⋮ and then select **Delete**. The Delete template dialog box is displayed.
4. Click **Delete**.

Results

The template of an object type is deleted successfully.

Linking to Db2 object templates

A super administrator can add object templates and link them to other object templates. The Link to template button is available only for custom (site) templates.

Procedure

To link Db2 template to another Db2 template, complete the following steps:

1. On the navigation menu, click **Manage**, and then **DDL templates**.

   The DDL templates page opens. This page shows you the list of object types and the number of templates that are available for a specific object type.
2. Expand an object type to view the templates and sub-templates that are available in an object type.
3. Click ⋮ and then select **Edit** or click a site template. The **Edit template** page is displayed.

4. Click the **Link to template** button.
5. Select a template from the **Select link to templates** drop-down.
6. Click **Submit**.

   Note: You can delete the template from the linked templates list by clicking ⊖.

Results

The template is linked to other object templates successfully.

Chapter 10. Messages

These topics provide reference information for Unified Management Server and data management product messages.

Message format

Messages issued by IBM Unified Management Server for z/OS and any activated data management product adhere to the following format:

```
IZPxxnnny
```

IZP

Indicates that the message was issued by IBM Unified Management Server for z/OS or any activated data management product.

xx

Identifies the type of the message:

DB

Indicates that the message was issued during Db2 application provisioning.

DC

Indicates that the message was issued when you were working with Db2 commands.

DI

Indicates that the message was issued during IMS subsystem discovery.

DS

Indicates that the message was issued during Db2 subsystem discovery.

DZ

Indicates that the message was issued during subsystem discovery when an error that is not associated with any specific subsystem occurred.

DT

Indicates that the message was issued during Db2 object template creation.

FL

Indicates that the message was issued while saving filters.

GN

Indicates that the message was issued during site rule creation.

GQ

Indicates that the message was issued while processing `graphql` queries.

LG

Indicates that the message was issued when you specified the log level.

MS

Indicates that the message was issued when you requested an action for an IMS subsystem or an IMS subsystem environment.

PI

Indicates that the message was issued when you ran the post-installation script.

PL

Indicates that the messages are issued when you manage teams, environments, subsystems, applications, instances, and so on.

SC

Indicates that the message is associated with security and authorities.

SV

Indicates that the message is associated with the IBM Unified Management Server for z/OS.

TP

Indicates that the messages are issued by a third-party application while performing operations.

UU

Indicates that the message was issued by IBM Unified Management Server for z/OS plug-in for Zowe.

nnnn* or *nnn

Indicates the message identification number.

y

Indicates the severity of the message:

E

Indicates that an error occurred, which might or might not require operator intervention.

I

Indicates that the message is informational only.

W

Indicates that the message is a warning to alert you to a possible error condition.

IZPDB messages

Messages that are issued during Db2 application provisioning have the format IZPDBnnnnx.

IZPDB0039E Could not connect to the subsystem '*%1$s*' at URL '*%2$s*' with port '*%3$s*' and location '*%4$s*': Make sure hostname, port number, and location are correct. For encrypted connections, make sure the certificate for subsystem '*%1$s*' has been imported and SSL/TLS has been correctly configured. (SSL Exception: '*%5$s*').

Explanation:
A secured port that is used to connect to Db2 cannot establish a connection with the URL '*%2$s*' and the location '*%4$s*' because either the Db2 certificate is not imported in UMS, SSL/TLS is not correctly configured in Db2, or the subsystem location is not correct. For information about when this error is displayed, see Support for Db2 subsystems with encrypted JDBC connections.

System action:
Db2 rejects the connection.

User response:
Ask the system administrator to check if the certificate for the subsystem is imported and ensure that SSL/TLS of Db2 is correctly configured. Also ensure the hostname, port number, and subsystem location are correct.

IZPDB0046E The location '*%1$s*' provided for subsystem '*%2$s*' is incorrect. Make sure all parameters are correctly configured.

Explanation:
A secured port that is used to connect to Db2 cannot establish a connection with the URL '*%2$s*' and the location '*%4$s*' because the location provided for the subsystem is incorrect.

System action:
Db2 rejects the connection.

User response:
Ask the system administrator to check if all parameters are correctly configured.

IZPDB0047E The following fields: '%1$s' are blank in '%2$s'. Those fields must have a value.

Explanation:
The following fields are missing in the IZPDB2PM dataset: IZP_DB2_ADB_HLQ, IZP_DB2_ADB_PREFIX, IZP_DB2_GOC_HLQ, and IZP_DB2_GOC_PREFIX.

System action:
These parameters are used to locate Db2 administration packages in the system. If these parameters are not specified, the system cannot perform any Db2 administration tasks.

User response:
Add the missing fields in the IZPDB2PM dataset and restart the server to proceed further.

IZPDC messages

Messages that are issued when you work with Db2 commands have the format IZPDCnnnnx.

IZPDC0001E ***variable1* is mandatory. It cannot be null or empty.**

Explanation:
The user has not specified a value for *variable1*. This parameter cannot be empty.

System action:
The command or request is not processed.

User response:
Specify a value for *variable1* and try again.

IZPDC0002E ***variable1* is invalid commandId.**

Explanation:
The value specified in *variable1* is an invalid Db2 command ID.

System action:
The command cannot run because the command ID is invalid.

User response:
Specify a valid command ID for *variable1* and try again. `DISPLAY DATABASE` is an example of a valid command ID.

IZPDC0003E ***variable1* is invalid parameters.**

Explanation:
The value specified in *variable1* is an invalid Db2 parameter.

System action:
The request cannot be processed because *variable1* is an invalid Db2 parameter.

User response:
Specify a valid value for *variable1* and try again. `DATABASE` and `SPACENAM` are examples of valid parameters.

IZPDC0004E ***variable1* is invalid option.**

Explanation:
The user specified an invalid value for the Db2 command.

System action:
The request is not processed because the user specified an invalid value for the Db2 command.

User response:
Specify a valid value for the Db2 command. `CLAIMERS`, `LOCKS`, and `USE` are examples of valid values for the Db2 command.

IZPDC0005E ***variable1* is invalid.**

Explanation:
The request body is empty or null.

System action:
The request cannot be processed because the request body is empty or null.

User response:
Specify a valid request body and try again.

IZPDC0006E **Failed to connect with *variable1*.**

Explanation:
A connection could not be established with *variable1*.

System action:
The request cannot be processed because a connection could not be established with *variable1*.

User response:
Check the database connection and try again.

IZPDC0007E **Internal Server Error**

Explanation:
An internal error occurred in UMS.

System action:
UMS will not process this task or request.

User response:
See details of the error and contact IBM Software Support.

IZPDI messages

Messages that are issued during IMS subsystem discovery have the format IZPDInnnx.

IZPDI004W **Type-2 command failed: CMD='*cmd*' RC=*rc* RSN=*rsn*. *message_id message_text*.**

Explanation:
The IMS type-2 command *cmd* failed with a return code of *rc* and a reason code of *rsn*. The return and reason codes are followed by the message ID and the message text that are returned from the CSL.

System action:
Processing continues.

User response:

For an explanation of the return code (*rc*) and the reason code (*rsn*), see the "CSLOMCMD: command request" topic in the *IMS System Programming APIs*. If you cannot identify the cause of the command failure or if the problem persists, contact IBM Software Support.

IZPDI005W **Malformed type-2 XML document. Element '*element_name*' not found.**

Explanation:
The XML response returned from an IMS CSL type-2 command is missing required element '*element_name*'.

System action:
Processing of the type-2 response is skipped, and discovery processing continues.

User response:
See the product documentation to enable diagnostic tracing. If the problem persists, contact IBM Software Support.

IZPDI007W **Multiple errors or warnings were returned.**

Explanation:
Multiple errors or warnings were returned during IMS subsystem discovery processing. Only partial results might be returned.

System action:
Processing continues.

User response:
Check the log for any preceding error messages.

IZPDI010E **IMS module load request failed: Module=*module_name* ABEND=*xxx-yy* IMSplex=*imsplex_name***

Explanation

IMS subsystem discovery failed to load an IMS module, either CSLSRG00 or CSLSDR00. These modules are required for the discovery to issue type-2 commands. The abend code (*xxx-yy*) provides the reason for the LOAD request.

In the message text,

- *module_name*: Module name (CSLSRG00 or CSLSDR00)
- *xxx-yy*: Abend code
  - *xxx*: System completion code
  - *yy*: Reason code
- *imsplex_name*: IMSplex name

A typical problem is ABEND=806-04, which indicates that the module could not be found. IMS subsystem discovery attempts to load the module from the LINKLIST, or if not found, the data sets specified by the IMS control region's STEPLIB and DFSRESLB DD statements.

System action:
The IMS type-2 commands are not issued for the IMS regions in the IMSplex. The regions are discovered but with missing information.

User response:
Contact IBM Software Support.

IZPDI011E **IMS discovery result truncated due to buffer constraint: Size=*nnn*MB**

Explanation:
The result buffer is too small for the IMS subsystem discovery response payload.

System action:
IMS subsystem discovery processing stops.

User response:
Contact IBM Software Support.

IZPDI012E **IMS discovery has abended: Function=*func_name* ABEND=*xxxxxxxx-yyyyyyyy* PSW=*pppppppp-qqqqqqqq* EP=*eeeeeeee* SSCDIMID=*imsid* MODID=*module_id***

Explanation

IMS subsystem discovery ended abnormally while processing the IMS subsystem indicated by *imsid*.

In the message text,

- *func_name*: Function name
- *xxxxxxxx-yyyyyyyy*: Abend code
  - *xxxxxxxx*: System completion code
  - *yyyyyyyy*: Reason code
- *pppppppp-qqqqqqqq*: PSW address
- *eeeeeeee*: Entry point address
- *imsid*: IMS ID (SSID)
- *module_id*: Module ID

System action:
The region is discovered but with missing information. Discovery continues to look for other regions.

User response:
Contact IBM Software Support.

IZPDI013W **IMS discovery did not find the region: SSID=*ssid* LPAR=*lpar_name* SYSPLEX=*sysplex_name***

Explanation

IMS subsystem discovery could not locate the requested IMS SSID on this system. If the region was previously discovered successfully, it might have been stopped.

In the message text,

- *ssid*: IMS ID (SSID)
- *lpar_name*: LPAR name
- *sysplex_name*: Sysplex name

System action:
IMS subsystem discovery processing ends normally, with no IMS subsystems being discovered.

User response:
Check to see if the region has been stopped. Retry the request after the region is restarted.

IZPDI014W **IMS discovery did not find any regions: LPAR=*lpar_name* SYSPLEX=*sysplex_name***

Explanation

IMS subsystem discovery did not locate any IMS regions on this system.

In the message text,

- *lpar_name*: LPAR name
- *sysplex_name*: Sysplex name

System action:
IMS subsystem discovery processing ends normally, with no IMS subsystems being discovered.

User response:
Check to see if the IMS regions on this system have been stopped. Retry the request after the regions are restarted.

IZPDI015E **IMS region SWA control block was not found: R15=*rc* SVA=*ssssss* CB=*cccc***

Explanation

IMS subsystem discovery failed to locate an SWA control block in either the IMS control region or the DBRC region while looking for data set information.

In the message text,

- *rc*: Return code (R15)
- *ssssss*: Shared virtual area (SVA)
- *cccc*: Control block

System action:
The region is discovered but with missing information.

User response:
Contact IBM Software Support.

IZPDI016E **Cross-memory access to IMS region failed: R15=*rc* Function=*func_name***

Explanation

IMS subsystem discovery failed to obtain cross-memory access to the IMS region. Control blocks in the private storage of the IMS control region and the DBRC region are used to obtain information.

In the message text,

- *rc*: Return code (R15)
- *func_name*: Function name

System action:
The region is discovered but with missing information.

User response:
Contact IBM Software Support.

IZPDI017E **Dynamic allocation of IMS library failed: EC=*err_code* IC=*info_code* IMS=*ssid* DSN=*dsn***

Explanation

IMS subsystem discovery failed to allocate the IMS data set dynamically. The error and information codes describe the error. Additional messages, returned by dynamic allocation and prefixed with IKJ, are issued with this message.

In the message text,

- *err_code*: Error code
- *info_code*: Information code
- *ssid*: IMS ID (SSID)
- *dsn*: Data set name

System action:
The region is discovered but with missing information.

User response:
Contact IBM Software Support.

IZPDI018E **IMS library open request failed: ABEND=*xxx-yy* IMS=*ssid* DSN=*dsn***

Explanation

IMS subsystem discovery failed to open the IMS data set. The return and reason codes (*xxx-yy*) provide the reason for the failing OPEN request.

In the message text,

- *xxx-yy*: Abend code
  - *xxx*: System completion code
  - *yy*: Reason code
- *ssid*: IMS ID (SSID)

- *dsn*: Data set name

A typical problem is ABEND=913-38, which indicates that the discovery does not have security server permission to read the data set.

System action:
The region is discovered but with missing information.

User response:
If the problem is related to data set security, contact your security administrator. Otherwise, contact IBM Software Support.

IZPDI019E **IMS type 2 command error; Service=*macro_name* R15=*rc* R0=*rsn* IMSplex=*imsplex_name* USERID=*userid***

Explanation

IMS subsystem discovery issued a type-2 command to obtain information about the region, but the request failed. Additional messages, returned by IMS and prefixed with CSL, are issued with this message.

In the message text,

- *macro_name*: Failing macro name
- *rc*: Return code (R15)
- *rsn*: Reason code (R0)
- *imsplex_name*: IMSplex name
- *userid*: User ID

System action:
The region is discovered but with missing information.

User response:
For more information about the additional CSL messages that accompany this message, see the "CSL messages" topic in *IMS Messages and Codes*.

IZPDI023E **Started Task information in the JES spool could not be interpreted: DDNAME=<ddname> JOBNAME=<jobname> JOBID=<jobid> RECORD=<spool data>**

Explanation

The JES spool output for the reported IMS region could not be interpreted. The failing record is reported. In JESJCLIN, a statement similar to the following is expected://STARTING EXEC IMSPROD1,PARM1='AUTO=N'.

In JESYSMSG, a message similar to the following is expected: 2 IEFC001I PROCEDURE IMSPROD1 WAS EXPANDED USING SYSTEM LIBRARY IMSPROD.IMS.PROCLIB

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
Contact IBM Software Support.

IZPDI024E **Started Task information is missing in the JES spool: JOBNAME=<jobname> JOBID=<jobid> REASON=<reason_number>**

Explanation

The JES spool output that describes how the reported IMS region is started could not be located. In JESJCLIN, a DD statement similar to the following is expected: //STARTING EXEC IMSPROD1,PARM1='AUTO=N'.

In JESYSMSG, a message similar to the following is expected: 2 IEFC001I PROCEDURE IMSPROD1 WAS EXPANDED USING SYSTEM LIBRARY IMSPROD.IMS.PROCLIB. This message is expected when the IMS region is not a started task.

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
Contact IBM Software Support.

IZPDI025E **JES STATUS request failed: FUNCTION=SSST(80) STATTYPE=<request type> RC=<code> SSOBRETN=<code> STATREAS=<code> JOBNAME=<jobname> JOBID=<jobid>**

Explanation:
The subsystem interface extended status function call SSSI(80) was used to retrieve status information about the reported IMS region. The request failed with the reported return and reason codes. Details about this service, including return codes, are available at Extended status function call — SSI function code 80.

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
Contact IBM Software Support.

IZPDI026E **JES STATUS request failed: JOBNAME=<jobname> JOBID=<jobid> was not found**

Explanation:
The subsystem interface extended status function call SSSI(80) was used to retrieve status information about

the reported IMS region. The reported IMS region job name and job ID was not found.

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
If the IMS region has been purged, then run discovery again after the region is restarted. Otherwise, contact IBM Software Support.

IZPDI027E **JES SPOOL request failed: FUNCTION=SSJI(71) SSJIFREQ=<function> RC=<code> SSOBRETN=<code> SSJIRETN=<code> JOBNAME=<jobname> JOBID=<jobid> DDNAME=<ddname> <reason>**

Explanation:
The subsystem interface JES job information services call `SSJI(71)` was used to read spool output for the reported IMS region. The request failed with the reported return and reason codes. A reason of "ACCESS DENIED" is reported when the failure is caused by security authorization checking. Details about this service, including return codes, are available at SPOOL Read Service.

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
If the reason is "ACCESS DENIED" then grant the discovery user ID access to the IMS region spool output and retry the request. Otherwise, contact IBM Software Support.

IZPDI028E **Control Block returned by JES request was not recognized: BLOCK=<name> VALUE=<hex string> JOBNAME=<jobname> JOBID=<jobid> DDNAME=<ddname>**

Explanation:
The contents of a JES control block, returned by the subsystem interface, were not recognized. BLOCK is the name of the control block being examined. VALUE is the 4 bytes hexadecimal in the block that was not recognized.

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
Contact IBM Software Support.

IZPDI029E **JES subsystem is not supported: SSID=<name>**

Explanation:
The primary JES subsystem is not JES2. Only JES2 is supported for discovering how an IMS region is started.

System action:
Information about how to start the IMS region is not discovered. The discovery process continues.

User response:
Contact IBM Software Support.

IZPDS messages

Messages that are issued during Db2 subsystem discovery have the format `IZPDSnnnnx`.

IZPDS0001E **We could not find any subsystem matching id: *variable1***

Explanation:
The subsystem with the provided ID cannot be found in the UMS.

System action:
The UMS system will not start discovery for the subsystem with the provided ID.

User response:
Check if the subsystem ID exists.

IZPDS0001W **Error during initial system discovery. Retrying in *<number>* seconds, next warning in *<number>* seconds.**

Explanation:
Zowe System Services (ZSS) server process was not available at UMS startup. Discovery and other services may not be available until they start up.

System action:
UMS will continue to poll for ZSS availability at intervals.

User response:
If ZSS becomes available before UMS has polled for it, the user may need to refresh data on the Manage Subsystems pages as well as manually initiate infrastructure discovery (**Discovered tab** on the **Manage Subsystems** page) or object discovery on subsystems (per-subsystem menu item).

IZPDS0002E **The received id: *variable1* is not well formatted**

Explanation:
The subsystem ID provided is not formatted correctly.

System action:
The UMS system will not start discovery for the subsystem with the provided ID.

User response:
Check if the subsystem ID is correct.

IZPDS0003E The id cannot be blank

Explanation:
The subsystem ID is blank.

System action:
The UMS system will not perform any discovery.

User response:
Retry with a correct subsystem ID.

IZPDS0005E Could not find discovery plan for this discovery session.

Explanation:
No discovery plan is found for this discovery session.

System action:
The UMS system will not perform any discovery plan.

User response:
Ensure that the discovery plan is created for the discovery session.

IZPDS0006E Could not find zss service for discovery.

Explanation:
The ZSS service is not available.

System action:
The UMS system cannot complete subsystem discovery.

User response:
Ensure that the ZSS service is available.

IZPDS0007E zss error occurred: *variable1*

Explanation:
A ZSS service error occurred because of the reason shown in the message.

System action:
The UMS system cannot complete discovery.

User response:
Ensure that the ZSS service is available and confirm that the password has not expired for the current user.

IZPDS0100E The dataset *variable1* does not exist.

Explanation:
The data set *variable1* does not exist.

System action:
The tools discovery process skips the possible YAML file location *variable1* and continues with the process.

User response:
Check that the data set *variable1* exists.

IZPDS0101E The UNIX System Services file *variable1* does not exist.

Explanation:
The UNIX System Services file *variable1* does not exist.

System action:
The tools discovery process skips the possible YAML file location *variable1* and continues with the process.

User response:
Check that the UNIX System Services file *variable1* exists.

IZPDS0102E The yaml *variable1* does not contain or has a null value for the following key: *variable2*.

Explanation:
The key *variable2* within the YAML file *variable1* does not exist or contains a null value.

System action:
The YAML file *variable1* is no longer processed, and the tools discovery process continues.

User response:
Check that YAML file *variable1* contains the key *variable2* with a non-null value.

IZPDS0103E The prefix *variable1* for path *variable2* is not supported.

Explanation:
A prefix of a path is used to assist in the discovery of the file. The prefix *variable1* specified is not supported.

System action:
The tools discovery process skips the path *variable2* as a possible discovery location.

User response:
Change the prefix *variable1* to a valid value, which includes DSN and DIR.

IZPDZ messages

Messages that are issued during discovery of any subsystem type have the format IZPDZnnnnx.

IZPDZ0001E **ZSS or Cross-Memory Server internal error - *text*.**

Explanation:
An internal error occurred in the ZSS or the Zowe cross-memory server. The reason of the error is indicated by variable *text*.

System action:
Processing stops.

User response:
Contact IBM Software Support.

IZPDZ0002E **ZSS or Cross-Memory Server internal error - *text*. *additional_info1* *additional_info2* ...**

Explanation:
An internal error occurred in the ZSS or the Zowe cross-memory server. The reason of the error is indicated by variable *text*. The message might be followed by additional diagnostic information such as the subsystem name and the internal return and reason codes.

System action:
Processing stops.

User response:
Contact IBM Software Support.

IZPDT messages

Indicates that the message was issued during Db2 object template creation.

IZPDT0001E **The template '*%1$s*' could not be processed for the following reason: '*%2$s*'**

Explanation:
List of template processing errors.

System action:
No further operations can be performed on the template.

User response:
Verify the template content and resolve the processing errors.

IZPDT0002E **The template '*%1$s*' of type '*%2$s*' could not be built because the following variables are not defined: '*%3$s*'**

Explanation:
User has not provided values for all the global and user variables in the template.

System action:
System will not be able to process the template and generate the DDLs.

User response:
Enter appropriate values for global and user variables.

IZPDT0003E **There is no DDL template for the Db2 object of type '*%1$s*'**

Explanation:
There is no DDL template for the input Db2 object.

System action:
System displays an error message if there is no template.

User response:
Enter a valid name and type to get the template.

IZPDT0004E **The template '*%1$s*' of type '*%2$s*' located in '*%3$s*' is incomplete, Please resolve one or more errors**

Explanation:
List of errors for the incomplete template.

System action:
System displays an error if the template is incomplete.

User response:
Resolve all the errors and try again.

IZPDT0005E **The template variable '*%1$s*' has not been assigned a value in the template '*%2$s*' of type '*%3$s*'**

Explanation:
Values are not provided for the variables.

System action:
System displays an error message due to missing variables.

User response:
Update the missing values for the variables and try again.

IZPDT0006E **There is no content in the template '*%1$s*' of type '*%2$s*'**

Explanation:
The template has been found but with empty content.

System action:
System displays an error message due to empty content in the template.

User response:
Check and update the template content.

IZPDT0007E **Unsupported global variable : '*%1$s*'. Applicable global variables : '*%2$s*'**

Explanation:
The template contains unsupported global variables.

System action:
System displays an error message due to unsupported global variables in the template.

User response:
Enter supported values for the global variables.

IZPDT0008E **Variable has unsupported Prefix : '*%1$s*' Supported prefixes : '*%2$s*'**

Explanation:
Variable in the template has unsupported prefix. Currently, global and user prefixes are supported.

System action:
System displays an error message due to unsupported prefixes in the template.

User response:
Use appropriate prefixes for global and user variables to proceed further.

IZPDT0009E **Invalid variable: '*%1$s*'. Variable shouldn't be empty or null and should not contains spaces**

Explanation:
The global and user variables should not be empty or contain any spaces.

System action:
System displays an error message due to invalid variables in the template.

User response:
Check and update the global and user variables.

IZPDT00010E **The template '*%1$s*' has syntax error: unsupported variable pattern format '*%2$s*'**

Explanation:
Unsupported pattern found in the template. The supported pattern is [(${GLOBAL.tsname})]

System action:
System displays an error message due to unsupported pattern in the template.

User response:
Check and update the pattern, such as [(${GLOBAL.tsname})] for global and [(${USER.bufferpool})] for user variables.

IZPDT00011E **The template '*%1$s*' has syntax error: unclosed brackets '*%2$s*'**

Explanation:
Unclosed parenthesis found in the template.

System action:
System displays an error message when unclosed parenthesis are found in the template.

User response:
Ensure the parenthesis are closed in the template.

IZPDT00012E **The template is missing the required global variables '*%1$s*'. The following global variables were found in the template : '*%2$s*'. The required global variables are '*%3$s*'.**

Explanation:
The template is missing the required global variables.

System action:
System displays an error message as the mandatory global variables are missing in the template.

User response:
Add the required global variables in order to proceed further.

IZPDT00013E **The provided request includes an unsupported Type Hierarchy, '*%1$s*' requires '*%2$s*'. Please adhere to the hierarchy guidelines, such as 'TB' requires 'TS' requires 'DB (or) 'TS' requires 'DB' (or) 'TB' requires 'DB'.**

Explanation:
The provided request includes an unsupported type of hierarchy. The hierarchy guidelines are 'TB' requires 'TS' requires 'DB (or) 'TS' requires 'DB' (or) 'TB' requires 'DB'.

System action:
System displays an error message if type hierarchy is not followed in the template folder.

User response:
Adhere to the type hierarchy rules for Db2 objects and the template folder structure.

IZPFL messages

Messages that are issued while saving filters have the format `IZPFLnnnnx`.

IZPFL0001E ***Variable1* with name *variable2* already exists. Specify a unique name.**

Explanation:
A filter with the name *variable2* already exists.

System action:
The filter cannot be created because a filter with the name *variable2* already exists.

User response:
Specify another name for the filter and save the filter.

IZPFL0002E ***Variable1* field is mandatory. It cannot be null or empty.**

Explanation:
variable1 is a mandatory field. You must specify a value for this field.

System action:
The request will not be processed because *variable1* is either empty or null.

User response:
Specify a value for *variable1* and try again.

IZPFL0003E ***variable1* is invalid. Usage: *variable2***

Explanation:
variable1 is invalid because it has exceeded its maximum limit of 32 characters.

System action:
The filter cannot be saved or updated because *variable1* has exceed its prescribed limit of 32 characters.

User response:
Specify a value for *variable1* that is less than 32 characters and save the filter again.

IZPFL0004E ***variable1* with name *variable2* does not exist. Specify a valid name.**

Explanation:
The filter cannot be updated or deleted because the filter with the name *variable2* does not exist.

System action:
The filter cannot be updated or deleted.

User response:
Specify a valid name in *variable2* and try again.

IZPFL0005E ***variable1* with id *variable2* does not exist. Specify a valid id.**

Explanation:
The filter cannot be updated or deleted because the filter with ID *variable2* does not exist.

System action:
The filter cannot be updated or deleted.

User response:
Specify a valid ID in *variable2* in the request URL and try again.

IZPFL0006E ***variable1* : *variable2* is/are not currently registered.**

Explanation:
You cannot save or update the filter because *variable 2* is not a registered subsystem.

System action:
The filter is not saved or updated.

User response:
Specify a registered subsystem in the request URL or contact the system administrator to register the subsystem you want to use.

IZPFL0007E ***variable1* : *variable2* is/are not allowed.**

Explanation:
You cannot create or update the filter because *variable 2* is not a valid object type for the filter.

System action:
The filter is not created or updated.

User response:
Specify a valid object type for *variable2* in the request body.

IZPGN messages

Messages that are issued during site rule creation have the format `IZPGNnnnnx`.

IZPGN0003E **Rule *pattern* of type *rule_type* cannot contain regex.**

Explanation

Naming rules of the given type are not allowed to contain the following symbols:

```
_ % #
```

System action:
The rule will not be created or updated.

User response:
Remove all of these symbols from the pattern.

IZPGN0007E **Rule *pattern* must contain regex character(s).**

Explanation

"Pattern-match" naming rules must contain at least one of the following symbols:

```
_ % #
```

System action:
The rule will not be created or updated.

User response:
Add at least one of these symbols to the pattern.

IZPGN0014E **Invalid rule *rule_pattern* of type *rule_type*. Invalid '.' usage.**

Explanation:
Naming rules of the given type cannot have multiple adjacent periods (..) or end with a period (.).

System action:
The rule will not be created or updated.

User response:
Ensure that the rule pattern does not have adjacent periods and does not end with a period.

IZPGQ messages

Messages that are issued while processing `graphql` queries.

IZPGQ001I **The `graphql` API timed out. Contact your administrator to increase the timeout limit.**

Explanation:
The `graphql` API returned an error because the timeout limit was exceeded. Change the value to a higher number to avoid timeout issues.

System action:
The request cannot be processed because the `graphql` API timeout limit was exceeded.

User response:
Navigate to the ZWEYAML PARMLIB member and edit the `components.izp.server.graphQLTimeout` parameter. Restart the UMS server.

IZPLG messages

Messages that are issued when specifying the log level have the format `IZPLGnnnnx`.

IZPLG0001E **Log level *variable1* is not supported.**

Explanation:
The input log level is not supported. Supported log levels are: SEVERE. WARNING, INFO, CONFIG, FINE, FINER, FINEST, ALL, and OFF.

System action:
The log level of UMS is not changed.

User response:
Use a supported log level to change the log level of UMS.

IZPMS messages

Messages that are issued when IMS processes are running have the format `IZPMSnnnnx`.

IZPMS0005E **Submitted job terminated - Name: *variable1*, Id: *variable2*.**

Explanation:
The submitted job terminated with a status other than "complete". For example, the job status can be `SYSTEM ABEND`. The job name is *variable1* and the job ID is *variable2*.

System action:
The operation could not go through because the job did not complete normally.

User response:
Contact your system administrator on why the job failed or correct the problems and retry.

IZPMS0006E **Submitted job terminated - Name: *variable1*, Id: *variable2*, Return code: *variable3*.**

Explanation:
The submitted job terminated with return code *variable3*. This error code does not permit the job processing to continue. The job name is *variable1* and the job ID is *variable2*.

System action:
The operation could not go through because the job did not complete normally.

User response:
Contact your system administrator on why the job failed or correct the problems and retry.

IZPMS0007E **SQL cannot be issued. Field *variable1* needs to be defined in subsystem *variable2*.**

Explanation:
The user tried to use the IMS SQL processor, but the required field *variable1* was not defined in the IMS subsystem *variable2*.

System action:
Requested SQL is not issued because the JDBC connection to the target IMS Connect cannot be established.

User response:
Contact your system administrator to define the required field in the IMS subsystem.

IZPMS0039E **Cannot change the RECON data set of subsystem '*subsystem_name*'.**

Explanation:
The user tried to update an IMS subsystem *subsystem_name* with different RECON data set names from the currently registered ones. RECON data set names of subsystem cannot be changed.

System action:
The operation is not performed.

User response:
Specify a correct set of RECON data set names and retry the operation. If any of the RECON data set names was actually changed, the subsystem must be registered again.

IZPMS0043I **The group is not implemented as an IMSplex.**

Explanation:
The user tried to register or update a data sharing group, but the group is not configured as an IMSplex. No IMS product that is activated on the Unified Management Server supports this type of IMS configuration.

System action:
The operation is not performed.

User response:
Refer to the IMS configuration requirements for the IMS products that are activated on the Unified Management Server.

IZPMS0044I **No Operations Manager for the subject IMSplex is found or active.**

Explanation:
The user tried to register or update a data sharing group, but no Operations Manager (OM) was found or active in the subject IMSplex. At least one OM member must be active in the IMSplex.

System action:
The operation is not performed.

User response:
Refer to the IMS configuration requirements for the IMS products that are activated on the Unified Management Server.

IZPMS0050I **More than one IMS catalog is used in the group.**

Explanation:
The user tried to register or update a data sharing group, but multiple IMS catalogs were used in the data sharing group. There must be one and only one IMS catalog that is shared by all IMS members in the data sharing group.

System action:
The operation is not performed.

User response:
Refer to the IMS configuration requirements for the IMS products that are activated on the Unified Management Server.

IZPMS0051I **Different ACBMGMT types are found in the data sharing members.**

Explanation:
The user tried to register or update a data sharing group, but the type of ACB management (ACBMGMT) specified for each data sharing group member was not the same. All IMS members in a data sharing group must use the same ACB management type.

System action:
The operation is not performed.

User response:
Refer to the IMS configuration requirements for the IMS products that are activated on the Unified Management Server.

IZPMS0055I The subsystem is not implemented as an IMSplex.

Explanation:
The user tried to register or update an IMS subsystem, but it was not a data sharing group that belongs to a CSL IMSplex. Only a group of IMS subsystems that belong to an IMSplex and that share a same set of RECON data sets can be registered as an IMS data sharing group to be used as an IMS subsystem by the Unified Management Server.

System action:
The operation is not performed.

User response:
Confirm that the data sharing group you were trying to register satisfies all the prerequisite conditions for the Unified Management Server.

IZPMS0056E You cannot register the subsystem because it has one or more errors.

Explanation:
One or more errors were found when the user tried to register an IMS data sharing group.

System action:
The operation is not performed.

User response:
Refer to the IMS configuration requirements for the IMS products that are activated on the Unified Management Server.

IZPMS0057E You cannot update the subsystem because it has one or more errors.

Explanation:
One or more errors were found when the user tried to register an IMS data sharing group.

System action:
The operation is not performed.

User response:
Refer to the IMS configuration requirements for the IMS products that are activated on the Unified Management Server.

IZPMS0060E Another subsystem '*dsgroup*' is already registered with same RECON data sets.

Explanation:
The user tried to register a data sharing group, but another IMS data sharing group *dsgroup* was already registered with the same RECON data set names. A data sharing group is identified by RECON data set names.

System action:
The operation is not performed.

User response:
Verify the IMS data sharing group that has the same RECON data sets and is already registered. If it is the data sharing group that you intend to register, use the existing registration. If not, contact the IMS system administrator.

IZPMS0061E The subsystem defines more than one IMS Connects that have the same ODBM alias: '*alias*'.

Explanation:
The user tried to register an IMS data sharing group by specifying the same alias for multiple IMS Connect ports. You can specify only one IMS Connect port for an ODBM alias to be registered.

System action:
The operation is not performed.

User response:
Correct IMS Connect definitions for the data sharing group registration and retry the operation.

IZPMS0062E IMS Connect '*connect_name*' defines more than one ports or does not define any ports.

Explanation:
The user tried to register an IMS subsystem as a data sharing group. More than one port was selected from the IMS Connect server *connect_name* for a port type. Each IMS data sharing group to be registered can have at most one port with one or more ODBM datastore aliases for the use of IMS SQL processor and at most one port for the use of IMS command processor.

System action:
The operation is not performed.

User response:
Correct IMS Connect definitions of subsystem and retry the operation.

IZPMS0063E IMS RESLIB data set name is not determined for subsystem: '*subsystem_name*'.

Explanation:
IMS RESLIB data set name is not discovered for the subsystem *subsystem_name*.

System action:
The operation is not performed because a JCL for backend processing cannot be built.

User response:
Contact your system administrator to identify why no IMS RESLIB was discovered by the Unified Management Server.

IZPMS0067E The subsystem defines more than one command type IMS Connects.

Explanation:

The user tried to register an IMS data sharing group, but two or more IMS Connect ports for IMS command processor were specified. You can specify only one port for IMS command processor in an IMS data sharing group registration.

System action:
The operation is not performed.

User response:
Select one port from the list of IMS Connect ports for IMS command processor and retry the operation.

IZPMS0068E Value '*value*' supplied for field '*field_name*' is not discovered.

Explanation:
The user tried to test if a connection can be established with IMS Connect by giving hostname and port, but the given value had not been discovered by the Unified Management Server. The value *field_name* is either a "hostname" or a "port".

System action:
The operation is not performed.

User response:
Correct the hostname, port, or both, and retry the operation. If both hostname and port values are correct, the IMS Connect may not be active. Contact your system administrator.

IZPMS0069I '*field_name*' is not specified.

Explanation:
The user tried to register or update an IMS data sharing group, but no value was specified for the field *field_name*. The *field_name* can be a Command processor port.

System action:
Subsystem is registered or updated since the field reported in the message is an optional field. Some functionalities will not be available for the registered IMS data sharing group.

User response:
If you want to use all the features that can be provided by the activated IMS products for the registered IMS data sharing group, update the registration by specifying a valid value for the field *field_name*. If the *field_name* is "Command Processor port" and you want to use the IMS command processor for full functionality of IMS Administration Foundation, an IMS Connect port needs to be specified as a port for the IMS command processor. The port needs to satisfy specific conditions. For details, see "Software requirements for IMS Administration Foundation" on page 39.

IZPMS0100E No available IMS Connect server is registered for the IMSplex: '*plexname*'

Explanation:
The user tried to perform an operation that eventually issued one or more IMS commands, but no port of any IMS Connect that was connected to the IMSplex *plexname* and that can be used for the IMS command processor was found in any registered IMS data sharing group.

System action:
The operation is not performed.

User response:
Register at least one port of an IMS Connect server that is connected to the IMSplex *plexname* and that can be used for the IMS command processor in at least one IMS data sharing group.

IZPMS0101E 'DBD and PSB map service' failed due to a server problem: '*response_text*'

Explanation:
The user requested a DBD map or a PSB map, but the backend service returned an error with the message *response_text*.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the DBD and PSB map service provided by IMS Library Integrity Utilities. The system administrator may need to identify the IMS Tools Distributed Access Infrastructure's SOT address space in which the service was running. If the *response_text* includes the 'AII1301E TAS Server cannot be located', refer to IMS Administration Foundation and IMS Tools for the requirements to use the DBD and PSB map feature.

IZPMS0102E IMS Library Integrity Utilities is not configured or required modules are missing

Explanation:
The user requested a DBD map or a PSB map, but the backend service was not configured as expected.

System action:
The operation is not performed.

User response:
Check if the required module `FABXGMAP` is in the libraries that are defined under `imsToolsLoadlib` parameter in the PARMLIB member `IZPIMFPM` for the Unified Management Server that is used by the user.

IZPMS0103E No *ddname* is defined in the subsystem

Explanation:
The user requested a DBD map or a PSB map, but the required IMS library of DD name *ddname* was not

defined in the RECON ID record that was selected for the registration of the IMS data sharing group for which the subject DBD or PSB was defined. The *ddname* is either DBDLIB or PSBLIB.

System action:
The operation is not performed.

User response:
Identify the RECON ID that is registered in the data sharing group registration and specify a DBDLIB or a PSBLIB in the RECON ID definition.

IZPMS0104E Not enough space for a temporary data set is available on the DASD volume

Explanation:
The user requested a DBD map or a PSB map, but an error occurred while attempting to allocate a temporary data set. A possible cause of the error can be that enough DASD space was not available.

System action:
The operation is not performed.

User response:
Contact your system administrator to resolve the cause of the failure and retry the operation.

IZPMS0105E Server returned an invalid response

Explanation:
The user requested a get DBD map or a PSB map, but an unexpected response was returned from the backend service.

System action:
The operation is not performed.

User response

Collect the following information and contact IBM Software Support for assistance:

- The Unified Management Server log in the `zowe.logDirectory` directory.
- The configuration files in the UMS PARMLIB data set

IZPMS0106E IMS Command '*command*' was failed: RC=*rc*, RSN=*rsn*

Explanation:
The user tried to perform an operation that eventually issued the IMS command, that is, *command*, but the command failed with a return code of *rc* and a reason code of *rsn*.

System action:
The operation is not performed.

User response:
Refer to the section for the subject command in an *IMS Commands* reference to find the explanation for the pair of the return code *rc* and the reason code *rsn*. Resolve the cause of the failure and retry the operation.

IZPMS0107E No IMS Tools Knowledge Base server is registered for the IMS data sharing group: '*subsystem*'

Explanation:
The user requested a service that needs a connection to an IMS Tools Knowledge Base (IMS Tools KB) server to get an information related to the IMS data sharing group *subsystem*, but no IMS Tools KB server was selected in the registration for the data sharing group *subsystem*.

System action:
The operation is not performed.

User response:
Check if appropriate IMS Tools KB servers were specified in the PARMLIB member `IZPIMFPM` for the Unified Management Server that is used by the user. If no IMS Tools KB server was set up for the subject data sharing group, you need to ask your IMS system administrator or database administrator to set up an IMS Tools KB server and create a RECON ID in which the RECON data sets for the subject data sharing group are specified.

IZPMS0108E RECON '*recon_id*' is not discovered

Explanation:
The user tried to get a RECON information related to the RECON ID *recon_id*, but the RECON ID was not discovered in the selected IMS Tools Knowledge Base server.

System action:
The operation is not performed.

User response:
Check if the RECON ID is registered in the IMS Tools Knowledge Base server. Then run an infrastructure discovery process. If you are not a UMS super administrator, ask a UMS super administrator to run an infrastructure discovery process by clicking the "Refresh page" icon on the "Discovered" tab of the "Subsystems" page of the Unified Experience user interface. A UMS super administrator can also use the `PUT /ws/infrastructure/discovery` API call.

IZPMS0109E No *library* is specified in the RECON ID that was registered for the data sharing group '*subsystem*'. Check the RECON ID '*recon_id*' defined on an IMS Tools Knowledge Base server that belongs to the XCF group '*xcf_group*'.

Explanation:

The user tried to get a DBD map or a PSB map, but the required *library* was not defined in the RECON ID *recon_id* for the data sharing group *subsystem* on any IMS Tools Knowledge Base server that belongs to the IMS Tools KB XCF group '*xcf_group*'. *library* is either DBDLIB or PSBLIB.

System action:
The operation is not performed.

User response:
Register one or more DBDLIB or PSBLIB libraries to the RECON ID.

IZPMS0110E **No IMS Tools load module library is specified for the `imsToolsLoadlib` parameter**

Explanation:
The Unified Management Server tried to perform some IMS Tools functions, but a required IMS Tools load module or modules were not found in any data set that was specified for the `imsToolsLoadlib` parameter in the UMS PARMLIB member `IZPIMFPM`.

System action:
The operation is not performed.

User response:
Check the `imsToolsLoadlib` parameter of the `IZPIMFPM` member and specify an appropriate set of IMS Tools load module library data sets.

IZPMS0111E **No authorization: You don't have permission to access the data sets that the IMS Library Integrity Utilities (LIU) requires. Contact your system administrator to request access.**

Explanation:
The user tried to get a DBD map or a PSB map, but the user did not have permission to access the data set required for the function. The data set to be accessed is either DBDLIB libraries or PSBLIB libraries that are registered to the RECON ID of the subject data sharing group.

System action:
The operation is not performed.

User response:
Contact your system administrator to request the access to the data set. You need to inform the administrator of the RECON ID that is specified in the registration of the IMS data sharing group to which the DBD or PSB that you were tried to access belongs.

IZPMS0112E **No authorization: You don't have permission to access the IMS Tools Base Distributed Access Infrastructure (DAI) server '*jobname*'. Contact your system administrator to request access.**

Explanation:
The user tried to perform a function that required access to the IMS Tools Base Distributed Access Infrastructure server, but the user did not have permission to access the server that is identified by the job *jobname*.

System action:
The operation is not performed.

User response:
Contact your system administrator to request access to the server. For details of permissions for DAI servers, see the chapter "Summary of security-related settings" in *IBM IMS Tools Base for z/OS Configuration Guide*.

IZPMS0113E **No authorization: You don't have permission to access the IMS Tools Knowledge Base (IMS Tools KB) server '*xcf_group*'. Contact your system administrator to request access.**

Explanation:
The user tried to perform a function that required access to the IMS Tools Knowledge Base server, but the user did not have permission to access the server that is identified by the IMS Tools KB server XCF group *xcf_group*.

System action:
The operation is not performed.

User response:
Contact your system administrator to request access to the server. For details of permissions for IMS Tools KB servers, see the chapter "Summary of security-related settings" in *IBM IMS Tools Base for z/OS Configuration Guide*.

IZPMS0114E **No authorization: You don't have permission to access the IMS Tools Base Autonomics Director (AD) server '*xcf_group*'. Contact your system administrator to request access.**

Explanation:
The user tried to perform a function that required access to the IMS Tools Base Autonomics Director server, but the user does not have permission to access the server which is identified by XCF group *xcf_group*.

System action:
The operation is not performed.

User response:

Contact your system administrator to request access to the server. For details of permission for the AD server, see the chapter "Summary of security-related settings" in *IBM IMS Tools Base for z/OS Configuration Guide*.

IZPMS0115E No authorization: You don't have permission to run commands to the IMSplex '*imsPlexName*'. Contact your system administrator to request access.

Explanation:
The user tried to issue an IMS command through an IMS Connect and a CSL OM server for the IMSplex *imsPlexName*. The user did not have enough permission to perform the command.

System action:
The operation is not performed.

User response:
Contact your system administrator to request access. For details of IMS OM command security, see the section "CSL OM command security" in *IMS System Administration guide*.

IZPMS0116E No authorization: You don't have permission to view the resources. Contact your system administrator to request access.

Explanation:
The user tried to get IMSplex information by using the feature that issued the IMS command `QUERY IMSPLEX` through an IMS Connect and a CSL OM server for the subject IMSplex. The user did not have enough permission to perform the command.

System action:
The operation is not performed.

User response:
Contact your system administrator to request access. See the security setting for the `QUERY IMSPLEX` command in the section "CSL OM command security" in *IMS System Administration Guide*.

IZPMS0117E Invalid route parameter: '*parameter*'. '*' and '%' must be specified alone.

Explanation:
The user tried to perform a function to issue an IMS command with explicit IMS members to be routed by the *parameter*. The *parameter* includes `'*'` (all) or `'%'` (any) with other characters. The `'*'` and `'%'` characters must be specified as a single character.

System action:
The operation is not performed.

User response:
Correct the route parameter and retry the operation.

IZPMS0118E Requested service failed due to a problem on IMS Tools Base Distributed Access Infrastructure (DAI) TCP server: *response_text*. Contact your system administrator with the following information.
- DAI TCP server XCF group: *xcf_group*, - Job name: *jobname*, - Host name: *hostname*:*port*.

Explanation:
The user tried to perform a function that required access to the IMS Tools Base Distributed Access Infrastructure (DAI) server, but the DAI TCP server returned an error with the message *response_text*. The DAI TCP server can be identified by the DAI TCP server XCF group name *xcf_group*, job *jobname*, or a pair of *hostname* and *port*.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the DAI TCP server.

IZPMS0119E Requested service failed due to a problem on IMS Tools Base Distributed Access Infrastructure (DAI) Tools Access Server(TAS): *response_text*. Contact your system administrator with the following information. - DAI TAS server XCF group: *xcf_group*, - Job name: *jobname*.

Explanation:
The user tried to perform a function that required access to the IMS Tools Base Distributed Access Infrastructure (DAI) server, but the DAI Tools Access Server (TAS) returned an error with the message *response_text*. The DAI TAS is identified by the DAI TAS server XCF group name *xcf_group* and job *jobname*.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the DAI TAS.

IZPMS0120E Requested service failed due to a problem on IMS Tools Base Distributed Access Infrastructure (DAI) Subordinate Tools Access Server (SOT): *response_text*. Contact your system administrator with the following information. - Job name: *jobname*.

Explanation:

The user tried to perform a function that required access to the IMS Tools Base Distributed Access Infrastructure (DAI) server, but the DAI Subordinate Tools Access Server (SOT) returned an error with the message *response_text*. The DAI SOT is identified by job *jobname*.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the DAI SOT address space.

IZPMS0121E **Requested service failed due to a problem on IMS Tools KB server for REPORT service: *response_text*. Contact your system administrator with the following information. - IMS Tools KB server XCF group: *xcf_group*, - RECON ID: *recon-id*.**

Explanation:
The user tried to perform a function that required access to the IMS Tools Knowledge Base server for REPORT service that is identified by the IMS Tools KB server XCF group name *xcf_group* and the RECON ID *recon_id*, but the backend service returned an error with the message *response_text*. If *response_text* is "ITKB connection error (RC=28, RSN=12, NAME=*jobname*)" and Tools Base 1.7 APAR PH48047 is not applied, it may mean that the user did not have permission to access the server that is identified by the IMS Tools KB server XCF group name *xcf_group*.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the REPORT service.

IZPMS0122E **Requested service failed due to a problem on IMS Tools KB server for SENSOR service: *response_text*. Contact your system administrator with the following information. - IMS Tools KB server XCF group: *xcf_group*, - RECON ID: *recon-id*.**

Explanation:
The user tried to perform a function that required access to the IMS Tools Knowledge Base server for SENSOR service that is identified by the IMS Tools KB server XCF group *xcf_group* and RECON ID *recon_id*, but the backend service returned an error with the message *response_text*. If *response_text* is "Sensor Request Failed: 0 processed, 0 in error (Sensor INIT Error RC=12, RSN=11, NAME=*jobname*)" and Tools Base 1.7 APAR PH48047 is not applied, it may mean that the user did not have permission to access the server that is identified by the IMS Tools KB server XCF group *xcf_group*.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the named backend service component.

IZPMS0123E **Requested service failed due to a problem on IMS Tools Base Autonomics Director (AD) server: *response_text*. Contact your system administrator with the following information. - IMS Tools KB server XCF group: *xcf_group*, - RECON ID: *recon-id*, - AD server XCF group: *ad_xcf_group*.**

Explanation

The user tried to perform a function that required access to the IMS Tools Base Autonomics Director server that is identified by the IMS Tools KB server XCF group name *xcf_group*, RECON ID *recon_id* and AD server XCF group name *ad_xcf_group*, but the backend service returned an error with the message *response_text*.

- If the *response_text* is “Autonomics Director Error: Environment has no entries (RC=8, RSN=2, MOD=IAVUXSR)”, it is highly possible that the RECON ID was added after the AD server started. So, perform the AD REFRESH function for RECON and MONLIST types by using the AD server REFRESH command.
- If the *response_text* is "Autonomics Director Error: Environment requested by RECON EXTID not found. See IAV1000E. (RC=8, RSN=4, MOD=IAVUXSR)", see the explanation of the message IAV1000E. This error can occur when the RECON ID is renamed or deleted after the IMS data sharing group that is associated with the RECON ID is registered with UMS.
  - If the RECON ID was renamed, you need to update the registration on the UMS server.
  - If the RECON ID was deleted, you must recreate a RECON ID that is associated with the RECON data sets that are used for the subject data sharing group on an IMS Tools Knowledge Base server, which is accessible from the UMS server, and update the data sharing group registration on the UMS server.

  After adding a new RECON ID and optionally configuring the AD Monitor List, you must issue AD server REFRESH commands: a REFRESH command with the TYPE(RECON) parameter, followed by another REFRESH command with the TYPE(MONLIST) parameter.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error in the AD service. If the *response_text* is either of those mentioned in the Explanation, take the action that is explained. Actions for other errors would be self-explanatory from the *response_text*. If the cause of the error cannot be identified or the action you took did not resolve the error, contact IBM Software Support.

IZPMS0124E Requested service failed due to a problem on an IMS Tools Base server: *response_text*.

Explanation:
The user tried to perform a function that required access to an IMS Tools Base server, but the backend service returned an error with the message *response_text*. If the *response_text* indicates a missing configuration or configuration error, see the configuration requirements in Installing IMS Administration Foundation and its referenced sections. If the UMS server is configured with authentication type of MFA_JWT, see Enabling IMS Tools TCP server to receive RACF PassTickets.

System action:
The operation is not performed.

User response:
Contact your system administrator and identify the cause of the error.

IZPMS0125E No authorization: You don't have permission to access the IMS catalog that the IMS Library Integrity Utilities (LIU) requires, and LIU ended with the message *'message_text'*. Contact your system administrator to request access.

Explanation

The user tried to get the DBD or PSB Map but did not have enough permission to access the IMS catalog of the target IMS data sharing group. The IMS Library Integrity Utilities (LIU) program that was used to provide the DBD or PSB Map function ended with the message *'message_text'*. The message text provides information on the cause of the error.

System action:
The operation is not performed.

User response:
Refer to the message references for LIU and IMS, and contact your system administrator to request access to the data set.

IZPMS0126E You do not have enough privilege to use the application assigned to '*ims_service*' for the subject data sharing group. Contact your system administrator and security administrator.

Explanation:
The user tried to perform an operation that accesses the IMS service identified by the name '*ims_service*', but PassTicket generation to access the IMS service failed. The variable *ims_service* is one of the Command processor port, SQL processor port, and DAI TCP server.

System action:
The operation is not performed.

User response:
Contact your system administrator to isolate the issue in the `${zowe.logDirectory}/izp-server-<sysname>-<latest_timestamp>.log file`. The system administrator needs to collect IZPMS0126E and subsequent warning messages.

IZPMS0127E SAF authentication error occurred on IMS Connect. If IMS Connect issued message HWSP1500E, follow the instructions in the System programmer response section of message HWSP1500E.

Explanation:
The user tried to perform an operation that issued one or more IMS commands, but RACF security verification failed on the IMS Connect server job.

System action:
The operation is not performed.

User response:
Contact your system administrator and security administrator to request access to the resource for which you received the security violation. The system administrator may need to collect an error message from the IMS Connect server job.

IZPMS0128E *product_name* version *version1* is below required version *version2*.

Explanation:
The user tried to perform an operation that required the service provided by the product *product_name*. To use the service, the product version *version2* or later is required, but an earlier version (*version1*) has been installed.

System action:
The operation is not performed.

User response:
Contact your system administrator to install the product version *version2* or higher.

IZPMS0129E ***product_name* is not registered for the IMS Tools Knowledge Base server that belongs to the XCF group *xcf_group_name*.**

Explanation:
The user tried to perform an operation that required the service provided by the product *product_name*, but the product had not been registered to the IMS Tools Knowledge Base that belongs to the IMS Tools KB XCF group *xcf_group_name*.

System action:
The operation is not performed.

User response:
Contact your system administrator to request the registration of the product to the IMS Tools Knowledge Base that belongs to the XCF group *xcf_group_name*.

IZPMS0130E ***parameter_name* of *product_name* is missing as product information on the IMS Tools Knowledge Base server that belongs to the XCF group *xcf_group_name*.**

Explanation:
The user tried to perform an operation that required the service provided by the product *product_name*, but the necessary product information that is identified by the keyword *parameter_name* was not registered in the IMS Tools Knowledge Base server that belongs to the IMS Tools KB XCF group *xcf_group_name*.

System action:
The operation is not performed.

User response:
Contact your system administrator to request registration of the required product information to the IMS Tools Knowledge Base server.

IZPMS0131E **Subsystem which has the specified RECON data set *data_set_name* is not discovered.**

Explanation:
The user or API client passed a RECON data set name *data_set_name* to UMS as a request parameter, but no IMS subsystems were discovered by UMS that were using the given RECON data set.

System action:
The operation is not performed.

User response:
Check if a correct RECON data set name of an IMS subsystem is used. If you are a UMS super administrator who received this error while creating or updating an IMS data sharing group registration on the UMS UI, then refresh the IMS subsystem list by clicking the **Refresh** button on the **Discovered** tab of the **Subsystems** page and retry the operation.

IZPMS0132E **RECON ID *new_id* is already registered to the IMS Tools Knowledge Base server.**

Explanation:
The user tried to create a new RECON ID or rename an existing RECON ID, but the given RECON ID *new_id* was already registered to the target IMS Tools Knowledge Base server.

System action:
The operation is not performed.

User response:
Specify an unused RECON ID and retry the operation.

IZPMS0133E **Error on calling the IMS Administration Tool service *service_name* RC/RSN=*rc*/rsn: *response_text*.**

Explanation:
The user tried to perform an operation that eventually called a service of the IBM IMS Administration Tool for z/OS, but the service failed with return code *rc*, reason code *rsn*, and message *response_text*. The invoked service is indicated by the string *service_name*.

System action:
The operation is not performed.

User response:
Contact your system administrator to resolve the cause of the failure and retry the operation.

IZPMS0134E **No authorization: You don't have permission to access the IMS Administration Tool. Contact your system administrator to request access.**

Explanation:
The user tried to perform an operation that required a service provided by IBM IMS Administration Tool for z/OS (also referred to as IMS Administration Tool) but did not have permission to access the service. The required permissions vary depending on the service that was used. If the user tries to use the service while registering an IMS data sharing group or updating an existing IMS data sharing group registration, they should have permission to set up the IMS Administration Tool environment, including registering IMS systems. For details on the security requirements for IMS Administration Tool, see "Security setup for IMS Administration Tool" on page 82.

System action:
The operation is not performed.

User response:
Contact your system administrator to request access to the IMS Administration Tool.

IZPMS0135E You cannot update the RECON ID because it has one or more errors.

Explanation:
The user tried to update one or more data set names for the DBD (DBDLIB) or PSBLIB (PSBLIB) libraries stored in a RECON ID but one or more errors were found. One possible reason for this error is that the specified data set or data sets do not exist.

System action:
The operation is not performed.

User response:
Check if all the DBDLIB and PSBLIB data set names that were specified are correct. If there is an error in any of the data set names, correct all errors in the DBDLIB and PSBLIB names and retry the operation. If the same error message is displayed even after correcting the data set names, contact IBM Software Support.

IZPMS0136E RECON ID *recon_id* does not exist.

Explanation:
The user tried to get or update RECON ID information, but the given RECON ID that has the specified name *recon_id* was not registered to the target IMS Tools Knowledge Base server.

System action:
The operation is not performed.

User response:
Check if the RECON ID the user specified was correct. If not, specify a correct RECON ID and retry the operation.

IZPPI messages

Messages that are issued when you run the post-installation script or other installation and configuration related scripts that have the format `IZPPInnnnx`.

IZPPI0073I Post-installation retrieving certificate from Zowe keystore.

Explanation:
The script is attempting to import the Zowe certificate into the UMS keystore.

System action:
Wait for the script processing to complete.

User response:
Wait for the script processing to complete.

IZPPI0074I Zowe certificate status: OK

Explanation:
The script successfully imported the Zowe certificate into the UMS keystore.

System action:
The script completed successfully. No action is required.

User response:
The script completed successfully. No action is required. Proceed to the next script.

IZPPI0001E Error: LISTDS failed: *additional_info*

Explanation:
The named data set was unavailable for the script to obtain a list of members through the **`tsocmd`** command. This can happen while APF authorizing a data set if it is not available, or when copying a z/OS UNIX System Services file to a data set. Additional information will be provided with a root cause.

System action:
Action (APF authorization or file copy) ends abruptly.

User response:
Determine and correct access or availability of the named data set and rerun the command.

IZPPI0010W Experience post-installation status: *file* NOT OK

Explanation:
The named data management product post-installation script returned a nonzero exit code. Additional error messages are printed.

System action:
Data management experience post-installation continues with additional scripts.

User response:
Identify the cause by referring to the additional messages and take a corrective action.

IZPPI0011W Unknown log option `components.izp.server.log.destination`; logging to file.

Explanation:
Value for `components.izp.server.log.destination` in UMS started task JCL was not one of empty string, `FILE`, `STDOUT`, or `BOTH`.

System action:
`FILE` was chosen as value, and log was sent to file. Task was started.

User response:
Correct value in started task, stop task, and restart.

IZPPI0020E Failed to start *server_name*.

Explanation:
The server might fail to start for a number of potential reasons. Additional error messages are printed.

System action:
Server startup ends abruptly.

User response:
Identify the cause from additional messages and take a corrective action.

IZPPI0020W Failed to update `$ZWE_zowe_setup_dataset_proclib($PROCLIB_MEMBER)`. A copy has been saved to `$ZWE_components_izp_dataset_hlq.PROCLIB($PROCLIB_MEMBER)`. Copy it manually to `$ZWE_zowe_setup_dataset_proclib($PROCLIB_MEMBER)`.

Explanation:
The user running the STEPLIB concatenation procedure (IZPSTEPL job) does not have write access to the Zowe setup data sets.

System action:
The member was saved to another data set (`IZP HLQ.PROCLIB`).

User response:
Copy the specified member to the Zowe data set. A Zowe restart will be required.

IZPPI0028W Setting Program Control on *<file>*

Explanation:
Unified Management Server plugin for ZSS requires program control. A file was found without program control.

System action:
IZPIPLUG will attempt to set program control on the file.

User response:
If the action is successful, no user response is required. If this fails, you need to manually set program control.

IZPPI0035E ERROR: cannot import certificate

Explanation:
If your system is configured with secure access to z/OSMF, the post-installation process generates, by using OpenSSL, a file that contains the certificate to allow connection. The certificate is then imported into a Java KeyStore for access by the server. This message indicates that the import into a Java KeyStore format file failed, possibly due to permissions issues. Additional messages will be printed by the Java Keytool program.

System action:
Certificate generation script ends abruptly; post-installation script continues.

User response:
Verify host and port of services such as z/OSMF, SQL Tuning Services, and Db2 Analytics Accelerator Administration Services.

IZPPI0037E Post-install cannot copy files and/or datasets

Explanation:
This message is printed if any of the several actions performed by experience installation job IZPEXPIN or experience configuration during launch fails.

System action:
Data set and other copies abruptly ended; post-installation script continues.

User response:
Correct any permissions or other access problems, and rerun IZPEXPIN if appropriate, or relaunch Zowe.

IZPPI0049E Copying from `${PARMLIB}` (`$member`) to `$file` failed.

Explanation:
The internal script used to copy from the `PARMLIB` data set was unable to create one or more configuration YAML files. Additional error messages will be printed.

System action:
The `PARMLIB` member is not copied.

User response:
Correct any permissions or other access problems and rerun the command that caused the error.

IZPPI0051E Provided location *experience* is not a folder

Explanation:
Data management experience installation requires the **`components.izp.experiences`** variable to be set to the original SMP/E location of the data management product installation. This message is produced when this directory is invalid.

System action:
Post-installation of the data management product is ended abruptly.

User response:
Set `components.izp.experiences` array in ZWEYAML to correct yaml array syntax with all experience installation directories and resubmit `IZPEXPIN`.

IZPPI0052E Cannot find opt and var folders under experience

Explanation:
Data management experience installation calls an internal Unified Management Server script with a parameter to define the original SMP/E location for the data management product "opt" and "var" directories, based on the components.izp.experiences array. This message is produced when these directories are not found.

System action:
Post-installation of the data management product is ended abruptly.

User response:
Set `components.izp.experiences` array in ZWEYAML to correct yaml array syntax with all experience installation directories and resubmit `IZPEXPIN`.

IZPPI0056E {*PARMLIB*}(*membername*) was empty. Not copied. Provide corrected member.

Explanation:
PARMLIB(*membername*) was empty and did not contain any values when values were expected.

System action:
The member is not copied to corresponding configuration `yaml` file in `var/conf`.

User response:
Provide the corrected member and restart the failed command.

IZPPI0057E No value specified for *<variable>*

Explanation:
A required variable has no value and hence the operation has been terminated.

System action:
Operation ends abruptly until a value is provided.

User response:
Verify that you’re supplying all the required arguments and retry this operation.

IZPPI0058E Incorrect value '$`value`' specified for $`variable`

Explanation:
Variable has an invalid value and hence the operation has been terminated.

System action:
Operation ends abruptly until a valid value is provided.

User response:
Verify that you are supplying a valid value and retry the operation.

IZPPI0062E Cannot copy $ussfile to $tomember - $error_text

Explanation:
The system is attempting to copy a UNIX System Services file to a data set member and was unable to do so.

System action:
File is not copied to data set.

User response:
Examine the error text produced and determine what corrective action to take.

IZPPI0064E Could not locate a valid certificate

Explanation:
A valid z/OSMF CA certificate could not be located and added to the UMS TrustStore.

System action:
No certificate is added to the UMS TrustStore

User response:
Check the output of the script for any additional errors and resolve them before re-running `izp-gen-cacerts.sh`.

IZPPI0076E Cannot update classpath file `$classpath_file`

Explanation:
Started task user unable to create the class path file for UMS.

System action:
Class path is not updated; UMS will not function properly.

User response:
Examine preceding error messages, and address any permission issues, or determine what other corrective action to take.

IZPPI0077E Required jar location argument is not a directory.

Explanation:
When setting up class path, received an invalid jar location, possibly in experience installation.

System action:
Class path is not updated; UMS will not function properly.

User response:
Verify that experience installation location is correct, and that experience installation contains folder `opt/lib` and `opt/bin`.

IZPPI0078E Internal configuration error: $variable not set

Explanation:

An internal script did not provide appropriate parameters.

System action:
A message is printed, and operation ends abruptly.

User response:
Contact IBM Software Support.

IZPPI0088E The file, $FROM_FILE, is not readable. Verify that the file exists and is readable.

Explanation:
Internal error if unable to read from runtime directory for experience during IZPEXPIN.

System action:
New values for experience-specific PARMLIB are not integrated into experience PARMLIB member.

User response:
Verify that the user running IZPEXPIN has read access to the SMP/E location for experience, and that experience location is specified in PARMLIB(ZWEYAML).

IZPPI0089E The file, $TO_FILE, is not writable. Verify that the file exists and is writable.

Explanation:
Internal error if unable to write to temporary directory during IZPEXPIN.

System action:
New values for experience-specific PARMLIB are not integrated into experience PARMLIB member.

User response:
Verify that the user running IZPEXPIN can write to /tmp folder.

IZPPI0090E Unable to successfully write to temporary file, $TO_FILE.

Explanation:
Internal error if unable to write to temporary file during IZPEXPIN.

System action:
New values for experience-specific PARMLIB are not integrated into experience PARMLIB member.

User response:
Verify that the user running IZPEXPIN can create a file in /tmp folder.

IZPPI0100E ${PARMLIB}($member) was apparently edited with NUM ON. Remove line numbering and retry.

Explanation:
PARMLIB members are read by UNIX System Services programs and must be edited with numbering off.

System action:
PARMLIB member is not copied to HFS file system.

User response:
Edit PARMLIB member and remove numbering.

IZPPI0100I Check ZWEYAML member for new parameter *<element>*.

Explanation:
IZPSYNCY detected a new element in the updated ZWEYAML source.

System action:
A new element is incorporated into the new ZWEYAML.

User response:
Examine new ZWEYAML comments about this element and determine if site-specific updates are required for the default value.

IZPPI0108E <STORE_TYPE> is required when <STORE> is specified, and the protocol is not present.

Explanation:
There is a configuration problem with components.izp.security.certificate.keystore.location, components.izp.security.certificate.truststore.location, components.izp.security.certificate.keystore.type, and components.izp.security.certificate.truststore.type .

System action:
This is an invalid configuration. The Unified Management Server will not start.

User response:
Confirm that components.izp.security.certificate.keystore.location, components.izp.security.certificate.truststore.location, components.izp.security.certificate.keystore.type, and components.izp.security.certificate.truststore.type have valid values.

IZPPI0109E In-place migration from 1.1 to 1.2 is not supported. Select another location for workspace folder. Found: ${izp_workspace}

Explanation:
The workspace folder specified in PARMLIB(ZWEYAML) as components.izp.workspaceDirectory is the same as the 1.1 IZP_UMS_VARDIR.

System action:
Migration does not run and Zowe configuration step returns failure code.

User response:
Select another folder for 1.2 workspace location.

IZPPI0110E Failed to access directory $directory specified as <yaml definition element>.

Explanation:
The runtime directory specified in PARMLIB(ZWEYAML) or the zowe.yaml file by the yaml path specified is not found or is not readable.

System action:
Zowe validate step returns failure code and Zowe does not start.

User response:
Ensure that the correct directory is specified.

IZPPI0111E **Workspace directory, $ZWE_components_izp_workspaceDirectory, cannot be in run time directory.**

Explanation:
Workspace directory, specified in PARMLIB(ZWEYAML) by components.izp.workspaceDirectory, was found within the hierarchy of the directory specified by components.izp.runtimeDirectory.

System action:
Zowe validate step returns failure code and Zowe does not start.

User response:
Select another folder for workspace location.

IZPPI0112E **Required value for <yaml definition element> is not defined.**

Explanation:
Yaml element provided in message does not have a value. A value is required for the particular element.

System action:
Zowe validate step returns failure code and Zowe does not start.

User response:
Check the element in in PARMLIB(ZWEYAML) and provide a value.

IZPPI0113E **Error: LISTDS failed: $tsocmdout**

Explanation:
Set up for copying UNIX System Services file to data set member failed to determine content of data set.

System action:
Script terminates without copying file.

User response:
Examine the error text produced and determine what corrective action to take.

IZPPI0114E **Must use MFA_JWT authentication type for High Availability with UMS**

Explanation:
UMS high-availability support requires Zowe Single Sign-On (SSO) mechanism supported by API Mediation Layer. The SSO login type is selected by using the MFA_JWT authentication type.

System action:
Zowe start up is interrupted during IZP validation phase and Zowe does not start.

User response:
Set components.izp.server.authType to MFA_JWT and restart Zowe.

IZPPI0115E **Installation cannot modify ${file} via extattr. Manual change to Program Control required.**

Explanation:
Unified Management Server (UMS) plugin for ZSS requires program control. If this is not available, IZPIPLUG will attempt to set the bit with the extattr program during installation.

System action:
IZPIPLUG terminates without installing UMS in Zowe.

User response:
You need to manually set program control with UNIX System Services command extattr +p *<filename>*.

IZPPI0116E **Write permission for ${file} is not granted. Manual change to Program Control required.**

Explanation:
Unified Management Server (UMS) plugin for ZSS requires program control. If this is not available, IZPIPLUG will attempt to set the bit with the extattr program during installation. The file must be writable for this action.

System action:
IZPIPLUG terminates without installing Unified Management Server in Zowe.

User response:
You need to manually set program control with UNIX System Services command extattr +p *<filename>*.

IZPPI0200E **Could not create a copy of data set $environmentDataSet as a unix file $environmentFile.**

Explanation:
UMS was unable to find the environment data set that was created by IZPGENER or was unable to create a UNIX temporary file with the content.

System action:
Script terminates abruptly.

User response:
Provide corrected HLQ to location of ENVIRON data set.

IZPPI0201E **The provided environment data set `$environmentDataSet` is not valid.**

Explanation:
UMS found the environment data set created by IZPGENER but was unable to process it as it did not contain the necessary content.

System action:
Script terminates abruptly.

User response:
Validate that the data set provided is the same as the one that was created by IZPGENER, and that it contains a number of entries for `IZP_ZOWE_RUNTIME`, `IZP_SCHEMA_CHAIN`, and `IZP_CONFIG_CHAIN`. Rerun IZPGENER if needed to re-create the data set or provide a different HLQ.

IZPPI0202E **Could not encrypt credentials. Refer to the log file: $ `{LOG_PATTERN}`.**

Explanation:
Credential encryption was attempted but did not succeed.

System action:
Script terminates abruptly and encrypted credentials are not stored.

User response:
Refer to a log file for error message and provide corrected input and rerun script.

IZPPI0203E **The provided environment data set `$environmentDataSet` is not valid - `components.izp.dataset.hlq` value $ `{ZWE_components_izp_dataset_hlq}` does not equal `IZP_HLQ` value `$hlq`.**

Explanation:
UMS found the environ data set, but when processing it, found an anomaly that the HLQ provided to the script did not equal the HLQ in the ZWEYAML member.

System action:
Script terminates abruptly.

User response:
Ensure that the HLQ provided to the script is the same as the one entered in ZWEYAML as `components.izp.dataset.hlq`.

IZPPI0204E **Validation unsuccessful because of an error in running configmgr for $ `{ZWE_components_izp_dataset_parmlib}($membername)`.**

Explanation:
Zowe config manager did not run. Only occurs with Zowe 2.4 and earlier.

System action:
UMS did not start.

User response:
Correct problems based on error messages and restart the UMS component. Error messages may be difficult to understand in earlier versions of Zowe. Check the parmlib member mentioned in the preceding messages for errors.

IZPPI0205E **Validation unsuccessful, it returned code $RC for $ `{ZWE_components_izp_dataset_parmlib}($membername)`.**

Explanation:
Zowe config manager ran but detected problems in ZWEYAML or main Zowe yaml file.

System action:
UMS did not start.

User response:
Correct problems based on error messages and restart the UMS component.

IZPPI0206E **Unable to access Zowe values. See preceding messages for details.**

Explanation:
Zowe installation lifecycle script was unable to access Zowe environment variables or unable to generate those variables through the Zowe configuration manager.

System action:
UMS installation as Zowe plug-in terminates.

User response:
Determine corrective action. You can launch Zowe installation script with different parameters or regenerate `IZPIPLUG` JCL job using `IZPGENER`.

IZPPI0207E **Unified UI configuration cannot modify Zowe workspace ($action).**

Explanation:
During Zowe launch of UMS, Unified UI configuration values are stored in the Zowe workspace. A number of actions are taken. If one action fails, this message will indicate which action failed. Failures are likely to be permission related.

System action:
Unified UI configuration is incomplete. UMS may not function.

User response:
Determine what permissions need to be adjusted, either in files or in granting access to the started task user.

IZPPI0210E **${experienceRuntime} does not contain any folders with an opt/bin folder. Possibly, the older version is incompatible with UMS 1.2.**

Explanation:
UMS 1.2 requires experiences that are packaged for this version. If an older experience version is provided in the `components.izp.experiences` array, it will be detected during the Zowe validation lifecycle script and this message will be printed.

System action:
Zowe will not start due to failed validation.

User response:
Provide the location of the compatible experience version.

IZPPI0211E **UMS must not be in `components.izp.experiences`. Found *<directory>* within *<parent directory>*.**

Explanation:
UMS is not an experience and must not be included in the experiences list.

System action:
The validation script returns an error and Zowe does not start.

User response:
Remove the UMS line from the experiences list in the `ZWEYAML` parmlib member.

IZPPI0212E ***<experience>* directory improperly specified in `components.izp.experiences`. Should be *<experience parent directory>*.**

Explanation:
Experiences need to be specified as their parent directory, as some experiences have multiple sub directories.

System action:
The validation script returns an error and Zowe does not start.

User response:
Modify the experiences list in ZWEYAML PARMLIB member as specified.

IZPPI0213E **Member ZWEYAML is missing from data set.**

Explanation:
`IZPSYNCY` or `IZPMIGRA` was unable to read `ZWEYAML` from the PARMLIB location.

System action:
The `IZPSYNCY` or `IZPMIGRA` operation terminates.

User response:
Update `IZPSYNCY` or `IZPMIGRA` JCL with the corrected PARMLIB location and resubmit it.

IZPPI0214E **Problem parsing dataset *<location>*.**

Explanation:
`IZPSYNCY` or `IZPMIGRA` encountered an invalid YAML in the PARMLIB(ZWEYAML) or UMS 1.1 input PARMLIB data set.

System action:
The `IZPSYNCY` or `IZPMIGRA` operation terminates.

User response:
Correct the input data set YAML format and resubmit it.

IZPPI0215E **Problem parsing dataset *<location>* for update.**

Explanation:
`IZPSYNCY` or `IZPMIGRA` encountered an invalid YAML in the input of SMP/E location SIZPPARM.

System action:
The `IZPSYNCY` or `IZPMIGRA` operation terminates.

User response:
Rerun `IZPGENER` to update schema locations or provide alternative input locations in JCL and resubmit it.

IZPPI0216E **Invalid number of arguments to : *<number>*.**

Explanation:
An internal error is encountered during `IZPSYNCY` or `IZPMIGRA`.

System action:
The `IZPSYNCY` or `IZPMIGRA` operation terminates.

User response:
See details of the error and contact IBM Software Support.

IZPPI0217E **Error accessing *<dataset>*.**

Explanation:
`IZPGENER` was unable to open the *<dataset>* data set.

System action:
The `IZPGENER` operation terminates.

User response:
Verify that HLQ for the named data set is correctly specified in `IZPGENER` job and/or `ZWEYAML` PARMLIB member.

IZPPI0218E **Failed to record members from *<dataset>*.**

Explanation:

IZPGENER was unable to determine the member list of the *<dataset>* data set.

System action:
The IZPGENER operation terminates.

User response:
Verify that HLQ for the named data set is correctly specified in the IZPGENER job and/or ZWEYAML PARMLIB member, and if it is a partitioned data set.

IZPPI0219E Edit failed. Stopping at *<member name>*.

Explanation:
IZPGENER was unable to modify the *<member name>* in the output data set.

System action:
The IZPGENER operation terminates.

User response:
Verify that the ISPF 'edit' macro is available and the ISPF libraries are correctly specified in IZPGENER JCL.

IZPPI0220E ConfigMgr could not *<action>*.

Explanation:
IZPGENER invoked Zowe configuration manager for the named operation during YAML configuration validation. The configuration manager encountered an error while performing the action. This is due to errors invoking the configuration manager or invalid YAML configuration.

System action:
The IZPGENER operation terminates.

User response:
Verify zowe.setup.dataset.loadlib in STEPLIB definition of IZPGENER. For validation errors, determine and correct problematic YAML specifications.

IZPPL messages

Messages that are issued when you manage subsystems, environments, teams, applications, instances, and so on, have the format IZPPLnnnnx.

IZPPL0001E Object *variable1* with id *variable2* already exists.

Explanation:
Policy object with id already exists.

System action:
There already exists object with this id. Hence the intended operation is not successful.

User response:
Contact system administrator to find out more about the operation status.

IZPPL0001I Provisioning still in progress.

Explanation:
Instance provisioning is still in progress.

System action:
No further operations can be performed on the instance, and it is not ready for use.

User response:
Wait until provisioning reaches an end state - error or complete.

IZPPL0001W Objects are unavailable due to provisioning error.

Explanation:
Instance is not ready for use due to unknown reasons.

System action:
No further operations can be performed on the instance, and it is not ready for use.

User response:
Contact system administrator to find out more about the instance status.

IZPPL0002E Object *variable1* with name *variable2* already exists.

Explanation:
Policy object with name already exists.

System action:
There already exists object with this name already exists. Hence the intended operation is not successful.

User response:
Verify the name of the object and retry.

IZPPL0002W Objects have already been deprovisioned.

Explanation:
No operations can be performed on instance as instance is already deprovisioned.

System action:
Objects cannot be fetched from a deprovisioned instance.

User response:
Objects cannot be fetched from a deprovisioned instance.

IZPPL0003E Object *variable1* with id *variable2* does not exist.

Explanation:
Policy object with given id does not exist.

System action:

The policy object with the given id does not exist.

User response:
Verify the id of the object and retry.

IZPPL0004E Object *variable1* with name *variable2* does not exist.

Explanation:
Policy object with given name does not exist.

System action:
The policy object with the given name does not exist.

User response:
Verify the name of the object and retry.

IZPPL0005E Object *variable1* with id *variable2* can't depend on self.

Explanation:
A cyclic dependency has been encountered. Object *variable1* with id *variable2* cannot depend on self.

System action:
A cyclic dependency has been encountered. The intended action cannot be completed.

User response:
Retry the operation after removing the circular dependency.

IZPPL0006E Invalid request parameter: *variable1*

Explanation:
The value for request parameter *variable1* is invalid.

System action:
The operation cannot proceed as an invalid request parameter has been specified.

User response:
Verify the request parameter and retry the operation.

IZPPL0008E Objects are still under preparing.

Explanation:
Objects are busy.

System action:
The object cannot be used as it is still being processed.

User response:
Objects are busy, please try again later.

IZPPL0009E Object *variable1* with id *variable2* is still used by *variable3*.

Explanation:
The object cannot be deleted or deprovisioned because it is being referenced by *variable 3*.

System action:
The delete operation cannot be completed.

User response:
Ensure that all dependencies are removed and retry the operation.

IZPPL0010E Object *variable1* with name *variable2* could not be updated.

Explanation:
Failed to update *variable1* with name *variable2*.

System action:
The update operation has failed.

User response:
Contact system administrator to find out more about the update status.

IZPPL0013E Field name *variable1* incorrect value *variable2*

Explanation:
Variable2 is an invalid value for *variable1*.

System action:
The validation failed because of an invalid input field.

System action:
Retry the operation after correcting the input field.

IZPPL0014E Exceeds instance limit quota for *variable1*. (max = *variable2*)

Explanation:
Instance *variable1* has exceeded its maximum size of *variable2*.

System action:
The operation could not be completed as one of the fields has exceeded the prescribed limits.

User response:
Retry the operation after correcting the input fields to be within limits.

IZPPL0015E Objects from multiple subsystems are not supported.

Explanation:
Objects across subsystems are not supported.

System action:
Objects from multiple subsystems cannot exist in the same application and hence the creation failed.

User response:
Retry the operation after retrieving objects from the same subsystem in your application.

IZPPL0016E No subsystems are available for the environment: *variable1*

Explanation:
No subsystems are available for selected environment *variable1*.

System action:
The operation cannot be completed because the environment has no subsystems to process.

User response:
Contact your system administrator.

IZPPL0017E **Unable to create security profile for the team *variable1***

Explanation:
Security profile for team cannot be created and hence the team creation fails.

System action:
The team cannot be created as the security profile creation failed.

User response:
Contact your system administrator to find out more details about the security profile creation.

IZPPL0018E **Unable to delete security profile for the team *variable1***

Explanation:
Security profile for team cannot be deleted and hence the team deletion fails.

System action:
The team cannot be deleted as the security profile deletion failed.

User response:
Contact your system administrator to find out more details about the security profile deletion.

IZPPL0019E **The subsystem with id *variable1* does not match name *variable2* and type *variable3***

Explanation:
Server summary is not consistent.

System action:
The server summary information is not correct.

User response:
Contact your system administrator.

IZPPL0020E **Application object *variable1* of type *variable2* not found on subsystem *variable3* named *variable4*.**

Explanation:
Application object could not be found.

System action:
The application object being retrieved was not found on the subsystem.

User response:
Contact your system administrator or retry the operation with correct parameters.

IZPPL0021E **Invalid JSON data: *variable1***

Explanation:
The input JSON for the operation could not be parsed successfully.

System action:
The operation did not complete as the input JSON could not be parsed successfully.

User response:
Contact your system administrator or retry the operation with correct parameters.

IZPPL0022E **This line exceeds the maximum 72 characters per line: *variable1***

Explanation:
The input line has crossed the maximum length of 72 characters.

System action:
The operation did not complete as the input line length was more than the prescribed 72 characters.

User response:
Retry the operation after correcting the input line.

IZPPL0023E **Editor will not save empty DDL**

Explanation:
Editor does not allow empty DDL to be saved.

System action:
Editor does not allow empty DDL to be saved.

User response:
You are not allowed to save an empty DDL. Correct the DDL and retry.

IZPPL0024E **Report not found for jobid: *variable1***

Explanation:
The report with the given id was not found.

System action:
The operation could not go through as the report could not be fetched.

User response:
Contact your system administrator on why the report fetch failed or correct the report id and retry.

IZPPL0025E **Application *variable1* is not ready. status: *variable2***

Explanation:
The application with given id is not ready.

System action:
The operation could not be completed as the application with given id is not ready.

User response:
Contact your system administrator on why the application with given id could not be fetched.

IZPPL0026E **Instance exists for this id: *variable1***

Explanation:
The instance with the given id exists.

System action:
The subsystem deletion cannot be completed as an instance with the given id exists.

User response:
Try to clean up any instances with the given id and then retry the operation.

IZPPL0027E **Invalid timeout format: *variable1*. Valid examples: '10', '(30,15)', 'NOLIMIT', 'MAXIMUM'**

Explanation:
The timeout has been specified in an incorrect format.

System action:
The operation could not be completed as the timeout has been specified in an incorrect format.

User response:
Correct the timeout format and retry the operation.

IZPPL0028E **User does not have the correct privileges for this action. Please contact the administrator.**

Explanation:
You do not have privileges for this operation.

System action:
The operation could not be completed because you are not authorized for this operation.

User response:
You do not have the correct privileges for this action. Contact the administrator.

IZPPL0029E **Specified data set *variable1* could not be found**

Explanation:
The data set could not be found.

System action:
The operation could not be completed because the specified data set could not be found.

User response:
Contact your system administrator.

IZPPL0030E **Do not specify a member for this data set: *variable1***

Explanation:
Data set name cannot contain a member name.

System action:
The Admin OC operation failed because the data set name contained a member name.

User response:
Contact your system administrator (mention a data set without a member name).

IZPPL0032E **User is not a reviewer.**

Explanation:
The user is not a reviewer.

System action:
The operation could not be completed as the user is not a reviewer.

User response:
Contact your system administrator to check the user authorities.

IZPPL0033E **This pull request has not been approved yet.**

Explanation:
The pull request has not been approved yet.

System action:
The operation could not be completed as the pull request review has not been completed yet.

User response:
Contact your system administrator and check on the pull request review status.

IZPPL0034E **There is already a pull request for the same branches by id: *variable1*.**

Explanation:
There is already an existing pull request for the same branches.

System action:
The operation could not be completed because there is already an existing pull request for the same branches.

User response:
Contact your system administrator and check on the pull request status.

IZPPL0036E **The following application(s) rely on the Team: *variable1***

Explanation:
The team is already referenced in another instance or application.

System action:
The team cannot be deleted as it is being referenced in another instance/application.

User response:
Contact your team administrator to ensure there are no references to the given team and then retry the operation.

IZPPL0037E **requiredApp *variable1* contains object(s) from a different subsystem**

Explanation:
Required applications cannot be from a different subsystem.

System action:

The operation could not be completed as the required applications for an application cannot be from a different subsystem.

User response:
Check your application to ensure it does not have required applications from a different subsystem.

IZPPL0038E Instance objects from required applications cannot be modified

Explanation:
Instance objects from required applications cannot be modified.

System action:
The operation could not complete because instance objects from a required application cannot be modified.

User response:
Do not try to edit or modify instances of the required application. This operation is not supported currently.

IZPPL0039E New Db2 object must be from the same subsystem as the instance

Explanation:
New Db2 object must be from the same subsystem as the instance.

System action:
The operation could not be completed as the objects that are to be created need to be part of the same subsystem.

User response:
Check your application to ensure it does not have objects from a different subsystem.

IZPPL0040E Db2 object *variable1* of type *variable2* does not exist on subsystem *variable3*. Please provide DDL

Explanation:
The DDL for the object was not provided.

System action:
The operation could not be completed as the DDL was not supplied.

User response:
Please supply the DDL for the object and retry the operation.

IZPPL0041E A Db2 object *variable1* of type *variable2* already exists on subsystem *variable3*. Clear the 'ddl' input field to add an existing object.

Explanation:
Cannot add the existing object to an instance when the ddl is supplied. Either a non-existing object should be supplied or empty DDL should be provided to add existing object to an instance.

System action:
The operation could not be completed to add the new object.

User response:
Either add the object to an instance which does not exist or clear the ddl from request payload while adding a new object to an instance.

IZPPL0042E Site rules violation or syntax error detected

Explanation:
A site rule violation was detected.

System action:
The operation could not be completed as a site rule violation was detected.

User response:
Try to resolve the site rule violation and retry the operation.

IZPPL0043E DDL did not match specified Db2 object *variable1* of type *variable2*

Explanation:
Total number of on-source objects cannot be greater than 1.

System action:
The operation could not be completed as the count of on-source objects are not exactly 1.

User response:
Contact your system administrator.

IZPPL0044E A Db2 object *variable1* of type *variable2* already exists in another *variable3*

Explanation:
The Db2 object exists as part of another application.

System action:
The operation could not be completed because the object that is being created is already part of another application.

User response:
Try to ensure that your objects are not part of any other application. This is currently not supported.

IZPPL0045E A Db2 object of type *variable1* cannot be added to an instance

Explanation:
A new database or storage group cannot be created.

System action:
The operation could not be completed as a new database or storage type cannot be created.

User response:

Try to ensure you are not trying to create any database or storage group as part of your application.

IZPPL0046E A newly added Db2 object of type *variable1* must have a qualifier valid for this instance's application. Valid qualifiers: *variable2*

Explanation:
Cannot work with specified object types.

System action:
The operation could not be completed as certain disallowed object types are being used.

User response:
Try to ensure you are working with only allowed object types.

IZPPL0047E Objects named *variable1* and *variable2* of type *variable3* already exist on this subsystem. Specify a different name.

Explanation:
Cannot provisioning a new object when it already exists on the subsystem.

System action:
Provisioning cannot be done with given schema/ database.

User response:
Retry the instance provisioning.

IZPPL0048E String delimiter must be an apostrophe or a quotation mark, not *variable1*.

Explanation:
String delimiter must be an apostrophe or a quotation mark.

System action:
The operation failed as an incorrect string delimiter was used.

User response:
Use an apostrophe or a quotation mark as string delimiters and retry.

IZPPL0049E Decimal point must be a period or a comma, not *variable1*.

Explanation:
Decimal point must be a period or a comma.

System action:
The operation failed as an incorrect decimal point was used.

User response:
Use a period or a comma as the decimal point and retry.

IZPPL0050E Escape character must be an apostrophe or a quotation mark, not *variable1*.

Explanation:
Escape character must be an apostrophe or a quotation mark.

System action:
The operation failed as an incorrect escape sequence was used.

User response:
Use an apostrophe or a quotation mark as escape sequence and retry.

IZPPL0051E The Team with name: *variable1* relies on current Environment.

Explanation:
There is a team associated with the environment and hence environment cannot be deleted.

System action:
Delete environment operation ends abruptly.

User response:
Verify that there is no associated team to the environment and retry deleting the environment.

IZPPL0052E The *variable1* operation for policy object *variable2* is not allowed.

Explanation:
This is an illegal operation and hence not allowed.

System action:
Operation ends abruptly until valid input is provided.

User response:
Once a pull request is created, it cannot be deleted.

IZPPL0053E The value of *variable2* for query parameter *variable1* is not allowed.

Explanation:
Invalid value of the query parameter has been passed and hence the operation has been terminated.

System action:
Operation ends abruptly until a valid query parameter value is provided.

User response:
Verify that correct query parameter value has been passed and retry this operation.

IZPPL0054E The query parameter *variable1* is required.

Explanation:
The query parameter is missing or blank and hence the operation has been terminated.

System action:

Operation ends abruptly until a valid query parameter value is provided.

User response:
Verify that correct query parameter value has been passed and retry this operation.

IZPPL0056E Id *variable1* duplicated in list.

Explanation:
Duplicate environments cannot be associated with a team and hence team cannot be created or updated.

System action:
Team cannot be created or updated until one valid environment is provided.

User response:
Verify that duplicate environments are not being associated with a team and retry creating or updating the team.

IZPPL0057E Field *variable1* length must be in-range [*variable2*, *variable3*]

Explanation:
A string with invalid length has been passed.

System action:
Operation ends abruptly until a string with valid length has been passed.

User response:
Pass a string with valid length and retry the operation.

IZPPL0058E Team with id *variable1* must be linked to at least 1 Environment. Cannot delete Environment with id *variable2*

Explanation:
Not used anywhere.

System action:

User response:

IZPPL0059E Instance with id *variable1* has an active pull request.

Explanation:
An active pull request already exists for the instance and hence a new pull request cannot be created for this instance.

System action:
Pull request creation is ended abruptly.

User response:
Verify that there are no active pull requests for the instance and then try re-creating a pull request for the instance.

IZPPL0060E *variable1* and *variable2* are mutually exclusive

Explanation:
While invoking apply changes for applications, both SRCIN(source ddl), COMMIT(commit ID) should not be specified.

System action:
Apply DDL changes operation is ended abruptly until valid options are passed.

User response:
Verify that valid request payload is provided, and then retry apply changes.

IZPPL0062E Limit *variable1* of value *variable2* cannot exceed Environment *variable3* of value *variable4*

Explanation:
The instance provision limit for team or user during team creation should be greater than 0 and should not exceed the instance provision limit set for the environment. Also, the instance provision limit for user should be less than that of team.

System action:
Team creation operation ends abruptly until valid instance provision limit for user and team is set.

User response:
Verify that proper instance provision limit is set for team and user and retry team creation.

IZPPL0063E Subsystem with ID *variable1* must be in a completed state

Explanation:
The subsystem has to be in completed/ready state before an application is created or updated.

System action:
Application creation ends abruptly until subsystem is in completed/ready state.

User response:
Verify that subsystem is in complete/ready state and then retry application creation.

IZPPL0064E Could not parse value *variable1* as type *variable2*.

Explanation:
Subsystem registration failed due to invalid subsystem id.

System action:
Subsystem registration ended abruptly.

User response:
Contact system administrator to find out more about the subsystem registration failure.

IZPPL0065E Cannot delete subsystem *variable1* while it is installing.

Explanation:
Subsystem cannot be deleted while it is getting registered.

System action:
Subsystem deletion ends abruptly because its registration is still in progress.

User response:
Verify that the subsystem's registration status is not in progress, and then retry deleting subsystem.

IZPPL0066E ***variable1* violates the environment rule *variable2* : *variable3***

Explanation:
Instance provision failed because one or more rules assigned to an environment are violated.

System action:
Instance provision is ended abruptly until all rules assigned to an environment are adhered to.

User response:
Verify that all environment rules are adhered to, and then retry instance provision.

IZPPL0068E **Could not assign user *variable1* to team *variable2* with role *variable3*, reason *variable4***

Explanation:
The user with this role could not be assigned to the team.

System action:
User assignment to the team ended abruptly.

User response:
Contact system administrator to find out more about why user could not be assigned to the team.

IZPPL0069E **User does not have the correct privileges to assign *variable1* to team *variable2***

Explanation:
The logged-in user does not have correct privileges to update the user.

System action:
User update operation is ended abruptly until correct privileges are available to the logged-in user.

User response:
Verify that logged-in user has correct privileges to update user, and then retry updating the user.

IZPPL0074E **The Instance with name: *variable1* relies on current Environment.**

Explanation:
The environment cannot be deleted because there is an active instance using this environment.

System action:
Delete Environment operation ended abruptly because there is an active instance using this environment.

User response:
Verify that there is no instance using the environment and retry deleting the environment.

IZPPL0075E **Modify the Schema Provisioning rule. Application schema length is *variable1* (schema - *variable2*), and provisioning pattern is *variable3*.**

Explanation:
Instance could not be provisioned because the instance schema length is greater than the length allowed by Db2 (128 characters).

System action:
Instance provision is abruptly ended until schema with valid length is provided.

User response:
Verify that instance schema with length less than 128 is provided, and then retry instance provision.

IZPPL0077E **Operator *variable1* is no longer supported for a new assignment. You can use operator *[variable2]* for object *variable3* and attribute *variable4*.**

Explanation:
The simple site rule could not be created because it contains unsupported verification type.

System action:
Simple site rule creation is abruptly ended until valid verification type is provided in it.

User response:
Verify that supported verification type is provided in the rule and retry simple rule creation.

IZPPL0078E **Operator *'operator'* is no longer supported for a new assignment. Usage: You can use operators [*operators*] for object *'obj'* and attribute *'attr'*.**

Explanation:
The simple site rule could not be created because it contains more than one unsupported verification type.

System action:
Simple site rule creation is abruptly ended until valid verification type is provided in it.

User response:
Verify that supported verification types are provided in the rule and retry simple rule creation.

IZPPL0079E **Attribute *'attr'* is no longer supported for object *'obj'*. Usage: *text***

Explanation:
The simple site rule could not be created because it contains an unsupported attribute.

System action:
Simple site rule creation is abruptly ended until a valid attribute is provided in it.

User response:
Verify that a supported attribute is provided in the rule and retry simple rule creation.

IZPPL0082E **Invalid *'text'* of value *'instance_limit_value'*. Provisioned instance limits must be between -1 and 65535**

Explanation:
The environment cannot be created or updated because an invalid value of provisioned instance limit has been provided.

System action:
Environment creation or update operation is abruptly ended until valid provision instance limits are provided.

User response:
Verify that a valid provisioned instance limit of -1 and 65535 has been provided, and then retry creating or updating environment.

IZPPL0083E **Unable to create team *'team_name'* because the limit has been reached. Delete unused teams or increase the size of your TEAMLIST dataset before creating more.**

Explanation:
The team could not be created because the `TEAMLIST` data set limit has reached.

System action:
Team creation operation is abruptly ended until `TEAMLIST` data set limit is increased.

User response:
Verify that sufficient `TEAMLIST` data set size is available by increasing the limit or deleting the unused teams and then retry team creation.

IZPPL0084E **Invalid character, the characters '*chars*' are not allowed in field '*field_name*'**

Explanation:
The name or ID contains invalid character "/" and hence the object cannot be created or updated.

System action:
The object creation is abruptly ended until the invalid character "/" is removed from the name and ID.

User response:
Verify that the name and ID does not contain invalid character "/" and retry the create/update operation.

IZPPL0085E **Unable to initialize DB2 Administration and Object Comparison tool Job Manager.**

Explanation:
DDL could not be generated because Db2 Administration and Object Comparison tool Job Manager could not be initialized.

System action:
DDL generation is abruptly ended until Administration and Object Comparison tool Job Manager is properly initialized.

User response:
Contact your system administrator to find out more details about the Administration and Object Comparison tool Job Manager initialization failure.

IZPPL0086E **Fail to process the request because data *'file_name'* is corrupted.**

Explanation:
The policy object is not found and hence unable to process the request.

System action:
The request is ended abruptly because the policy object could not be located.

User response:
Contact your system administrator to find out more details about the request failure.

IZPPL0087E **Unable to find job with jobid: *'job_id'***

Explanation:
Not able to generate a job report because the job with this ID does not exist.

System action:
Job report cannot be generated until a valid job ID is provided.

User response:
Verify that a valid job ID is provided, and then retry generating the job report.

IZPPL0088E **Instance(s) exist for *'appl_name'***

Explanation:
There is an active instance and therefore the application cannot be deleted.

System action:
Application deletion is ended abruptly until all active instances are deprovisioned.

User response:
Verify that all active instances are deprovisioned, and then retry the operation.

IZPPL0089E **Specified subsystem *variable1* is not active or not found.**

Explanation:
The user tries to manage a subsystem *variable1*, but it's not up and running at the moment.

System action:
The operation is not performed.

User response:
Start the subsystem.

IZPPL0090E **Active subsystem does not exist in environment: *variable1*.**

Explanation:
The user has made a provisioning request, but no subsystems of the environment *variable1* are active and running.

System action:
The operation is not performed.

User response:
Start a subsystem that is assigned to the environment.

IZPPL0091E **Specified subsystem *variable1* is unsupported version.**

Explanation:
The product version of the specified subsystem *variable1* is unsupported.

System action:
The operation is not performed.

User response:
Contact your system administrator.

IZPPL0092E **Specified subsystem *variable1* is not discovered by subsystem discovery.**

Explanation:
The user tries to manage a subsystem, but the specified subsystem *variable1* has not been discovered.

System action:
The operation is not performed.

User response:
Start the subsystem if it has not started.

IZPPL0093E **These changes are not mergeable due to merge conflicts. You need to deprovision this instance and redo your changes to resolve the change conflict.**

Explanation:
There are merge conflicts and hence the pull request cannot be merged.

System action:
Pull request merge operation is ended abruptly until the merge conflicts are resolved.

User response:
Verify that all the merge conflicts are resolved, and then retry the pull request merge operation.

IZPPL0094E **Synchronization still in progress.**

Explanation:
The instance DDL is out of sync with respect to application and its synchronization is in progress.

System action:
No operation is allowed on the instance because it is getting synchronized with the application.

User response:
Verify that the synchronization is completed successfully, and then retry the operation.

IZPPL0095E **Objects are unavailable due to synchronization error.**

Explanation:
The instance is in synchronization error state and hence the DDL is not available.

System action:
Getting object DDL is ended abruptly until instance is in synchronization error state.

User response:
Verify that instance is not in synchronization.

IZPPL0096E **Field *variable1* must be in the range [*variable2 - variable3*].**

Explanation:
The specified value of the field *variable1* is not in the permitted range of values. The value must be between *variable2* and *variable3*. For example, the port number of IMS Connect.

System action:
The operation is not performed.

User response:
Specify a valid value for the field.

IZPPL0097E **Cannot process the request for subsystem *'subsystem_name'* while previous operation is not complete.**

Explanation:
Subsystem cannot be updated or removed because it is not in ready state.

System action:
Subsystem update or removal operation is ended abruptly until it is in ready state.

User response:
Verify that the subsystem is in ready state, and then retry subsystem operation.

IZPPL0098E **You cannot raise a pull request when the instance is not in a ready state.**

Explanation:
The instance is not yet in ready state. Hence pull requests cannot be created for this instance.

System action:
Pull request creation for this instance abruptly ended because the instance is not yet in ready state.

User response:
Verify that the instance is in ready state, and then retry pull request creation.

IZPPL0099E **The last attempt to apply the DDL changes has not failed. Hence you cannot resume applying DDL changes.**

Explanation:
The last attempt to apply the DDL changes has not failed. Hence you cannot resume applying DDL changes.

System action:
Resume Apply changes ended abruptly because the last attempt to apply changes has not failed.

User response:
Verify last attempt to apply DDL changes is in 'Apply Error' state and then resume apply changes.

IZPPL0100E **You cannot edit the DDL when the DDL apply changes has failed.**

Explanation:
Edit DDL is not allowed when DDL apply changes has failed and instance status is in 'Apply error'.

System action:
Edit DDL is ended abruptly until DDL apply changes succeed.

User response:
Verify DDL apply changes is successful, and then retry editing DDL.

IZPPL0131E **The number of values specified in *variable1* parameter exceeds the maximum allowable number. The maximum number is *variable2*.**

Explanation:
The parameter *variable1* can have multiple values, but the user assigns more than the permitted number of values for that variable. The maximum number of values that can be assigned to that parameter are shown as *variable2*.

System action:
The operation is not performed.

User response:
The number of values that you assign to the parameter *variable1* should be less than the permitted number of values specified in *variable2*.

IZPPL0134E **You cannot update the team that owns this application because the active pull request '%1$s' exists for this application.**

Explanation:
There are active pull requests for the application.

System action:
You cannot update the team that owns this application because the active pull request '<pull request numbers>' exists for this application.

User response:
Merge or decline the pull request and try again.

IZPPL0212E **Failed to get linkname for Db2 subsystem '%1$s', reason '%2$s'**

Explanation:
You did not run the full install on the `SYSPROC.ADMIN_COMMAND_DB2` stored procedure. In particular, the stored procedure uses a specific table to process the `-DIS DDF` command, and you did not create that table.

System action:
UMS failed to get additional DDF information for the subsystem.

User response:
UMS calls the Db2 supplied stored procedure `ADMIN_COMMAND_DB2` for the `-DISPLAY DDF` command. Make sure `ADMIN_COMMAND_DB2` stored procedure is configured and operational. The required global temporary tables for the stored procedure must be created. For example, `SYSIBM.DDF_CONFIG`. For details, see the Db2 sample job `SDSNSAMP(DSNTESR)`.

IZPPL0236E **Object 'snapshot' with the name '%1$s' already exists.**

Explanation:
Snapshot name should be unique across the particular instance.

System action:
The request will not be processed because snapshot name is not unique across the particular instance.

User response:
Specify a unique snapshot name and submit the request.

IZPPL0238E **There are uncommitted changes in 'instance' id: '%1$s'. Instance name: '%2$s'.**

Explanation:

Snapshot can be created only when there are no uncommitted changes in the instance.

System action:
The request will not be processed because the instance has uncommitted changes.

User response:
Apply the uncommitted changes and try creating the snapshot.

IZPPL0243E There is an exception in the snapshot request: '%1$s'.

Explanation:
An exception occurred while getting the job id report.

System action:
The current request will not be processed.

User response:
Retry the operation after some time.

IZPPL0244E There is an error in copying object ddl files to '%1$s'.

Explanation:
An error occurred while copying the ddl files.

System action:
The current request will not be processed.

User response:
Refer the log file for issues. Fix the issue and retry the request.

IZPPL0246E Snapshot is not in a ready state to restore in instance id: '%1$s'. Current status : '%2$s'.

Explanation:
Snapshot can be restored only when the snapshot status is READY.

System action:
The request will not be processed because the snapshot is not in ready state.

User response:
The current snapshot is not usable. You need to use another snapshot for the restore operation.

IZPPL0248E Application ownership cannot be changed.

Explanation:
Ownership of the application cannot be updated while it is updating or initiating.

System action:
The request will not be processed until the application is in a ready or complete state.

User response:
Wait until the application is in a ready state and try again.

IZPPL0249I Some instances are in flight status.

Explanation:
Application ownership cannot be changed because some application instances are in progress state.

System action:
The request will not be processed until application instances are in complete state.

User response:
Check the instance statuses and wait until the instances that are in progress state transition to a ready state.

IZPPL0250E Failure occurred while changing application settings.

Explanation:
Failed to save application setting changes.

System action:
The request will not be processed because the application YAML file from the policy folder has been deleted or the YAML file is locked.

User response:
Contact the system administrator to find out more about the application settings change failure.

IZPPL0251I 'To Team (name)' is not assigned to all environments for the instances owned by 'From Team (name)'.

Explanation:
'<To team>' is not assigned to all environments for the instances owned by '<From team>'.

System action:
Only the application's team gets changed, and the instance's team remains as it is. The instances will be in a read-only state. The user gets an information message on the GUI.

User response:
Add the environments which are missing from the old team to the new team before proceeding.

IZPPL0252I Application team change operation is in progress. No modifications are allowed until the operation gets completed. Application name: 'application name' and instance name: 'instance name'.

Explanation:
An application team change operation is in progress. No modifications are allowed until the operation is completed.

System action:

Instance modification is not allowed as an application team change operation is in progress. The user gets an information message in the instance details screen.

User response:
Try again after the ownership change has been completed.

IZPPL0253E **Instance Adoption Failed. Application name: '%1$s' From team name: '%2$s' To team name: '%3$s'. Reason for failures: '%5$s'.**

Explanation:
While instances are in progress state, if you try to adopt the instances, you get this exception.

System action:
The request will not be processed because the instance is in progress state.

User response:
When the instance gets into a ready state, try again to adopt the instance.

IZPPL0254E **There is no team administrator in given To team. Team name: 'Team name'.**

Explanation:
When you try to change the application's team from the 'From' team to the 'To' team that does not have an administrator, this exception occurs.

System action:
The request will not be processed because the team does not contain any team administrators.

User response:
Add at least one team administrator to the team and try again.

IZPPL0257E **Objects are unavailable due to a snapshot restore error.**

Explanation:
While restoring the snapshot, an error occurred. Instance is in snapshot restore error status.

System action:
The current request will not be processed.

User response:
This snapshot is not usable for the restore operation. Use another snapshot.

IZPPL0258E **Pass the valid snapshot id to deprovision all snapshots in an instance. Instance id: '%1$s'. Instance name : '%2$s'.**

Explanation:
Valid snapshot id is not provided to delete a snapshot.

System action:
The current request will not be processed.

User response:
Confirm the valid snapshot id is provided and retry the request.

IZPPL0259E **Please pass the valid snapshot identifier or/and instance identifier**

Explanation:
The snapshot identifier or/and instance identifier entered to restore the snapshot is invalid.

System action:
No further operations can be performed on the snapshot.

User response:
Verify the snapshot identifier or/and instance identifier.

IZPPL0260E **Invalid request parameter: '%1$s'. Users %2$s are not members of team %3$s that owns application %4$s**

Explanation:
Validate the reviewer list while creating the pull request.

System action:
Pull request cannot be created as reviewers are not a part of the application owning team.

User response:
Additional reviewers should be a part of the application owning team.

IZPPL0261E **The object is not editable because the object '%1$s' is not in home application or the application owning team is different from instance owning team**

Explanation:
Checks whether the object is editable.

System action:
No further operations can be performed as object is not editable.

User response:
Check your instance to ensure there are no objects from different applications.

IZPPL0262E **The mask not defined for object '%1$s' of type '%2$s' in instance '%3$s'**

Explanation:
Checks whether the mask is defined in the instance or not.

System action:

No further operations can be performed as the mask is not defined for the application object.

User response:
Define the mask for the application objects.

IZPPL0263E **The object '%1$s' of type '%2$s' cannot be added to this application because home application of the required object '%3$s' of type '%4$s'**

Explanation:
Triggers and indexes are accompanied by tables. If the table is added as a non-core in the instance, then trigger and index are also added as a non-core object and cannot be added as a core object.

System action:
No further operations can be performed as the required objects of the trigger are not present in the home application.

User response:
Add index or trigger as a core or non-core object based on the table defined in the instance.

IZPPL0264E **Home application id not defined for the object '%1$s' of type '%2$s' in %3$s**

Explanation:
Home application ID for an object is not defined in an instance.

System action:
No further operations can be performed as the home application ID is not defined for Db2 Object.

User response:
Update home application ID using Application-Update API(PUT API).

IZPPL0265E **Add an existing object from target not supported for Db2 object '%1$s' of type '%2$s' in instance '%3$s'**

Explanation:
Addition of a database using add via target option in an instance is not supported.

System action:
No further operations can be performed as addition of database through add via target option is unsupported.

User response:
Do not add database in the instance using add via target option.

IZPPL0266E **To provision the instance or snapshot, resolve one or more errors**

Explanation:
Errors that are identified while provisioning instance or snapshot.

System action:
Instance or snapshot cannot be provisioned as there are multiple errors.

User response:
Resolve all the errors in order to provision the instance or snapshot.

IZPPL0267E **Input object type '*%1$s*' does not match the actual object type '*%2$s*' in instance '*%3$s*'**

Explanation:
A mismatch between the input and actual object type in the catalogue table, when adding an object using the source or destination option.

System action:
Object cannot be added, as the input and actual object type does not match.

User response:
Enter the correct value for the input object type.

IZPPL0268E **The target application '*%1$s*' specified is not applicable for the instance '*%2$s*'. The target application must be the same as the instance provisioning application ID, or it must match one of the instance provisioning application associated application IDs.**

Explanation:
Target application ID should be aligned with instance provisioning application IDs.

System action:
Objects cannot be added using the DDL option, as target application is not aligned with instance provisioning applications.

User response:
Ensure that the target application is aligned with the instance provisioning applications.

IZPPL0269E **The target application '*%1$s*' specified is not applicable for the Db2 Object '*%2$s*' in instance '*%3$s*'. The target application must be the same as the instance provisioning application ID, or it must match one of the instance provisioning application associated application IDs.**

Explanation:
Target application ID should be aligned with instance provisioning application IDs.

System action:
Object cannot be added using the DDL option, as target application is not aligned with instance provisioning applications.

User response:
Ensure that the target application is aligned with the instance provisioning applications.

IZPPL0270E **DDL syntax error encountered: '*%1$s*'**

Explanation:
DDL syntax error.

System action:
No further operations can be performed as the DDL has syntax errors.

User response:
Verify DDL and rectify the syntax errors.

IZPPL0271E **DDL site rule error encountered: '*%1$s*'**

Explanation:
DDL site rule error.

System action:
No further operations can be performed as the DDL has site rule errors.

User response:
Verify DDL and rectify the site rule errors.

IZPPL0272E **Input object of type '*%1$s*' does not support version '*%2$s*' as an attribute**

Explanation:
Validates the supported attributes for the input object (for example, version for stored procedures) and identifies that the input object of type table contains the version attribute.

System action:
No further operations can be performed as the version attribute is not supported by the Db2 object.

User response:
Remove the version attribute and try again.

IZPPL0273E **Input object of type '*%1$s*' does not support specific name '*%2$s*' as an attribute**

Explanation:
Validates the supported attributes for the input object (for example, specific name for user-defined-function), and identifies that the input object of type table contains a specific name attribute.

System action:
No further operations can be performed as the specific name attribute is not supported for the Db2 object.

User response:
Remove the specific name attribute and try again.

IZPPL0274E **The DBA User is '*%1$s*' to execute SQL Queries under Logged-in User '*%2$s*' for reason: '*%3$s'***

Explanation:
The reason to execute the queries with DBA user instead of the log on user.

System action:
Queries will be executed by DBA user.

User response:
Execute the queries with DBA user instead of the log on user.

IZPPL0275W **User '*%1$s*' request role of '*%2$s*' when expanded PR viewing was disabled, using role '*%3$s*'**

Explanation:
The pull-request view permission for super admin and team users should be enabled. If it is disabled and the Get pull request API is invoked with role as ALL, then the role is reset to author or reviewer.

System action:
The role will be reset to author or reviewer, if the `enableExpandedPRViewing` is set as `false`.

User response:
Set `enableExpandedPRViewing` to `true` and try again.

IZPSC messages

Messages that are associated with security and authorities have the format `IZPSCnnnnx`.

IZPSC0002E **Fail to load dba configuration, reason *variable1*.**

Explanation:
Loading of the DBA configuration file failed with the reason provided.

System action:
The system cannot obtain DBA configuration to perform certain tasks.

User response:
Check if the DBA configuration file is corrupted. If not, contact IBM Software Support.

IZPSC0003E **Fail to create dba configuration, reason *variable1*.**

Explanation:
Creating the DBA configuration file failed with the reason provided.

System action:
The DBA configuration file is not created.

User response:
Contact IBM Software Support.

IZPSC0004E **Fail to connect to zss security authority. Please contact the administrator.**

Explanation:
UMS cannot connect to ZSS security authority.

System action:
UMS will not process the request.

User response:
Contact the administrator or IBM Software Support.

IZPSC0005E **dba configuration file not found.**

Explanation:
The DBA configuration file is not found.

System action:
UMS will not be able to perform certain tasks because no DBA configuration is found.

User response:
Contact IBM Software Support.

IZPSC0006E **The user id *variable1* in the request does not match user id *variable2* in the url**

Explanation:
The user ID in the URL and the user ID in the request content do not match.

System action:
UMS will not process the request.

User response:
Check if the request is formatted incorrectly.

IZPSC0007E **User id cannot be null, empty or only contain whitespace characters**

Explanation:
The user ID cannot be blank or null.

System action:
UMS will not process the request.

User response:
Check if the request is formatted incorrectly.

IZPSC0001W **Security cache was not initialized with stored policy data**

Explanation:
Serializing the policy data failed with the reason provided.

System action:
The server starts, but the security cache is not initialized.

User response:
Refresh the security cache. For details, refer to Setting up users and teams.

IZPSC0002W **Security cache of users and teams was not initialized. To initialize the security cache, click on the refresh button for Users/Teams (requires user with special privileges).**

Explanation:
The security cache was not initialized.

System action:
User interface (UI) will not show users.

User response:
Refresh the security cache. For details, refer to Setting up users and teams.

IZPSC0003W **User '%1$s' is not in the security cache**

Explanation:
A new user has been added to Unified Management Server (READ or UPDATE access to `IZP.SUPER` or `IZP.ADMIN`) since the last security cache refresh.

System action:
User interface (UI) will not show the new user.

User response:
Refresh the security cache to see the new user. For details, refer to Setting up users and teams.

IZPSC0013E **The data set '*%1$s*' could not be allocated. *%2$s(rc=%3$s, rsn=%4$s)*.**

Explanation:
There was an error when trying to allocate a data set.

System action:
The data set is not allocated.

User response:
Check that the data set does not have an exclusive lock on it.

IZPSC0014E **The data set '*%1$s*' could not be opened.**

Explanation:
There was an error when trying to open a data set.

System action:
The data set is not opened.

User response:
Check that the data set was properly created with the documented attributes.

IZPSC0016E Method is not supported with the current security model

Explanation:
The URL is not available with useSafOnly enabled.

System action:
The server will not perform the operation.

User response:
Use a different URL.

IZPSC0017E User '%1$s' is missing 'read' access to 'izp.function.users.get' in the 'izp' class

Explanation:
The user tried to refresh users but is missing the required profile access.

System action:
The server will not refresh users in the security cache.

User response:
Grant user READ access to `IZP.FUNCTION.USERS.GET` in IZP class.

IZPSC0018E User '%1$s' is missing 'read' access to 'izp.function.teams.get' in the 'izp' class

Explanation:
The user tried to refresh teams but is missing the required profile access.

System action:
The server will not refresh teams in the security cache.

User response:
Grant user READ access to `IZP.FUNCTION.TEAMS.GET` in IZP class.

IZPSC0019E User '%1$s' is missing 'read' access to 'izp.function.roles.get' in the 'izp' class

Explanation:
The user tried to refresh roles but is missing the required profile access.

System action:
The server will not refresh teams in the security cache.

User response:
Grant user READ access to `IZP.FUNCTION.ROLES.GET` in IZP class.

IZPSC0020E There were no users returned from zss so you cannot refresh users

Explanation:
Unified Management Server security ZSS plugin is not working correctly because of an internal error.

System action:
The server will not refresh users or teams in the security cache.

User response:
Contact IBM Software Support.

IZPSV messages

Messages that are associated with the Unified Management Server have the format IZPSVnnnnx.

IZPSV0001E Internal server error (*variable1*).

Explanation:
An internal error occurred in UMS with a detailed message.

System action:
UMS will not process the request or task.

User response:
See details of the error and contact IBM Software Support.

IZPSV0002E Internal server error: *variable1* (*variable2*).

Explanation:
An internal error occurred in UMS with a detailed message (*variable1*) and reason (*variable2*).

System action:
UMS will not process the request or task.

User response:
See details of the error and the reason. If necessary, contact IBM Software Support.

IZPSV0003E Internal server error: *variable1* reason *variable2* (*variable3*).

Explanation:
An internal error occurred in UMS with a detailed message (*variable1*), the reason (*variable2*), and the stack trace (*variable3*).

System action:
UMS will not process the request or task.

User response:
See details of the error and the reason. If necessary, contact IBM Software Support.

IZPSV0004E This endpoint is disabled. To enable it, contact IBM Software Support.

Explanation:

The endpoint is disabled because no data management product is installed.

System action:
UMS will not process the request or task.

User response:
Contact IBM Software Support.

IZPSV0005E **This endpoint is not yet enabled. It may be disabled for testing purposes. Please wait for it to be included in an official release.**

Explanation:
This endpoint is not yet enabled.

System action:
UMS will not process the request or task.

User response:
This endpoint might be disabled for testing purposes. Wait for it to be included in an official release. If the problem persists, contact IBM Software Support.

IZPSV0009E **IBM Unified Management Server is being shut down, reason '<reason text>'**

Explanation:
A problem occurred during UMS initialization, and UMS was unable to proceed. The reason text should help pinpoint the issue, which could be in an experience configuration, file permissions, etc.

System action:
UMS process terminates.

User response:
Evaluate reason text and take corrective action.

IZPTP messages

Messages that are issued by a third-party application while performing operations have the format `IZPTPnnnnx`.

IZPTP0001E ***'%1$s'* not configured.**

Explanation:
The host and port for the *{service}* are missing.

System action:
{service} is not configured.

User response:
Specify all the configuration details, restart the server, and then try again.

IZPTP0002E **Failed to establish connection.**

Explanation:
The host and port for the *{service}* are invalid.

System action:
Connection to the *{service}* was refused.

User response:
Specify valid configuration details, restart the server, and then try again.

IZPTP0003E **Failed to establish connection with *'%1$s'*.**

Explanation:
The host and port for the *{service}* are invalid.

System action:
Connection to the *{service}* was refused.

User response:
Specify valid configuration details, restart the server, and then try again.

IZPTP0004E ***'%1$s'* encountered an error. *'%2$s'*.**

Explanation:
The *{service}* encountered an error with the reason *{reason}*.

System action:
The *{service}* encountered an error and cannot proceed.

User response:
Proceed by referring to the information that is mentioned in *{error-from-service}*.

IZPTP0005E ***'%1$s'* encountered an error. *'%2$s'*. *'%3$s'*: *'%4$s'*.**

Explanation:
The *{service}* encountered an error.

System action:
The *{service}* encountered an error and cannot proceed.

User response:
Proceed by referring to the information that is mentioned in *{error-from-service}*.

IZPTP0006E **Correct privileges missing for this action. Try relaunching the Unified Experience console.**

Explanation:
Correct privileges missing for this action. Try relaunching the Unified Experience console.

System action:
The operation could not be completed because you do not have correct privileges for this operation.

User response:
Try to relaunch the Unified Experience console.

IZPTP0007E **SQL Tuning Services are unavailable. Contact your administrator to resolve this issue.**

Explanation

SQL Tuning Services are unavailable due to one of the following reasons:

- You do not have permission to perform the current task.
- SQL Tuning Services parameters are missing.
- SQL Tuning Services are down.

System action:
SQL Tuning Services could not be configured with the provided parameters.

User response

Contact your administrator to isolate the issue in the `zowe.logDirectory/izp-server<latest time stamp>.log` file. Identify one of the following error messages and resolve the issue before proceeding:

- Correct privileges missing for this action
- Tuning Service is not configured
- Tuning Service is unavailable

IZPUU messages

Messages that are issued by the Unified Management Server plug-in for Zowe have the format `IZPUUnnnnx`.

IZPUU0001E **Unable to access IBM Unified Management Server for z/OS**

Explanation:
The IBM Unified Experience for z/OS failed to communicate with the Unified Management Server and to retrieve necessary information that is required for the application to run.

System action:
The IBM Unified Experience for z/OS displays an error and does not function.

User response:
Check that the Unified Management Server is running and accessible, and then close and reopen the IBM Unified Experience for z/OS. If the problem persists, contact IBM Software Support.

IZPUU0002E **Unable to access current user roles from IBM Unified Management Server for z/OS**

Explanation:
The IBM Unified Experience for z/OS failed to retrieve details about the current user from the Unified Management Server.

System action:
The IBM Unified Experience for z/OS displays an error and does not function.

User response:
Check that the user has correct access as a user for the Unified Management Server, and then close and reopen the IBM Unified Experience for z/OS application. If the problem persists, contact IBM Software Support.

IZPUU0004E **Authorization failed, user not authenticated for the Unified Management Server.**

Explanation:
The logged in user does not have authorization to use the Unified Management Server 1.2 functionality.

System action:
The IBM Unified Experience for z/OS displays an error and does not function for this user.

User response

Add the user to Unified Management Server 1.2 via the `IZPUSRMD` job. Relaunch the Unified Management Server console.

Note: You can perform this action only when `useSafOnly` is set to `false`. If set to `true`, the EMS standard user management must be used. For details, refer to "Setting up users and teams" on page 46.

IZPUU0005E **The request is unable to communicate with the Zowe Auxiliary Address Space. Confirm the Zowe Auxiliary Address Space service is running and try again.**

Explanation:
The Zowe Auxiliary Address space started task was not reachable via Zowe plug-ins.

System action:
The Zowe plug-in returns an error because it needs the Zowe Auxiliary Address space to function.

User response:
Confirm that the Zowe Auxiliary Address space started task is running on the correct lpar and try again.

IZPUU0006E **The request timed out because there was no response from the Zowe Auxiliary Address Space. Try again after some time.**

Explanation:

The Zowe Auxiliary Address space only waits three seconds for Zowe plug-ins, resulting in failed requests.

System action:

The Zowe plug-in returns an error because it did not receive any data from the Zowe Auxiliary Address space.

User response:

Attempt a different operation before trying again. If the problem persists, try again later.

Chapter 11. References

This section documents IBM Unified Experience for z/OS reference information.

Configuring multifactor authentication for UMS

To improve security by using multifactor authentication, enable the Zowe Single Sign-On (SSO) provided by the Zowe API Mediation Layer. Zowe SSO leverages an access token to communicate with the z/OS services that are accessible through the Zowe API Mediation Layer. To enable Zowe SSO for Unified Management Server, configure the login user authentications from the UMS server and Zowe components to Db2 or IMS subsystems and subsystem tools services over TCP/IP to allow PassTicket-based authentication.

Before you begin

The UMS multifactor authentication requires IBM Z Multi-Factor Authentication and z/OSMF version 2.4 or later.

The Zowe API Mediation Layer must be enabled within the Zowe yaml file by enabling gateway by including the yaml definition `components.gateway.enabled: true`. For details, see the Zowe documentation for component configuration.

About this task

Before implementing UMS multifactor authentication, configure IBM Z Multi-Factor Authentication for new and existing users. To configure multifactor authentication for UMS, complete the following tasks.

Note: If you plan to use Zowe SSO and not MFA, steps 6 and 8 are not required.

Procedure

1. Shut down Zowe.
   a) Stop ZSS cross-memory server.

   ```
   /p <job_name>
   ```

   If Zowe defaults have been chosen, `job_name` should be replaced with ZWESISTC.

   b) Stop Zowe server.

   ```
   /p <job_name>
   ```

   If Zowe defaults have been chosen, `job_name` should be replaced with ZWESLSTC.
2. Apply the IBM APAR PH39582 required for z/OSMF.
3. All users must have access to z/OSMF.

   For RACF the user must be connected to either `IZUUSER` or `IZUADMIN` group. For other ESMs, refer to the z/OSMF documentation for your specific ESM.
4. To enable Zowe API Mediation Layer in UMS, configure the `authType` parameter.
   a) By using the `<system_admin_id>` user ID, open the `PARMLIB(ZWEYAML)` member.

   b) Edit the `components.izp.server.authType` parameter to `MFA_JWT`. The default value for the `authType` parameter is `STANDARD_JWT`.

   Note: `MFA_JWT` is only supported if API Mediation Layer Gateway is enabled. The `components.gateway.enabled` value must be true in the `zowe.yaml` file.

5. Configure Zowe for Single-Sign-On (SSO) and Multi-Factor Authentication (MFA). For details, see Zowe SSO overview, Zowe gateway component configuration, and Zowe App Server configuration for MFA.
6. Update RACF to use PassTickets for each MFA user by running the following command:

   ```
   ALU <ums_user_id> MFA(FACTOR(AZFPTKT1) ACTIVE NOTAGS)
   ```

   Where *<ums_user_id>* is a UMS user ID for which MFA login is configured. For details, see Using IBM MFA with PassTickets.
7. To RDEFINE resources in PTKTDATA and permit access to subsystem or subsystem tool services,

   For Db2 for evaluating PassTickets, refer to the following links:

   - Enabling Db2 to receive RACF PassTickets
   - Enabling Db2 to receive PassTickets for RACF protected user IDs

   For IMS Connect, refer to Enabling IMS Connect to receive RACF PassTickets.

   For IMS Tools, refer to "Enabling IMS Tools TCP server to receive RACF PassTickets" on page 258.
8. Permit access to the IBM Z Multi-Factor Authentication started task to verify PassTickets. Run the following command:

   ```
   PERMIT IRRPTAUTH.<applname>.* CLASS(PTKTDATA) –
   ID(<MFA STC user>) ACCESS (READ)
   ```

   For details, see Using IBM MFA with PassTickets.

   Notes:

   - To find the Db2 PassTicket <applname>, run the -DIS DDF Db2 command.
   - For a data sharing group, the Db2 PassTicket <applname> is the IPNAME or GENERICLU.
   - For a standalone Db2 subsystem, the Db2 PassTicket <applname> is the IPNAME or LUNAME.
   - For IMS Connect services <applname>, refer to Enabling IMS Connect to receive RACF PassTickets. For IMS Tools TCP server <applname>, refer to "Enabling IMS Tools TCP server to receive RACF PassTickets" on page 258.
9. The Zowe started task (<*Zowe_STC_User*>) needs access to the PTKTDATA class. Run the following command:

   ```
   PERMIT IRRPTAUTH.<applname>.* CLASS(PTKTDATA) -
   ID(<Zowe STC User>) ACCESS(UPDATE)
   ```

   Notes:

   - To find the Db2 PassTicket <applname>, see the Db2 command.
   - For IMS Connect services <applname>, refer to Enabling IMS Connect to receive RACF PassTickets. For IMS Tools TCP server <applname>, refer to "Enabling IMS Tools TCP server to receive RACF PassTickets" on page 258.
10. Refresh the profile by running the following command:

    ```
    SETROPTS RACLIST(PTKTDATA) REFRESH
    ```

11. Start Zowe.

    a) Go to System Display and Search Facility (SDSF) in 3270.

    b) Start ZSS cross-memory server.

    ```
    /S <job_name>,REUSAID=YES
    ```

    If Zowe defaults have been chosen, *<job_name>* should be replaced with ZWESISTC. You must specify REUSASID=YES after *<job_name>*.

    c) Start Zowe server.

```
/S ZWESLSTC,JOBNAME=ZWE1SV
```

If Zowe defaults have been chosen, *<job_name>* should be replaced with ZWESLSTC.

Zowe component lifecycle

Unified Management Server 1.2 starts as a component of Zowe. The understanding Zowe component lifecycle helps you troubleshoot issues related to Zowe and UMS.

Zowe component lifecycle

When Zowe launches, it validates, configures, and starts all of its components, including UMS. If there is a failure in the validation step of any component, Zowe will not start. If there is a failure in the configure step of any component, Zowe will proceed to start all the components, including the one whose configure step failed.

As of Zowe version 2.6, you can prevent Zowe from starting when there is a failure during the configure step of any component by setting `zowe.launchScript.onComponentConfigureFail` to "exit." The default is "warn," which will cause Zowe to print a warning message if a component configure step fails.

If there is a failure that causes a component to not start or to later halt unexpectedly, Zowe will attempt to restart the component a number of times, depending on how recently it failed. Zowe will not rerun the configure step for these restarts.

There are a number of situations in which UMS will not validate, configure, or start, depending on various error conditions. UMS will print messages when it fails to validate, configure, or start. It is recommended to set `zowe.launchScript.onComponentConfigureFail` to "exit" if you are running Zowe 2.6 or above.

YAML format

Unified Management Server 1.2 leverages YAML format for capturing configuration values. It is important to understand how YAML format works.

A sample YAML file format is shown below. The line numbers are included for reference and are not part of the YAML specification.

```
1  test:
2    a:
3      b:
4        c: First value
5    b: Second value
6    c:
7      a: Third value
8      b: Fourth value
9    d:
10   # This is a comment entry
11     a:
12       b:
13         c: Fifth value
14   e:
15     - This
16     - is
17     - an
18     - array
```

The first fragment in the dot sequence is the topmost key. The last fragment in the dot sequence is the value. Each of the fragments in between are subkeys that lead to the value. Although there are duplicate subkey names, they are treated as separate entities. Refer to the following table for identifying how a path to a value is determined in a YAML file.

Path to value	Value	Line number
test.a.b.c	First value	4

Path to value	Value	Line number
test.b	Second value	5
test.c.a	Third value	7
test.c.b	Fourth value	8
test.d.a.b.c	Fifth value	13
test.e	This is an array	14

Rules for editing a YAML file

Refer to the following rules for editing a YAML file:

- You must use spaces and not tabs.
- The file is case sensitive.
- If a line starts with a '#' symbol, it is considered as a comment and is ignored.
- The value of a key must follow a colon and single space.

Additional files in the `log/usermod` folder

The `log/usermod` folder contains several files that provide useful information about different activities, such as successful and failed commands, internal errors, and ignored users. You can use these files to debug issues with your entities, users, and commands.

Important: This section is not applicable when `useSafOnly` is set to `true`. The additional files in the `log/usermod` folder can be used only when `useSafOnly` is set to `false`.

`izpCommand` and `izpUserlist` are the most important files because these are the only two files that you can use for manual maintenance. When you use **`-DRYRUN`** to run the modify user script, these two files get copied into the *`components.izp.dataset.hlq.USERLIST`*`.COMMANDS` and *`components.izp.dataset.hlq.USERLIST`*`.PROPOSED` data sets so that you can easily access them.

The *`components.izp.dataset.hlq.USERLIST`*`.COMMANDS` data set contains a list of all the commands that should be run based on the parameters you passed, such as **`-CONNECT`**. The *`components.izp.dataset.hlq.USERLIST`*`.PROPOSED` contains the proposed state of the user list.

The following table describes the various files that you will find in the `log/usermod` folder:

Table 52. Additional files in `log/usermod` folder

File name	Description
`izpUsers`	A list of users that will be or were modified. If you use the **`-DRYRUN`** option, `izpUsers` lists the users that will be modified. Without the **`-DRYRUN`** option, `izpUsers` lists the users that were modified as a result of the various commands that were run.
`izpCommands`	A list of all commands. This file is populated only when you use **`-DRYRUN`** to run the modify user script. If you run the script without **`-DRYRUN`**, the files `izpCommandsSucceeded` and `izpCommandsFailed` are populated instead of `izpCommands`.
`izpCommandsSucceeded`	A list of commands that ran successfully. This file is populated when you run the modify user script without **`-DRYRUN`**.

Table 52. Additional files in log/usermod folder (continued)

File name	Description
izpCommandsFailed	A list of commands that failed. This file is populated when you run the modify user script without **-DRYRUN**.
izpUnprocessedEntities	A list of users and groups that could not be processed. Most probably, either the user and the groups did not exist or you did not have access to view them.
izpIgnoredUsers	A list of users who were not eligible to perform the operation for which they were being used. For example, if a user that already existed in a group was being added to it, that user would be ignored and listed in the izpIgnoredUsers file.
izpUserlist.tmp	The last state of the user list.
izpUserlist	The current state or proposed state of the user list depending on whether or not you used **-DRYRUN**.
izpErrors	The internal errors that occurred within the script and caused it to end unexpectedly.

To see the contents of these files on your terminal, run IZPUSRMD as a shell script:

1. Navigate to the location of the script: components.izp.runtimeDirectory/ums/opt/bin.
2. Run the command in the following format:

   ```
   ./izp-modify-users.sh <-DRYRUN | > <-CONNECT | > <ADD | REMOVE> <SUPER | ADMIN>
   <USERS_AND_GROUPS>
   ```

   For example:

   ```
   ./izp-modify-users.sh add super usera groupa userb groupb
   ```

After the script completes running, if the files listed in the Table 52 on page 238 contain data, it is displayed on the terminal.

WLM recommendations for Zowe, UMS started tasks, and jobs

For the Zowe started task, the major work is running in z/OS UNIX System Services child processes and hence it is classified as an OMVS workload in the Workload Manager (WLM). The work in Zowe task can be compared to users working in TSO, so the recommendation is to use a high-performance, one-period velocity goal. To classify the workload in WLM, note the address space name of the main STC and z/OS UNIX System Services child processes depends on your values for ZOWE_PREFIX and ZOWE_INSTANCE. For more information, see Zowe environment variables. Refer to the Zowe manual for updated recommendations.

For Unified Management Server, the major work is also running a z/OS UNIX System Services child process. The Unified Management Server serves API requests from the Zowe virtual desktop or other REST API consumers like pipelines, so the recommendation is to use a high-performance, one-period velocity goal. In addition, Unified Management Server submits batch jobs to reform certain operations. Some of these operations are synchronous and the UI is blocked until the job is completed (for example, apply DDL to an instance), whereas other operations are asynchronous without blocking the UI (for example, merge a pull request, provision an instance, create application or deploy DDL to higher environments).

Again, the recommendation is to use a high-performance, one period velocity goal for those jobs. For improved system performance you can consider a two-period goal allowing short-running jobs that blocks the UI to complete with high performance but give lower priority and performance to longer running jobs. However, do note that setting a too low priority and performance goal can result in a batch job taking an extended time to complete. If this job is implementing a disruptive DDL change this might unnecessarily extend the period where affected Db2 objects are unavailable.

To classify the workload in Workload Manager (WLM), note the default name of the STC is `IZPSRV`. The z/OS UNIX System Services child processes will use default names, such as `IZPSRV` where n is an integer 1-9. The batch jobs submitted by Unified Management Server have a default job name prefix of IZP, but it can be reconfigured at the global level using the `PARMLIB` member `ZWEYAML` element `components.izp.server.jobPrefix` or at the environment level within the UI.

Complex Db2 site rule variables

When you create complex site rules, you can use the following variables in the Python expressions to represent object attributes. For example, when writing an expression that applies to database objects you use the variable `encoding_scheme` to represent the encoding scheme of that database. The table below describes the objects and attribute variables. You can also do a GET call to the following API address `<server address>:<port>/ws/ddl/site-rules/allowed-rules/` and it will return the same list.

Table 53. Object attribute variables

Object	**Attributes**
alias	name, schema
column	name, coltype, tbname
database	encoding_scheme, bpool, stgroup, name, indexbp
index	name, creator, type, tbname, storname, vcatname, bpool, partitioned, padded, gbpcache, pctfree, freepage, closerrule, copy, compress, clustering, priqty, secqty, erase, uniquerule, defer, define
procedure	name, creator, language, asutime
sequence	name, schema, datatype, start, increment, cycle, order, cache, maxvalue, minvalue
stogroup	name, volumes, vcatname, dataclass, mgmtclas, storclas
synonym	name, creator
table	auditing, encoding_scheme, column, tsname, valproc, append, datacapture, name, edproc, clustertype, dbname, creator, partitioning, primary_key, foreign_key, volatile, pagenum, trackmod, member_cluster, dssize, log, organizationtype
tablespace	encoding_scheme, priqty, secqty, erase, gbpcache, maxrows, partitions, maxpartitions, freepage, pctfree, pctfree_update, dbname, log, bpool, storname, closerule, name, trackmod, dssize, segsize, locksize, lockmax, compress, define, insertalg, tstype **Note:** Before creating the UTS site rule by using the **`tstype`** attribute, ensure that the following conditions are satisfied: • **`SEGSIZE`** for the tablespace object should be specified and should have a value that is greater than 0. • Either **`MAXPARTITION`** or **`NUMPARTS`** should be specified and should have a value that is greater than 0.
trigger	name, schema, granularity, sqlpl, instead_of, secure

Table 53. Object attribute variables (continued)

Object	Attributes
userDefinedFunction	name, schema, external
view	name, creator

Db2 DDL file name definitions and validations

The DDL file of each object in an application contains the details of that object. User friendly file names help locate relevant files, especially if an application contains multiple objects of the same type.

To define the template for the file name of an object, you can use the following format:

```
DDL_FILE_NAME_PATTERN:
  NICE_NAME: true
  TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  TS: <SSID(4)>_<QUALIFIER(8)>_<NAME(8)>.ddl
  DB: <SSID(4)>_<NAME(8)>.ddl
  AL: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  VW: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  SP: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  SP_SQL: <SSID(4)>_<QUALIFIER(80)>_<NAME(80)>_<VERSION(73)>.spsql
  SP_JAVA: <SSID(4)>_<QUALIFIER(102)>_<NAME(128)>.javaspsql
  UDT: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  UDF: <SSID(4)>_<QUALIFIER(80)>_<NAME(80)>_<SPECIFIC_NAME(72)>.udfsql
  SEQ: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  SY: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  TG: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  IX: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  SA: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
  GV: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl
```

When you define the templates by using this format, the file is saved in the folder that is named according to the object type. For example, if you define the template for table objects by using this format, the file is created in the TB folder and the name of the file is *SSID_QUALIFIER_NAME.ddl*.

When you specify the DDL definition by using the new template, new files will be created. The existing DDL definition files will not be renamed and will be available in their original location. If you want to name the DDL files of an existing application according to the updated templates, you must delete the application and recreate it.

Note: When you define an object that was being used in an application that is now deleted, a unique identifier string will be added to the DDL file name of this object. This string ensures that the file name remains unique.

There are multiple ways to define the attributes in the templates. While some attributes are required, others are optional. You can define the attributes as needed.

Defining templates

You can define a template by using some or all these attributes: `SSID`, `QUALIFIER`, `NAME`, `VERSION`, and `SPECIFIC_NAME` attributes. Ensure the templates meet the following requirements:

- Every template must have a mandatory NAME attribute.
- Attribute is defined in the format `<attribute(maxLength)>`.
  - An attribute is specified only once in the template. For example, `<SSID(4)>_<NAME(50)>_<NAME(40)>.ddl` is an invalid template because NAME is repeated.
  - Correct attributes are specified in each template. For example, `VERSION` can be specified only for the SP_SQL object.
- The template must have an extension. You can specify any extension. For example: `TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(8)>.ddl` and `TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(8)>.table` are both valid templates.

- File and folder restriction:
  - The name of the folder must be less than or equal to 255 characters.
  - The name of the file must be less than or equal to 246 characters. Nine characters in the file name are reserved to add a unique identifier string. This string is not a mandatory part of the file name and is added only when it is necessary to make the file name unique.

  If the name of the file or the folder exceeds this limit, the following error message is logged: `DDL file name pattern` *`template_name`* `is too long based on specified` *`maxLengths`*. You must resolve the errors and then restart the UMS server.

In addition to the mandatory requirements, there are multiple ways to define the templates.

Table 54. Template definitions

Template definitions	**Example**
Define the files according to the template definitions that are specified in the `IZPD2DPM PARMLIB` member without specifying the template for each Db2 object type.	Specify `NICE_NAME: true` **Template definition** Not needed The templates will be defined according to the template definitions that are specified in the `IZPD2DPM PARMLIB` member.
Define the template for only one Db2 object type.	Specify `NICE_NAME: true` **Template definition** `TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl` The files for only the table objects will be named according to this template. The template definitions that are specified in the `IZPD2DPM PARMLIB` member will be used for other object types.

Rearrange the attributes in the template definition.	**Template definition**
	`TB: <QUALIFIER(108)>_<SSID(4)>_<NAME(128)>.ddl`
	Template definition with rearranged attributes
	`TB: <NAME(128)>_<QUALIFIER(108)>_<SSID(4)>.ddl`

Change the definition of only one attribute in the template definition.	**Template definition**
	`TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl`
	New template definition
	`TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(`**`120`**`)>.ddl`

Table 54. Template definitions (continued)

Template definitions	Example

Create a folder structure for the object definition file.	While defining the template, replace the underscore (_) with a forward slash (/).
	Template definition
	`TB: <SSID(4)>_<QUALIFIER(108)>_<NAME(128)>.ddl`
	New template definition
	`TB: <SSID(4)>/<QUALIFIER(108)>/<NAME(25)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `NAME = TG1023`
	New file name and folder structure
	A folder named SMEMP341 is created inside a folder named ZX1C, and the file `TG1023.ddl` is created inside SMEMP341.
	Note: The file for this object, which was created before you specified this new template definition, will be available in its original location.

Trim the file name.	**Template definition**
	`<SSID(4)>_<QUALIFIER(108)>_<NAME(5)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `NAME = VERYLONGTABLENAME`
	File name
	`ZX1C_SMEMP341_VERYL.ddl`

Include a specific string in the template definition.	**Template definition for creating a file name**
	`TB: <SSID(4)>_<QUALIFIER(108)>_Table_<NAME(25)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `NAME = TG1023`
	File name
	`ZX1C_SMEMP341_Table_TG1023.ddl`
	Template definition for creating a folder structure
	`TB: <SSID(4)>/<QUALIFIER(108)>/Table/<NAME(25)>.ddl`
	Folder structure
	A folder named `Table` is created and the file named `TG1023.ddl` is placed inside the `Table` folder.

Table 54. Template definitions (continued)

Template definitions	Example

Attribute values are same for several objects.	**Template definition**
	`<SSID(4)>_<QUALIFIER(108)>_<NAME(4)>.ddl`
	Attribute values for the object
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `TABLE1`
	Attribute values for another object
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `TABLE2`
	File name
	`ZX1C_SMEMP341_TABL.ddl` for the first object and `ZX1C_SMEMP341_TABL_`*`UNIQUE123`*`.ddl` for the next object.
	Nine random characters (one underscore and 8 characters) are suffixed to the file name to make the file name unique. These characters are added only when duplicate file names might be created.

The attribute value contains trailing spaces.	**Template definition**
	`<SSID(4)>_<QUALIFIER(108)>_<NAME(25)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `TG1023<extra spaces>`
	File name
	`ZX1C_SMEMP341_TG1023.ddl`
	The trailing spaces in the attribute values are removed while creating the file name.

The attribute value contains leading spaces.	**Template definition**
	`<SSID(4)>_<QUALIFIER(108)>_<NAME(25)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `<extra spaces>TG1023`
	File name
	`ZX1C_SMEMP341_<extra spaces>TG1023.ddl`
	The spaces at the beginning of the attribute values are included while creating the file name.

Table 54. Template definitions (continued)

Template definitions	Example

The attribute value contains a ligature.	**Template definition**
	`<SSID(4)>_<QUALIFIER(108)>_<NAME(25)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `NAME = NæME`
	File name
	`ZX1C_SMEMP341_NME.ddl`
	The ligatures in the attribute value are ignored while creating the file name.

The attribute value contains an accented character.	**Template definition**
	`<SSID(4)>_<QUALIFIER(108)>_<NAME(25)>.ddl`
	Attribute values
	`SSID = ZX1C`, `QUALIFIER = SMEMP341`, and `NAME = NáME`
	File name
	`ZX1C_SMEMP341_NaME.ddl`
	The accented characters in the attribute value are normalized while creating the file name.

Db2 operations used by DevOps experiences and Administration Foundation

DevOps experiences and Administration Foundation can perform various Db2 operations by using the DBA user ID defined in configuration YAML files.

Table 55. Db2 operations used by DevOps experiences and Administration Foundation

Primary authentication ID	DevOps experiences or Administration Foundation	Function
DBA user ID	DevOps experiences or Administration Foundation	Db2 subsystem discovery
	DevOps experiences or Administration Foundation	Register, update, and delete a Db2 subsystem
	DevOps experiences or Administration Foundation	Load graphdb from Db2 catalog tables
	DevOps experiences	Add or remove users from a team that owns an instance
	DevOps experiences	Create, update, delete, and dry-run an application
	DevOps experiences	Perform the following actions in an instance: • Provision or deprovision • Delete • Pull objects from upstream • Restore objects to last commit state • Edit definition of object • Add objects to instance • Apply changes to instance
	DevOps experiences	Merge, approve, decline, resume, revert, and synchronize a pull request
	DevOps experiences	Create, delete, and restore a snapshot

Note: The operations that are not listed in the above table are run by the authenticated user.

IMS commands used by IMS Administration Foundation

IMS Administration Foundation uses some IMS commands in its backend processes to perform various actions. IMS Administration Foundation uses type-2 commands wherever possible.
Type-1 commands, such as **`/DISPLAY ACT`** are used in IMS page of the IMS components. All IMS commands, including the type-1 commands, are issued through the IMS CSL Operations Manager; thus the CSL SCI security and the CSL OM command security apply if they are activated. In addition to the IMS commands, IMS DBRC commands are issued by using the UMS DBA user ID for some backend system processes.

IMS commands issued from the Zowe auxiliary address space

The following commands can be issued by the started task ID of the Zowe auxiliary address space (ZWESASTC).

Table 56. IMS commands issued from the Zowe auxiliary address space

Command verb	Resource type	Keyword parameters
QUERY	IMS	TYPE(ALL) SHOW(ALL)
	IMSCON	TYPE(CONFIG) SHOW(ALL)
		TYPE(PORT) SHOW(ALL)
	IMSPLEX	SHOW(ALL)

IMS commands issued from the DBA user ID

The following IMS commands can be issued by the DBA user ID for IMS Administration Foundation functions. The DBA user ID is specified by the value for the key `components.izp.security.pkcs11.dbaUser` in the ZWEYAML configuration member of the UMS PARMLIB.

Table 57. IMS type-2 commands issued from the DBA user ID for IMS Administration Foundation

Command verb	Resource type	Keyword parameters
QUERY	DB	SHOW(DEFNTYPE)
	PGM	SHOW(DEFNTYPE)
	TRAN	SHOW(DEFNTYPE)
	RTC	SHOW(DEFNTYPE)
	OLC	LIBRARY(OLCSTAT) SHOW(ACTVLIB)
	IMSCON	TYPE(ODBM) SHOW(ALL)
	ODBM	TYPE(DATASTORE) NAME(*) SHOW(ALIAS)
EXPORT	DEFN	TARGET(RDDS)

Table 58. IMS type-1 command issued from the DBA user ID for IMS Administration Foundation

Command verb	Keyword parameters
DISPLAY	MODIFY

IMS DBRC commands issued from the DBA user ID

IMS DBRC commands are used to get information on IMS database objects for registered IMS data sharing groups. The commands are issued by an IMS object discovery job. The object discovery jobs run at the timing of the registration. The jobs run periodically after the registration of an IMS data sharing group. If you are using the DBRC command security for the data sharing group to be registered, you need to allow the UMS DBA user ID to use the IMS DBRC command security resources that are listed in the Table 59 on page 248.

Note: The DBA user ID is specified by the value for the key `components.izp.security.pkcs11.dbaUser` in the ZWEYAML configuration member of the UMS PARMLIB.

Table 59. DBRC command security resources involved with commands issued by UMS DBA user ID

Command verb	Modifier	Qualifier	Description	RACF resource profile example
LIST	DB	*dbname*	The qualifier *dbname* can be either of the following: • The DBD name of a non-partitioned full-function database, a HALDB master database, or a DEDB. • The partition name of a HALDB database.	*safhlq*.LIST.DB.*dbname*
	DB	ALL	This command is used to get the following information of a database: • Database data set names for a non-partitioned full-function databases or HALDB partitions. • Area names for DEDB databases.	*safhlq*.LIST.DB.ALL
	DBDSGRP	ALL	This command is used to get the following information: • DBDS groups (DBDSGRPs) and their members • DB groups (DBGRPs) and their members • Recovery groups (RECOVGRPs) and their members	*safhlq*.LIST.DBDSGRP.ALL
	CAGRP	ALL	This command is used to get the following information: • CAGRPs and their members • Properties for each CAGRP	*safhlq*.LIST.CAGRP.ALL
STDBRC	Not applicable	Not specified	The resource profile name *safhlq*.STDBRC is used to control access to the DBRC API. This implies that no specific IMS subsystem will be registered with DBRC when the DBRC commands listed above are issued from the UMS DBA user ID.	*safhlq*.STDBRC

Note: UMS login users can issue DBRC commands from IMS command processor. If you are using the DBRC command security, the DBRC command security setting applies to the DBRC commands as well as the OM command security setting for the DBRC commands that are issued by UMS login users.

For details of the DBRC security, see the *IMS System Administration* guide. A list of command verbs, modifiers, and qualifiers to be used for DBRC command authorization support can be found in the topic "Resource names for command authorization". Examples of resources profiles can be found in the topic "DBRC API security features" in the *IMS System Programming APIs* guide.

IMS commands issued with the login user credentials

For using IMS Administration Foundation functions, the following IMS commands can be issued with the user's login credential. Zowe ZSS server authenticates the user, but neither the ZSS server nor the Unified Management Server is involved in resource permission for any IMS resource. IMS command permissions are checked only by the IMS Operations Manager and individual IMS subsystems.

Table 60. IMS commands issued with the login user credentials

Command verb	Command keyword	Command
DIS	OTMA	DISPLAY OTMA
	ACT	DISPLAY ACT REGION
	STRUC	DISPLAY STRUC ALL
QRY	DB	QUERY DB NAME(*name1*,*name2*,...) SHOW(ALL,LOCAL)
		QUERY DB NAME(*name*) SHOW(ALL,LOCAL)
		QUERY DB NAME(*name*) SHOW(DEFN,GLOBAL,IMSID)
	IMSPLEX	QUERY IMSPLEX NAME(*imsplex*) TYPE(*type1*,*type2*,...) SHOW(ALL)
		QUERY IMSPLEX NAME(*) SHOW(ALL)
	MEMBER	QUERY MEMBER TYPE(IMS) SHOW(ATTRIB)
	PGM	QUERY PGM NAME(*name1*,*name2*,...) SHOW(ALL,LOCAL)
		QUERY PGM NAME(*name*) SHOW(ALL,LOCAL)
		QUERY PGM NAME(*name*) SHOW(DEFN,GLOBAL,IMSID)
	RTC	QUERY RTC NAME(*name1*,*name2*,...) SHOW(ALL,LOCAL)
		QUERY RTC NAME(*name*) SHOW(ALL,LOCAL)
		QUERY RTC NAME(*name*) SHOW(DEFN,GLOBAL,IMSID)
	TRAN	QUERY TRAN NAME(*name1*,*name2*,...) SHOW(ALL,LOCAL)
		QUERY TRAN NAME(*name*) SHOW(ALL,LOCAL)
		QUERY TRAN NAME(*name*) SHOW(DEFN,GLOBAL,IMSID)
RML		RMLIST DBRC='RECON STATUS'

IMS Administration Foundation and IMS Tools

If you want to use the following features in IMS Administration Foundation in UMS, you must install some additional IMS Tools products.

The following table lists IMS Administration Foundation features and their prerequisite IMS Tools products.

Table 61. IMS Administration Foundation and IMS Tools

IMS Administration Foundation feature	IMS Tools products
Registering IMS subsystems and data sharing groups for IBM IMS Administration Tool for z/OS (also referred to as IMS Administration Tool)	You must install and configure the following IMS Tools product: - IBM IMS Administration Tool for z/OS 1.1 (Program Number: 5655-CAT) The PTF UI93612 provided by the APAR PH55647 must be applied to the IMS Administration Tool, and the UMS maintenance level must be 1.2.0.4 or later. To use the feature, the following requirements must be satisfied: - IMS Tools Base Distributed Access Infrastructure (DAI) servers must be configured with at least one TAS server and an IMS Tools Knowledge Base server. For details of configuration requirements, see Configuring IMS Tools Base servers. - IMS Administration Tool that is installed must be registered to the IMS Tools Knowledge Base. For details, see Registering IMS Tools product components. - For security setup requirements for IMS Administration Tool, see Security setup for IMS Administration Tool.
Viewing IMS Tools utility reports for databases, HALDB partitions, and DEDB areas	You must configure IMS Tools Knowledge Base (IMS Tools KB) and at least one IMS Tools KB server instance must be up and running. Only IMS Tools products can store their utility reports in the IMS Tools KB repository. To collect database usage statistics for non-partitioned full-function databases and HALDB partitions, you must install and configure one of the following IMS Tools Solution Pack products: - IBM IMS Database Solution Pack for z/OS 2.2 or later (Program Number: 5655-DSP) - IBM IMS Database Utility Solution for z/OS 2.1 or later (Program Number: 5698-DUL)
Viewing the database space usage statistics (database sensor data)	- You need at least one IMS Tools KB server instance and must initialize the Sensor Data repository of IMS Tools KB. - To collect database usage statistics for DEDB areas, you must install and configure IBM IMS Fast Path Solution Pack for z/OS 2.1 or later (Program Number: 5698-FPP).

Table 61. IMS Administration Foundation and IMS Tools (continued)

IMS Administration Foundation feature	IMS Tools products
Viewing the database space usage exceptions	• You need at least one IMS Tools KB server instance and must initialize the Sensor Data repository of IMS Tools KB. • You need to make the previous feature available, and you also must configure the Policy Services component of IMS Tools Base. The policies and rules for the REORG domain must be installed and configured. The Policy Services policies and rules must be installed into the repository that is managed by the IMS Tools KB instance configured for the previous feature. • You must configure at least one AD server instance that is connected to the IMS Tools KB server instance that was configured for Viewing the database space usage statistics (database sensor data) feature.
Viewing the database exceptions related with database backup, change accumulation, and recovery	You need at least one IMS Tools KB server instance and must initialize the Sensor Data repository of IMS Tools KB. • You must install and configure IBM IMS Recovery Solution Pack for z/OS 2.1 or later (Program Number: 5655-ISR). • You must configure the Policy Services component of IMS Tools Base. The policies and rules for the RECOVERY domain must be installed and configured. • You must configure at least one AD server instance that is connected to the IMS Tools KB server instance that was configured for the following features: Viewing the database space usage statistics (database sensor data) and Viewing the database space usage exceptions.

Table 61. IMS Administration Foundation and IMS Tools (continued)

IMS Administration Foundation feature	IMS Tools products
Visually displaying database segment tree structure defined by a DBD or by a DB PCB in a PSB (DBD and PSB Map feature)	You must install and configure the following IMS Tools product: • IBM IMS Library Integrity Utilities for z/OS 2.2 (Program Number: 5655-U08) The PTF `UI81472` provided by the APAR PH47086 must be applied. If the PTF `UI92487` for APAR PH54565 for IMS Library Integrity Utilities (LIU) is not applied, then this feature requires DBDLIBs and PSBLIBs to be registered when registering an IMS data sharing group, even if IMS catalog was used. If you want to use the DBD and PSB Map feature, create DBDLIB and PSBLIB from the IMS catalog and specify those libraries in the RECON ID definition for the data sharing group for which the subject IMS catalog is used. After the IMS LIU maintenance has been applied, you do not need to specify a DBDLIB or PSBLIB when you register an IMS data sharing group that uses IMS catalog to use the DBD and PSB Map feature. The UMS maintenance level must be 1.2.0.2 or later. To use the improved IMS catalog support, the following requirements must be satisfied: • At least one TAS server must be started on each z/OS system, in the sysplex, on which a data sharing member IMS can be started. Those TAS servers must share the same XCF group name that is specified by the TAS server configuration parameter `XcfGroupName`. For details on the parameters, see TAS configuration parameters section in *IBM IMS Tools Base for z/OS Configuration Guide*. • The UMS login users who want to use the DBD and PSB Map feature must be authorized to run IMS BMP jobs with the IMS catalog PSB DFSCP000 for all target IMS data sharing groups.

To use any of these features for an IMS data sharing group that you register, you need to create a RECON ID for the data sharing group before you register the data sharing group. For details of creating a RECON ID, see Adding a new RECON environment in *IBM IMS Tools Base for z/OS Configuration Guide*.

Configuring IMS Tools Base servers

You must configure some IBM IMS Tools Base servers to enable the IMS Administration Foundation features for IMS data sharing groups. At least one set of the following servers needs to be set up if you plan to use any feature listed in Table 61 on page 250.

- Distributed Access Infrastructure (DAI) servers
  - DAI must be configured with at least one TAS server
- IMS Tools Knowledge Base (IMS Tools KB) server
- Autonomics Director (AD) server

  This sever is required if you plan to use either or both of the following features:

  - Viewing the database space usage exceptions.

- Viewing the database exceptions related with database backup, change accumulation, and recovery.

For details on the configuration of DAI servers, see Configuring Distributed Access Infrastructure of the *IBM IMS Tools Base for z/OS Configuration Guide*.

For details on the configuration of IMS Tools Knowledge Base, see Configuring IMS Tools Knowledge Base and RECON IDs.

For details on the configuration of IMS Administration Foundation to use DAI TCP server, IMS Tools KB server, and optional AD server, see Installing IMS Administration Foundation.

To use some of the IMS Tools services from IMS Administration Foundation, you must register the related products by using the 'IMS Tools product registration' process. For details, see Registering IMS Tools product components.

Configuring IMS Tools Knowledge Base and RECON IDs

As a part of the configuration for IMS Tools Knowledge Base, you are guided to create RECON IDs, which are sometimes called *locales* in IMS Tools Base documentation. A RECON ID can be created on an IMS Tools Knowledge Base server instance for a set of RECON data sets that are used for an IMS data sharing group.

Note: Each data sharing group needs to be associated with a RECON ID when the group is registered to the Unified Management Server. You must create a RECON ID for the RECON data sets that are used by the data sharing group before you register the group.

If IMS Administration Tools is installed with the maintenance provided by the APAR PH55647 and the IMS Administration Tool is registered to IMS Tools Knowledge Base, you can create a RECON ID on the IMS Tools Knowledge Base server for an IMS data sharing group when registering the data sharing group by using the IMS subsystem registration feature of the Unified Management Server. For details on the IMS Tools registration, see Registering IMS Tools product components. For details on the IMS subsystem registration, see Registering IMS subsystems as a data sharing group.

Important: The RECON ID content requirement for displaying database segment tree structure defined by a DBD or DB PCB varies depending on the IMS configuration and the maintenance level of UMS and IMS LIU.

- If the IMS catalog is not used for a data sharing group that is associated with a RECON ID, you must specify DBD and PSB libraries when you create the RECON ID.
- If you are using IMS-managed ACBs for an IMS data sharing group, you must specify the IMSCATHLQ field in the RECON ID definition.
- For the IMS data sharing groups for which the IMS catalog is used, the following requirements apply:
  - If the UMS maintenance level is less than 1.2.0.2, you must specify DBD and PSB libraries when you create the RECON ID even when the IMS-manage ACBs feature is used.
  - If the UMS maintenance level is 1.2.0.2 or later, the following requirements apply:
    - If the PTF UI92487 provided by the APAR PH54565 has not been applied to IMS LIU, you must specify DBD and PSB libraries in the RECON ID definition for the data sharing group.
    - If the PTF UI92487 provided by the APAR PH54565 has been applied to IMS LIU, the DBD and PSB libraries specified in the RECON ID will not be used for data sharing groups that uses IMS catalog independent of whether the IMS-managed ACBs are used or not. For each of those data sharing groups, its IMS catalog database is always used as the source of DBD and PSB definitions.

Registering IMS Tools product components

If you want to use one of the features listed in the Table 61 on page 250, you must register associated IMS Tools product or products to the IMS Tools Knowledge Base. You must register the following IMS Tools Base 1.7 components as base registration requirements:

- Distributed Access Infrastructure

- FPQ Repository Server
- IMS Tools Knowledge Base Services

For additional registration requirements, see the Table 62 on page 254.

If you use the IMS Tools Setup function for post-SMP/E-installation customization of IBM IMS Tools Base, product registration can be performed automatically. IMS Tools Setup generates and runs the required IMS Tools product registration JCL. For details, see Registering products in the *IBM IMS Tools Base for z/OS Configuration Guide*.

Important: The UMS DBA user ID must have READ access for the load module libraries of all IMS Tools that are registered to the IMS Tools Knowledge Base.

Table 62. IMS Administration Foundation features and IMS Tools registrations

IMS Administration Foundation feature	IMS Tools components to be registered
Registering IMS subsystems and data sharing groups for IMS Administration Tool	• IMS Tools Base 1.7 – Generic Exits and Tools Online System Interface • IMS Administration Tool 1.1
Viewing IMS Tools utility reports for databases, HALDB partitions, and DEDB areas	One or more components of the IMS Tools product that support the report function of the IMS Tools Knowledge Base
Viewing the database space usage statistics (database sensor data)	• IMS Tools Base 1.7 - Policy Services • IMS Tools Base 1.7 - IMS Generic Exits • IMS Tools Base 1.7 - Tools Online System Interface • For non-partitioned full-function databases and HALDBs, register one of the following products that you have installed: – IMS Database Solution Pack - Full-Function Database Sensor – IMS Database Utility Solution - Full-Function Database Sensor • For DEDB, register the following product: – IMS Fast Path Solution Pack
Viewing the database space usage exceptions	• IMS Tools Base 1.7 – Policy Services • IMS Tools Base 1.7 – Autonomics Director • For non-partitioned full-function databases and HALDBs, register one of the following products that you have installed: – IMS Database Solution Pack - Full-Function Database Sensor – IMS Database Utility Solution - Full-Function Database Sensor • For DEDB, register the following product: – IMS Fast Path Solution Pack
Viewing the database exceptions related to database backup, change accumulation, and recovery	• IMS Tools Base 1.7 – Policy Services • IMS Tools Base 1.7 – Autonomics Director • IMS Recovery Solution Pack - Database Recovery Facility, Database Recovery Facility Extended Function, and Recovery Sensor components

Table 62. IMS Administration Foundation features and IMS Tools registrations (continued)

IMS Administration Foundation feature	IMS Tools components to be registered
Visually displaying database segment tree structure defined by DBD or DB PCB in a PSB (DBD and PSB Map feature)	IMS Library Integrity Utilities 2.2

Note: For information on the JCLs used for registering each IMS Tools product component, see Reference: Product registration JCL for IMS Tools.

Policy based database monitoring with IMS Tools

If you have installed one of the following IMS Tools products and you want to use the feature of viewing database space usage exceptions, you must install REORG policies and rules of IMS Tools Base Policy Services:

- IBM IMS Database Solution Pack for z/OS
- IBM IMS Database Utility Solution for z/OS
- IBM IMS Fast Path Solution Pack for z/OS

If you have installed IBM IMS Recovery Solution Pack for z/OS and you want to use the feature of viewing database exceptions that are related with database backup and recovery readiness, you must install RECOVERY policies and rules of IMS Tools Base Policy Services.

For details of installing Policy Services policies and rules into IMS Tools Knowledge Base, see Configuring Policy Services in the *IBM IMS Tools Base for z/OS Configuration Guide*. You can use the installed policies as they are or by configuring threshold values for rules selected for each policy. You can also create your own policy. For details of configuration of policies and rules thresholds, see Using Policy Services in the IBM IMS Tools Base for z/OS Policy Services User's Guide and Reference.

Databases, HALDB partitions, and DEDB areas that you want to use for the exception viewing feature must be registered to an Autonomics Director server as a Monitor List entry. The policies you installed, configured, or created can be specified when you register a Monitor List entry. If you have a large number of such databases, HALDB partitions, or DEDB areas, it is recommended to use the batch Monitor List Registration utility. For details of this utility, see Monitor List Registration utility (IAVBUTL0) in the IBM IMS Tools Base for z/OS Autonomics Directory User's Guide and Reference.

The registered Monitor List entries are used to evaluate database statistics, database backup frequency, or database recovery readiness. You must run some IMS Tools utilities with Database Sensor capability enabled or Stand-alone Database Sensors. For details of setting up JCLs for those utilities or sensor jobs, see the IBM IMS Solution Packs Data Sensor User's Guide. Those statistics and collected RECON information are sometimes called as `sensor data` in IMS Tools documentation. The collected sensor data can be retrieved by IMS Administration Foundation's statistics API and can be viewed on the Unified Experience UI.

If a Monitor List entry is defined for a database, a HALDB partition, or a DEDB area, its sensor data are used to evaluate against the rules selected by the policy that was chosen for a database, a HALDB partition, or a DEDB area when you define a Monitor List entry for the Autonomics Director server. Autonomics Director generates (rule-defined threshold) exceptions and associated recommended actions and this information is stored in the Monitor List entry for the subject database, HALDB partition, or DEDB area. This information will be displayed in the Exceptions tab of the DBD page of IMS Administration Foundation's Unified Experience user interface.

For better understanding of the entire scenario, see "Scenario: Policy-based database monitoring and tuning" in IBM IMS Solution Packs IMS Database Space Tuning Guide. This guide provides information on how to interpret reported exceptions and what actions are possible or recommended.

Setting up IMS Tools Base server security

For details of the security setup for IMS Tools Base servers and components, see Summary of security-related settings in the *IBM IMS Tools Base for z/OS Configuration Guide*.

Managing IMS Tools Base servers and services

The start and stop information on IMS Tools Base servers is available in Starting and stopping various IMS Tools Base servers in the *IBM IMS Tools Base for z/OS Configuration Guide*. Also, refer to the following documentation for individual servers and services:

- IBM IMS Tools Base for z/OS Distributed Access Infrastructure User's Guide and Reference
- IBM IMS Tools Base for z/OS IMS Tools Knowledge Base User's Guide and Reference
- IBM IMS Tools Base for z/OS Autonomics Director User's Guide and Reference
- IBM IMS Tools Base for z/OS Policy Services User's Guide and Reference
- IBM IMS Tools Base for z/OS IMS Tools Common Services User's Guide and Reference

Setting up security for IMS Connect servers

This section covers security setup procedures that may become necessary for the Unified Management Server to communicate IMS Connect servers securely.

Making CA certificates for IMS Connect servers available for UMS

The TCP/IP communication from the Unified Management Server to IMS Connect servers can be secured by using Transport Layer Security (TLS).

The following IMS features use IMS Connect servers and are eligible for secure communication with TLS:

- The IMS SQL processor uses IMS Universal JDBC Driver
- The IMS command processor uses IMS Connect Client for Java API to issue IMS type-1 and type-2 commands through IMS Operations Manager

These two features support TLSv1.2 for TLS connections to IMS Connect. Lower protocol versions that are supported by IMS Universal JDBC Driver or IMS Connect Client for Java are not supported under the Unified Management Server.

TLS provides server authentication and client authentication. Server authentication is appropriate in situations where the Unified Management Server needs to ensure that it is communicating with the correct IMS Connect servers. There are two methods for setting up server authentication between the Unified Management Server (as a client) and IMS Connect servers (as servers):

- Using SAF keyring for the Unified Management Server
- Using file-based certificate management

Using SAF keyring for the Unified Management Server are the recommended method for enabling the TLS connection although the file-based certificate management is also covered in this section.

In the server authentication, the IMS Connect server can serve any client. In cases where proof of the Unified Management Server's identity is also important, use the client authentication (or mutual authentication), which builds upon server authentication. Client authentication is optional.

For details of configuring IMS Connect as a server with TLS support with using IBM z/OS Communications Server Application Transparent Transport Layer Security (AT-TLS) and policy agent on z/OS, see the IMS documentation.

For the procedure to prepare necessary CA certificates to be used for TLS connections to IMS Connect server ports and add them to the truststore to be used by the Unified Management Server, see “Setting up secure communication for UMS” on page 49.

Setting up TLS client authentication for IMS Connect servers

If TLS client authentication is configured by the AT-TLS policy configuration for an IMS Connect server port that you will use from the Unified Management Server, the IMS Connect server needs to have a certain CA certificate in the server's keyring that is used as the server's truststore. The certificate is the CA certificate for the root CA for the signing chain of the certificate for the Unified Management Server.

Tip: Use client authentication in a production environment to protect against unauthorized access to IMS Connect.

If the root CA certificate for the signing chain of the certificate for the Unified Management Server does not exist in the RACF or other ESM's database, you first need to add it to the database. Then, you need to connect the root CA certificate to the keyring that is specified, in the AT-TLS policy, for an IMS Connect server port to be used from UMS through AT-TLS. This must be done for each keyring for an IMS Connect server port that is to be used from the Unified Management Server.

To connect a CA certificate, especially a shared certificate in a keyring, to an IMS Connect server's started task ID, the started task ID may need to have certain SAF privileges for some specific `IRR.DIGICERT` resources in the `FACILITY` class. For details, see the security administration guide for your security manager. For example, if you are using RACF, see the RACF and digital certificates section in z/OS Security Server RACF Security Administrator's Guide.

Enabling IMS Connect to receive RACF PassTickets

If you want to use the UMS authentication type of MFA_JWT, you must configure IMS Connect DRDA interface for JDBC connections and IMS Connect interface for OM commands.

Note: The IMS Connect server must be configured with RACF=Y to enable PassTicket support for each IMS Connect server to be used from UMS.

For configuration details of PassTicket support for IMS Connect DRDA interface, see Enabling RACF PassTicket for UMS server. The application name given and defined by your security administrator for PassTicket-based authentication must be specified in the ODACCESS statement of the HWSCFxxx configuration member for the IMS Connect server. For details, see the ODACCESS statement section of the *IMS System Definition Guide*.

For configuration details of PassTicket support for IMS Connect interface for OM commands, see Enabling RACF PassTicket for UMS server. The application name given and defined by your security administrator for PassTicket-based authentication must be specified for each IMS Connect port that is selected as the UMS IMS Command Processor port for an IMS data sharing group when you register it.

At UMS run-time, PassTickets for a UMS login user is generated by the Zowe API Mediation Layer gateway that is called by UMS.

Setting up security for IMS Tools TCP server

This section covers security setup procedures that may become necessary for the Unified Management Server to communicate securely with the Distributed Access Infrastructure (DAI) TCP server for IMS Tools.

Making CA certificates for DAI TCP server available for UMS

The TCP/IP communication from the Unified Management Server to the Distributed Access Infrastructure (DAI) TCP server for IMS Tools can be secured by using Transport Layer Security (TLS).

TLS provides server authentication and client authentication. Server authentication is appropriate in situations where the Unified Management Server needs to ensure that it is communicating with the correct DAI TCP server.

Tip: Use client authentication in a production environment to protect against unauthorized access to the TCP Server.

In the server authentication, the DAI TCP server can serve any client. In cases where proof of the Unified Management Server's identity is also important, use the client authentication (or mutual authentication), which builds upon server authentication. Client authentication is optional.

For the server authentication, the setup procedure is similar to the one for IMS Connect servers. See IMS documentation

Setting up TLS client authentication for DAI TCP Server

If TLS client authentication is configured by the AT-TLS policy configuration for the TCP server port that you will use from the Unified Management Server, the TCP server needs to have a certain CA certificate in the TCP server's keyring that is used as the server's truststore. The certificate is the CA certificate for the root CA for the signing chain of the certificate for the Unified Management Server.

If the root CA certificate for the signing chain of the certificate for the Unified Management Server does not exist in the RACF or other ESM's database, you first need to add it to the database. Then, you need to connect the root CA certificate to the keyring that is specified, in the AT-TLS policy, for the TCP server port to be used from UMS through AT-TLS. Complete the steps described in “Setting up secure communication for UMS” on page 49.

To connect a CA certificate, especially a shared certificate in a keyring, to the TCP server's started task ID, the started task ID may need to have certain SAF privileges for some specific IRR.DIGICERT resources in the FACILITY class. For details, see the security administration guide for your security manager. For example, if you are using RACF, see the RACF and digital certificates section in *z/OS Security Server RACF Security Administrator's Guide*.

Enabling IMS Tools TCP server to receive RACF PassTickets

If you want to use the UMS authentication type of MFA_JWT, you must configure the login interface for the TCP server of IMS Tools Base Distributed Access Infrastructure (DAI).

To secure connections to the DAI TCP server by using PassTickets, you must define the following profiles in IBM RACF:

- An application profile in the APPL class

  You have to permit appropriate UMS user IDs to this profile for whom you want to allow PassTicket-based authentication.
- A same named profile in the PTKTDATA class

  You have to permit to this profile the Zowe started task user ID and optionally MFA server started task user ID.

For detailed configuration steps and required access levels, see Enabling RACF PassTicket for UMS server and “Configuring multifactor authentication for UMS” on page 235.

To enable PassTicket support for a DAI TCP server to be used from UMS, the TCP server must be configured by specifying the defined application name in the SecurityAppl parameter in the TCP server configuration member and the same application name in the UMS PARMLIB member IZPIMFPM. For details on how to specify the application name in these configuration members, see Installing IMS Administration Foundation.

At UMS run-time, PassTickets for a UMS login user are generated by the Zowe API Mediation Layer gateway that is called by UMS.

Enabling RACF PassTicket for UMS server

You can use RACF PassTickets to authenticate UMS connections to z/OS backend services, such as Db2 or IMS subsystem, or subsystem tool service.

Before you begin

To secure connections to z/OS backend services by using RACF PassTickets, you must define the following in RACF: PassTicket class, application profile, application name, and UMS user IDs (for whom you want to allow PassTicket-based authentication).

Procedure

1. Activate the PTKTDATA class. This is the class to which all profiles that contain PassTicket information are defined. To activate the class and the function, enter the following command:

   ```
   SETROPTS CLASSACT(PTKTDATA) RACLIST(PTKTDATA)
   ```

2. To define the name of the application for which the UMS server requires access using the PassTicket, enter the following commands:

   ```
   RDEFINE APPL <applname> UACC(NONE)
   SETROPTS CLASSACT(APPL)
   SETROPTS GENERIC(PTKTDATA)
   ```

   Where *applname* is a 1- to 8-character name given for the application. The site security administrator usually provides the application name based on certain naming conventions.
3. To define a profile for the application, enter the following command:

   ```
   RDEFINE PTKTDATA <applname> SSIGNON(<key_description>(<key>))
   ```

   The profile associates a secret and secured sign-on application key with the application. Where *applname* is the 1- to 8-character application name that you defined in the previous step. For details of the PTKTDATA class profile definition, see *z/OS Security Server RACF Security Administrator Guide*.
4. To permit a UMS user ID to the application, enter the following command:

   ```
   PERMIT APPLNAME CLASS(APPL) ID(<userid>) ACCESS(READ)
   ```

   Where *userid* is the user ID that is permitted to access the application.
5. To permit the Zowe STC ID to use the PassTicket generation service, enter the following commands:

   ```
   RDEFINE PTKTDATA IRRPTAUTH.<applname>.* UACC(NONE)
   PERMIT IRRPTAUTH.<applname>.* CLASS(PTKTDATA) -
   ID(<Zowe_STC_User>) ACCESS(UPDATE)
   ```

   Where *applname* is the application name defined in the prior step and *Zowe_STC_User* is the Zowe STC user ID to be used for the subject UMS server.
6. To refresh the PTKTDATA class and activate the changes, enter the following commands:

   ```
   SETROPTS RACLIST(APPL) REFRESH
   SETROPTS RACLIST(PTKTDATA) REFRESH
   ```

Defining a security class for UMS

This section covers the steps to define IZP class for SAF and migrate them to a different POSIT number.

Defining the SAF IZP class

If you want to create an IZP class before running the IZPGENER JCL or do not want to use the IZP class definition JCL created by IZPGENER, you need to define the SAF IZP class for the RACF security manager.

For example,

- Create the SAF IZP class for RACF security manager:

```
RDEF CDT IZP UACC(NONE)
RALT CDT IZP CDTINFO(POSIT(608))
RALT CDT IZP CDTINFO(RACLIST(ALLOWED))
RALT CDT IZP CDTINFO(MAXLENGTH(246))
RALT CDT IZP CDTINFO(FIRST(ALPHA))
RALT CDT IZP CDTINFO(OTHER(ALPHA NUMERIC SPECIAL))
SETR RACLIST(CDT) REFRESH
SETR CLASSACT(IZP) GENERIC(IZP) RACLIST(IZP)
```

Notes:

- For ACF2 commands, refer to SMP/E install data set member SIZPCUSA(IZPB1A).
- For TSS, refer to SMP/E install data set member SIZPCUST(IZPB1T).

Migrating to a different POSIT number

If you have already run the security JCLs, you can complete the below steps to migrate to a different POSIT number.

The default POSIT number specified by IBM is `608`. Identify the new POSIT number required for this procedure.

To migrate to a different POSIT number, complete the following steps:

1. Execute the following command to list the SETROPTS options for all classes:SETROPTS LIST

   ```
   SETROPTS LIST
   ```

   Record all active system options for the IZP class.
2. Record the current POSIT value of the IZP class. Run the following command to list the POSIT value:

   ```
   RLIST CDT IZP CDTINFO NORACF
   ```

3. Run the following command to change the POSIT number:

   ```
   RALTER CDT IZP CDTINFO(POSIT(60*8*))
   ```

   Ignore the `IRR52190I` message that is issued by `RALTER`.
4. Run the following command to refresh the CDT class on all systems sharing the RACF database that will use the IZP class:

   ```
   SETROPTS RACLIST(CDT) REFRESH
   ```

   Ignore the following `ICH14079I` message that is issued by `SETROPTS`: `ICH14079I RACF detected an error in the dynamic class descriptor table entry IZP, error code 08.`

   This message will also be issued during IPL on any system with the PTF for z/OS 2.5 and above, until step `10` is performed. These IPL messages can be ignored.
5. Activate the desired `SETROPTS` options. Using the `SETROPTS  LIST` output from Step 1 as reference, assuming for this example that `SETROPTS` options `CLASSACT`, `RACLIST`, `GENERIC`, and `GENCMD` were previously active for the IZP class, run the following command:

   ```
   SETROPTS CLASSACT(IZP) RACLIST(IZP) GENERIC(IZP) GENCMD(IZP)
   ```

6. Examine all dynamic and static CDT entries to see if any other existing class shares the previous POSIT value of the IZP class. If another existing class shares the current POSIT value, then continue at step 10. If no other existing class shares the previous POSIT value, continue with step 7 to ensure that any new class will not have unexpected options if you add a new class using that POSIT value in the future.

7. Add a new, unique, temporary dynamic class and assign it the previous POSIT value of the IZP class. For example, if the class name $TEMPCLS is not in use, and the previous POSIT value of the IZP class was 200, then run the following commands:

   ```
   RDEFINE CDT $TEMPCLS CDTINFO(POSIT(200))
   SETROPTS RACLIST(CDT) REFRESH
   ```

8. Deactivate the SETROPTS settings that you recorded in step 1. For example, if the SETROPTS options CLASSACT, RACLIST, GENERIC and GENCMD were active for the IZP class, you can issue the following command to deactivate those options.

   ```
   SETROPTS NOCLASSACT($TEMPCLS) NORACLIST($TEMPCLS) NOGENERIC($TEMPCLS) NOGENCMD($TEMPCLS)
   ```

9. Delete the temporary class by running the following commands:

   ```
   RDELETE CDT $TEMPCLS
   SETROPTS RACLIST(CDT) REFRESH
   ```

10. Once all of the systems sharing the RACF database are loaded with the PTF for z/OS 2.5 and above, the installation-defined dynamic version of the IZP class can be deleted. Delete the CDT class profile which defines the IZP class using the commands:

    ```
    RDELETE CDT IZP
    SETROPTS RACLIST(CDT) REFRESH
    ```

    Important: Do not delete the installation-defined dynamic class until all of the systems sharing the RACF database have been loaded with the PTF for z/OS 2.5 and above. If you are propagating changes to the CDT class using the RACF remote sharing facility (RRSF), then this also applies to systems on remote RRSF nodes.

References for system programmers

This section documents reference information that system programmers might find useful while installing and configuring Unified Management Server.

Subsections:

Basic concepts of z/OS UNIX

Understand the basic concepts of z/OS UNIX before you start installing and configuring Unified Management Server.

Execution path

UNIX System Services finds executable files and scripts to run by looking in a path environment variable. You can run files not in your path by specifying their locations, either with absolute or relative path names on the command line.

Working directory

In UNIX System Services, there is a concept of a working directory, which will give a default for actions of many commands, and the "starting point" for a relative path name.

Absolute path name

An absolute path name is one that begins with a slash (/). For example:

```
/usr/lpp/IBM/izp/v1r2m0/bin
```

Relative path name

A relative path name begins with any other character.

Special folders

There are two special folders "." and "..", which refer to the current directory and the parent directory, respectively. When issuing a command on the command line that refers to a script file in the current directory, you sometimes have to specify this as "`./script.sh`" if the current directory is not in your path.

Using z/OS UNIX from batch, TSO/E, and ISPF

You can access z/OS UNIX services from batch, TSO/E, or ISPF, using:

- Job control language (JCL) to run shell scripts or z/OS UNIX application programs as batch (background) jobs. This information describes the JCL that supports the z/OS UNIX file system.
- Executable files in batch. An *executable file* is any file that can be run as a program. An executable file can be a load module (which is a member of a PDS), a program object (which is either a member of a PDSE or a file in the z/OS UNIX file system), or an interpreted file (such as a REXX EXEC). For a file to be treated as an executable file, it must have execute permission allowed for the invoker.
- BPXBATCH, a utility that can do the following:
  - Run executable files in batch.
  - Run shell commands and executable files from the TSO/E READY prompt.
- TSO/E commands designed to work with MVS data sets. For the complete command descriptions, see "TSO/E commands" in *z/OS UNIX System Services Command Reference*.
- REXX programs that are written using z/OS UNIX extensions called *syscall commands*.
- The ISPF shell.

For details, see the topic "Using z/OS UNIX from batch, TSO/E, and ISPF" in *z/OS UNIX System Services User's Guide*.

Logging in to z/OS UNIX System Services (USS) using the ISPF shell

If you are an experienced MVS user, you might prefer to use the ISPF shell instead of shell commands or TSO/E commands to work with the z/OS UNIX file system. You can access the ISPF shell from a TSO/E command, or you can access it from within ISPF.

- To access the shell from the TSO/E Ready prompt, enter:

  ```
  ISHELL
  ```

- To access the shell from the Option line of an ISPF panel, enter:

  ```
  TSO ISHELL
  ```

- If you have the appropriate ISPF menu option installed, you can access the shell from the primary ISPF panel. Select option 6 and follow the panel instructions to access the shell.

For details, see the topic "Accessing the UNIX System Services ISPF shell" in *IBM InfoPrint XT for z/OS*.

Using ISPF to edit a z/OS UNIX file

ISPF Edit provides a full-screen editor you can use to create and edit z/OS UNIX files. You can access ISPF Edit in several ways:

- Using the **`oedit`** shell command
- Using the TSO/E **`OEDIT`** command at the TSO/E READY prompt or from the shell command line
- From the ISPF menu (if a menu option is installed)
- From the ISPF shell (accessed using the TSO/E **`ISHELL`** command)

For details, see the topic "Using ISPF to edit a z/OS UNIX file" in *z/OS UNIX System Services User's Guide*.

UNIX System Services commands

Here are several useful commands.

`cd [`*`directory`*`]`

The **cd** command stands for "change directory". This will change your working directory to either your home directory (if no parameters are given), or to the specified directory.

`extattr [`*`flag`*`] [`*`file`*`]`

The **`extattr`** command sets extended attributes on a file. The extended attributes are lost in various kinds of copy-type operations and might need to be reset. To set the program control attribute, enter:

```
extattr +p <filename>
```

`export VARIABLE=`*`value`*

UNIX System Services relies heavily on environment variables, which control a lot of different actions. The most important of these in terms of Unified Management Server is _BPXK_AUTOCVT. A number of files are provided in ASCII format, and these are tagged ISO8859-1. With _BPXK_AUTOCVT=ON, these files will be automatically read in the proper code page. We highly recommend adding the following command to a profile:

```
export _BPXK_AUTOCVT=ON
```

`ls`

The **`ls`** command displays a directory listing and is equivalent to 3.4 in ISPF. Syntax is:

```
ls -<one or more options>
```

Options you might find useful during Zowe installation include `-l` for long form display and `-a` to include all files including hidden files. Options can be used singly or combined. For example:

```
> ls -la
```

For complete usage notes, enter:

```
> man ls
```

pwd

The **pwd** command stands for "print working directory".

The following list contains some of other useful commands, categorized by usage:

Purpose	Commands

Manipulating files	`cp`
	`mv`
	`rm`

Viewing files	`cat` `more`
Manipulating directories	`mkdir` `rmdir`

Purpose	Commands

Changing permissions	`chgrp`
	`chmod`
	`chown`

Manipulating and viewing processes	`ps` `kill`

z/OS-specific commands	`ls -lET`
	`pax`
	`chtag`
	`tsocmd`
	`opercmd ($ZOWE_ROOT/scripts/internal/opercmd)`

Advanced commands	`ln`
	`grep`
	`vi` (editor)

For a complete documentation of all UNIX System Services commands, see *z/OS UNIX System Services Command Reference*.

Terminology reference

This section documents z/OS Security Services terminology reference information.

SAF

System Authorization Facility (SAF) is an interface that enables programs to use services to control access to resources such as data sets.

security profile

A security profile provides a secure way to protect resources on your system. If a user has access to the profile that protects a resource, then they are able to view the resource. To create a security profile, you must enter commands specific to your security manager. If you do not know what security manager you have installed, ask your security administrator. In order to run these commands, your TSO ID must have elevated privileges, which vary depending on your installed security manager. Note, that each user should only have access to one of the created profiles to ensure a proper configuration.

security class

A security class is a container for security profiles. The security profiles present in a class are specific to that class. . To create a security class, you must enter commands specific to your security manager. If you do not know what security manager you have installed, ask your security administrator. In order to run these commands, your TSO ID must have elevated privileges which vary depending on your installed security manager. Note that this class name must be IZP.

surrogate user

A surrogate user is used in place of a regular user to access protected resources. A surrogate user will usually not have a password, so if only a surrogate user has access to a resource, impersonation will have to be used to act as that user and access the resource. This is for security purposes, as a user will not be able to access protected resources outside of the product. Note that surrogate users created for this product must have an OMVS segment.

Chapter 12. APIs

Data management products provide their functionality through REST APIs. Swagger documentation for these APIs is available in HTML format in the Swagger editor.

Swagger documentation

When the Unified Management Server is running, you can open API Swagger documentation by opening the following URL in a browser:

```
https://<host>:<port>/ws/swagger-ui.html
```

where *<host>:<port>* are the host name or IP address, and the port number, of the Unified Management Server host computer. The port number is specified by the **`components.izp.server.port`** parameter, and its default value is 12023.

By default, the documentation displays the latest version. To find an older version, select it from the drop-down list. Only the endpoints of data management experiences that have been purchased and activated are available. Deprecated endpoints are gray, and the text is struck through.

Specifying versions in REST calls

To specify the version of an end point to use in a REST API call, you can use the `X-API-VERSION` header with any of the following values:

- No value (or not using the header): Calls the newest available version of the endpoint.
- `LATEST`: Calls the newest available version of the endpoint.
- `EARLIEST`: Calls the earliest available version of the endpoint.
- *`<version>`*: Calls the specified version, for example "1.1.0.0". You can specify any version that is available in the Swagger documentation.

Tip: It is recommended to use `X-API-Version` for pipelines or programs that leverage UMS APIs.

If you have coded the pipeline or programmed the plug in for Unified Management Server 1.1 (UI78605), pass the `X-API-VERSION` header. For example, pass `X-API-Version: 1.1.0.8` for Unified Management Server 1.1 (UI78605). This ensures your pipeline or program will work even if features are changed or scheduled to be removed. You can upgrade the `X-API-VERSION` header after testing future PTFs in your environment.

For information on deprecated and removed functions in Unified Management Server 1.2, see "Deprecated and removed functions in Unified Management Server" on page 18.

New endpoints support

All `REST API` endpoints will authenticate incoming requests using the data in `HTTP Auth Header`. The `Auth Header` value must be `"Bearer <access token>"`. The following three endpoints are supported in IBM Unified Management Server for z/OS 1.1.0.3.

Session start: Invoke the login endpoint to initiate your session.

```
POST ws/security/login
```

```
body {"id" : "username", "password" : "password"}

response

{
    "id": "username",
    "accessToken": "access-token",
    "refreshToken": "refresh-token",
    "accessTokenExp": 1614102132084,
    "refreshTokenExp": 1614186732084
}
```

Session operations: Invoke other REST endpoints that form the main part of the session. During the session, invoke the refresh-token endpoint whenever you need to refresh the `accessToken`.

```
POST ws/security/refresh-token
```

```
body

{
    "id":"username",
    "refreshToken":"refresh-token"
}

response

{
    "id": "username",
    "accessToken": "access-token",
    "refreshToken": "refresh-token",
    "accessTokenExp": 1614102132084,
    "refreshTokenExp": 1614186732084
}
```

Session end: End the session by invoking the logout endpoint.

```
POST ws/security/logout
```

```
body

{
    "id":"username"
}

response:

http status 200
```

Login endpoint issues access and refresh tokens. The following default validity is applied:

- Default access token: 30 minutes
- Refresh access token: 24 hours

Expiration time is epoch time in milliseconds. Refresh endpoint is used to refresh the access token. When the access token expires, users can use the refresh token to get a new access token. Logout operation is used to clear tokens in Unified Management Server.

URL encoding

Modern APIs accept both `<uuid>` and `<name>` as parameters. If you are using `<name>` as a parameter or in the URL, we recommend encoding the URL to handle special characters, such as #, ?, %, /, and others. Refer to the following code samples:

- Jenkins

  ```
  siteRuleNameConverted = URLEncoder.encode(siteRuleName, "UTF-8")
  ```

- Python

  ```
  from urllib.parse import urlparse
  siteRuleNameConverted = urllib.parse.quote_plus(siteRuleName)
  ```

Specifying error response levels

When you start the Unified Management Server, you can add `apiErrorResponseLevel` flag to set the level of detail in error messages returned by the API. For example:

```
-apiErrorResponseLevel <0|1|2>
```

where a value of "0" is minimal details, a value of "1" is partial details, and a value of "2" is full details. If a detail level is not specified when you start the server, the default level is 0.

Support for new APIs

To fetch the pull request details, following new APIs are added in the Db2 DevOps Experience 1.2.0.5 release.

/policy/pull-requests/report
Get pull request reports based on Category, Status, and Date Time Range.

/policy/pull-requests/report/application/{applictionId}
Get pull request reports for an application based on the Category, Status, and Date range.

/policy/pull-requests/report/instance/{instanceId}
Get pull request reports for an instance based on the Category, Status, and Date range.

/policy/pull-requests/report/team/{teamId}
Get pull request reports for a team based on the Category, Status, and Date range.

Notices

This information was developed for products and services offered in the U.S.A.

This material may be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

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

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

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785

U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.html.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Other company, product, and service names may be trademarks or service marks of others.

Terms and conditions for product documentation

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

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

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

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

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

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

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

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

Privacy policy considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at http://www.ibm.com/privacy and the section titled "Cookies, Web Beacons, and Other Technologies" in IBM's Online Privacy Statement at http://www.ibm.com/privacy/details. Also, see the "IBM Software Products and Software-as-a-Service Privacy Statement" at http://www.ibm.com/software/info/product-privacy.

Index

Product Number: 5698-UM1

SC27-9151