Oracle®

Oracle Financial Services Model Management and Governance Installation and Configuration Guide

Release 8.1.3.3.0
G18356-15
October 2025

Oracle Oracle Financial Services Model Management and Governance Installation and Configuration Guide, Release 8.1.3.3.0

G18356-15

Contents

5 Installation

6 Post-installation Steps

7 Upgrade Installation

8 Update Utility to Reconfigure Installation Parameters

9 Cloning the MMG Instance

10 Frequently Asked Questions (FAQs) and Error Dictionary

1 Preface

This section provides information about the Oracle Financial Services Model Management and Governance (OFS MMG) Installation and Configuration Guide.

Topics:

Related Topics

Audience

OFS MMG Installation and Configuration Guide is intended for Administrators and Implementation Consultants who handle installing and maintaining the Application Pack Components.

This document assumes that you have experience in installing Enterprise Components and basic knowledge about the following:

- OFS AAI Components
- OFSAA Architecture
- UNIX Commands
- Database Concepts
- Web Server or Web Application Server

Additional Resources

This section identifies additional resources to the OFS MMG Application. You can access the following documents from the Oracle Help Center:

- OFS Model Management and Governance Release Notes
- OFS Model Management and Governance User Guide

Additional related documents are as follows:

- OFS Analytical Applications 8.1.2.0.0 Technology Matrix

> **Note**
>
> In the Tech Matrix, the Java Version column is no longer applicable for release 8.1.3.3.0. Instead, Java "17" 2021-09-14 LTS is applicable.

Conventions

The following text conventions are used in this document:

Table 1-1 Document Conventions

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, file names, text that appears on the screen, or text that you enter.
Hyperlink	Hyperlink type indicates the links to external websites, internal document links to sections.

Abbreviations

The following table lists the Abbreviations used in this document:

Table 1-2 Abbreviations

Abbreviation	Meaning
BDP	Big Data Processing
DBA	Database Administrator
DDL	Data Definition Language
DEFQ	Data Entry Forms and Queries
DML	Data Manipulation Language
EAR	Enterprise Archive
EJB	Enterprise JavaBean
ERM	Enterprise Resource Management
FTP	File Transfer Protocol
HDFS	Hadoop Distributed File System
HTTPS	Hypertext Transfer Protocol Secure
J2C	J2EE Connector
J2EE	Java 2 Enterprise Edition
JCE	Java Cryptography Extension
JDBC	Java Database Connectivity
JDK	Java Development Kit
JNDI	Java Naming and Directory Interface
JRE	Java Runtime Environment
JVM	Java Virtual Machine
LDAP	Lightweight Directory Access Protocol
LHS	Left Hand Side
MFA	Multi-Factor Authentication

Table 1-2 (Cont.) Abbreviations

Abbreviation	Meaning
MOS	My Oracle Support
OFSAA	Oracle Financial Services Analytical Applications
OFS AAI	Oracle Financial Services Analytical Application Infrastructure
OFS MMG	Oracle Financial Services Model Management and Governance Application
OHC	Oracle Help Center
OLAP	On-Line Analytical Processing
OLH	Oracle Loader for Hadoop
ORAAH	Oracle R Advanced Analytics for Hadoop
OS	Operating System
RAM	Random Access Memory
RDBMS	Relational Database Management System
RHEL	Red Hat Enterprise Linux
SFTP	Secure File Transfer Protocol
SID	System Identifier
SSL	Secure Sockets Layer
TNS	Transparent Network Substrate
URL	Uniform Resource Locator
VM	Virtual Machine
WAR	Web Archive
XML	Extensible Markup Language
PGX	Parallel Graph AnalytiX
FQDN	Fully Qualified Domain Name

2

Introduction

Financial Institutions require models that work on traditional statistical techniques, modern machine-learning methods, computational and simulation models. Oracle Financial Services Model Management and Governance leverage the Data Studio environment to develop, deploy, and manage models at the enterprise level.

The OFS Model Management and Governance Application enables institutions to implement their IT policies while providing flexibility and freedom that Data Scientists and Statistical Modelers desire. OFS MMG's design facilitates financial institutions to manage external regulatory and internal governance policies by building testing models in a workspace environment. A workspace is provisioned and authorized for use (usually by an Administrator) before making it available to modelers. Administrative users grant analysts and modelers access to workspaces along with a subset of production data to build models. Validated and approved models can then be promoted from workspaces to the enterprise model repository. Models in the repository can then be woven into analytical application flows crafted by mixing data management tasks, model execution, and deterministic business logic.

Topics:

Related Topics

-
- Components of Oracle Financial Services Model Management and Governance

Components of Oracle Financial Services Model Management and Governance

The following are the components of Oracle Financial Services Model Management and Governance Application:

- Workspace Management
- Model Management
- Dataset
- Model Pipelines
- Model Actions
- Graphs
- Scheduler Service
- Audit Trail
- Data Studio Options
- Object Migration
- Model Training

For more information on how to use the application, see the OFS Model Management and Governance User Guide.

Installation Check List

To complete the installation process, you must perform the following steps listed in the Pre-install Checklist. Use this checklist to verify whether these steps are completed or not.

Table 2-1 Installation Checklist

Sl. No.	Activity
Pre-installation Steps	
1	Install all the prerequisite *hardware and software* as per the OFS Analytical Applications 8.1.2.0.0 Technology Matrix.
2	Configure the Database Instance Settings. **Note** In the Tech Matrix, the Java Version column is no longer applicable for release 8.1.3.3.0. Instead, Java 17.x is applicable.
3	Create the Installation, Download, and Metadata Repository Directories: • Installation Directory • Temporary Directory • Staging Area/Metadata Repository • Download Directory
4	Configure the following Operating System and File System Settings: • File Descriptor • Total number of processes • Port(s) • .profile file permissions • Add FTP or SFTP Configuration for file transfer (to access Staging Area and Metadata Directory)
5	Update the following Environment Settings as required for the installation in the .profile file: • Java Settings – Oracle Database Server and Client Settings – Add TNS entries in the TNSNAMES.ORA file – Time Zone Settings
Installation Steps	
6	Download the Installer Kit.
7	Extract the Installer Kit.
8	Configure the config.sh file.
9	Trigger the Application Installation.

Table 2-1 (Cont.) Installation Checklist

Sl. No.	Activity
Post-Installation Steps	
10	Access the MMG Application.
11	Create Application Users.
12	Map Application User(s) to User Groups.

> **Note**
>
> MMG v8.1.3.3.0 supports Python 3.8.x, Python 3.9.x, Python 3.10.x, Python 3.11.x, and Python 3.12.x. Apache Flink packages are not supported with Python 3.12.x.

3
Hardware and Software Requirements

The following are the hardware and software requirements:

Table 3-1 Hardware and Software Requirements

Category	Version
Processor Type	Linux x86-64
OS Version	Oracle Linux Server release 8.4 and later versions
Java Version	OpenJDK 17.0.x
Database Server	Oracle Database 19c (19.3+) Enterprise Ed.
Java	Version 17 and 21
Oracle Database	Version 19 and 23
Python	Version 3.8-12
Python	Version 3.10
OS	OEL9, OEL8 and Solaris

License Information

For details on the third-party software tools used, see the OFSAA Licensing Information User Manual Release 8.1.2.0.0.

4 Preinstallation

This section lists all the prerequisites to install OFS MMG.

Oracle Database Instance Settings

Ensure that the following database instance settings are configured:

- NLS_CHARACTERSET to AL32UTF8
- NLS_LENGTH_SEMANTICS to BYTE
- OPEN CURSORS limit to greater than 1000

Create the Database Schema on Oracle Database

Create the following Database Schemas:

- MMG Studio Schema
- MMG Graph Schema

Tablespace

You can either use the existing Tablespace or can create a new Tablespace during schema creation using the following script:

Permanent Tablespace

```
CREATE TABLESPACE <tablespace_name >

DATAFILE '<tablespace_name >.dat'

SIZE 1G

ONLINE;
```

Temporary Tablespace

```
CREATE TEMPORARY TABLESPACE <tablespace_name >

TEMPFILE '<tablespace_name >.dbf'

SIZE 100M;
```

Creating an Oracle User

You can create an Oracle user using the following script:

```
CREATE USER <oracle_user_name> IDENTIFIED BY <password> DEFAULT TABLESPACE USERS
TEMPORARY TABLESPACE TEMP QUOTA <quota_size>|UNLIMITED ON USERS
```

Creating the Application Configuration Schema

Create the application schema. You must create an Oracle User to create the application schema. For more details, see Creating an Oracle User section. This section discusses the various grants required for the Oracle Database User.

Assign the Grants

Assign the following grants to the workspace schema user.

```
grant create SESSION to <oracle_database_user>;

grant create PROCEDURE to <oracle_database_user>;

grant create SEQUENCE to <oracle_database_user>;

grant create TABLE to <oracle_database_user>;

grant create TRIGGER to <oracle_database_user>;

grant create VIEW to <oracle_database_user>;

grant create MATERIALIZED VIEW to <oracle_database_user>;

grant select on SYS.V_$PARAMETER to <oracle_database_user>;

grant create SYNONYM to <oracle_database_user>;

grant select on sys.v_$parameter to <oracle_database_user>;

grant select on sys.dba_free_space to <oracle_database_user>;

grant select on sys.dba_tables to <oracle_database_user>;

grant select on sys.Dba_tab_columns to <oracle_database_user>;

grant create RULE to <oracle_database_user>;

grant create any trigger to <oracle_database_user>;

grant drop any trigger to <oracle_database_user>;

grant select on SYS.DBA_RECYCLEBIN to <oracle_database_user>;

create or replace DIRECTORY fs_list_logs_dir AS '/file_store/fs_list/logs/';

grant read, write on DIRECTORY fs_list_logs_dir TO schema_name;

create or replace DIRECTORY fs_list_script_dir AS '/file_store/fs_list/script/';

grant read, execute on DIRECTORY fs_list_script_dir TO schema_name;

create or replace DIRECTORY fs_list_control_dir AS '/file_store/fs_list/
control/';

grant read on DIRECTORY fs_list_control_dir TO schema_name;

-- Directory creation and access grants
create or replace DIRECTORY fs_list_logs_dir AS '/file_store/fs_list/logs/';
grant read, write on DIRECTORY fs_list_logs_dir to &schemaname;

create or replace DIRECTORY fs_list_script_dir AS '/file_store/fs_list/
```

```
script/';
grant read, execute on DIRECTORY fs_list_script_dir to &schemaname;

create or replace DIRECTORY fs_list_control_dir AS '/file_store/fs_list/
control/';
grant read on DIRECTORY fs_list_control_dir to &schemaname;
```

Create the MMG Studio Schema

You must create an Oracle User to create the MMG Studio Schema. For more details, see Creating an Oracle User section.

Assign the following grants:

```
GRANT CONNECT, CREATE TABLE, CREATE VIEW, CREATE SEQUENCE TO
<mmgstudio_schema_name>;
```

A new user group created in MMG which requires permission for Studio operations, is to be added in the following files:

- mmgstudio/conf/application.yml
- mmgstudio/conf/ofsaa-permissions.init.yml

> **Note**
>
> Make sure the groups are in upper case, as AAI groups are always in upper case.

Create MMG Graph Schema

Ensure that create an Oracle User to create the MMG Graph Schema. For more details, see Creating an Oracle User section.

Assign Grants

This section discusses the various grants required for the Graph Schemas.

Assign the following grants for the schema:

1. Pre-installation grants for Graph Schema:

```
GRANT CREATE SESSION TO <GRAPH_SCHEMA>;

GRANT CREATE TABLE TO <GRAPH_SCHEMA>;

GRANT CREATE VIEW TO <GRAPH_SCHEMA>;

GRANT CREATE ANY PROCEDURE TO <GRAPH_SCHEMA>;

GRANT CREATE SEQUENCE TO <GRAPH_SCHEMA>;

GRANT CREATE JOB TO <GRAPH_SCHEMA>;

GRANT CREATE MATERIALIZED VIEW TO <GRAPH_SCHEMA>;

GRANT EXECUTE ON DBMS_SCHEDULER to <GRAPH_SCHEMA>;

GRANT EXECUTE ON DBMS_COMPARISON TO <GRAPH_SCHEMA>;
```

```
GRANT EXECUTE ON DBMS_RLS TO <GRAPH_SCHEMA>;

GRANT EXECUTE ON SYS.DBMS_SESSION TO <GRAPH_SCHEMA>;

GRANT EXECUTE ON DBMS_REDEFINITION TO <GRAPH_SCHEMA>;

GRANT REDEFINE ANY TABLE TO <GRAPH_SCHEMA>;

GRANT SELECT ON SYS.V_$PARAMETER TO <GRAPH_SCHEMA>;

GRANT SELECT ON <DATA_SOURCE_SCHEMA>.<TABLE_NAME> TO <GRAPH_SCHEMA>;
```

Example:

Change the `<DATA_SOURCE_SCHEMA>` to the schema used in the Graph pipeline.

> **Note**
>
> If a user has to execute the custom graph, the same permissions have to be provided for the input tables referred in Custom Graph Pipeline.

Create the Installation, Download, and Metadata Repository Directories

To install the application, create the following directories:

- **OFS MMG Download Directory (Optional):** This is the directory where the downloaded installer or patches can be copied. Create a download directory and copy the OFS MMG Application Pack Installer File (archive). Assign 755 permission to this directory.
- **Temporary Directory:** This is the default temporary directory where the installation files are stored for a short time to support faster installation. Configure adequate space on the `/tmp` directory. It is recommended to allocate more than 10 GB of space. Assign 755 permission to this directory and disable the `NOEXEC` option.

  > **Note**
  >
  > If the `NOEXEC` option is enabled, the extraction of files by the installer into the `/tmp` directory is prevented and the binaries will not execute in the directory, hence resulting in failure of the installation.

- **OFS MMG Installation Directory (Mandatory):** Create an installation directory where the product binaries are installed. Assign 755-user permission to the Installation Directory.
- **OFS MMG Staging/Metadata Directory (Mandatory):** This is a directory to hold the application metadata artifacts and additionally, act as the staging area for the flat files. This directory is also referred to as `FTPSHARE`. Create a Staging or Metadata Repository Directory to copy data files, save data extracts, and so on.
  You can configure this directory on a different mount or under a different user profile.

> **Note**
>
> Ensure that the OFS MMG Staging Directory is not set to the same path as the OFS MMG Installation Directory and is not a sub-directory inside the OFS MMG Installation Directory.

Configure the OS File System Settings and Environment Settings in the .profile File

A `.profile` file is a start-up file of a UNIX User. Create the `.profile` file at the home directory of the logged-in user if it is not already available. The user must have 755 permission on the file to execute it. This file consists of various parameters for Environment Settings, OS, and File System Settings.

To set the parameters for the `.profile` file, login as a non-root user, and configure the environment settings.

> **Warning**
>
> Do not modify any other parameters other than the parameters mentioned in the following subsections.

Configure Operating System and File System Settings

To install the application, configure the operating system and file system settings refer the parameters and configuration actions.

Table 4-1 Configure operating system and file system settings

Parameter	Configuration Action
File Descriptor Settings	In the `sysctl.conf` file, to change the number of file descriptors, do the following as the root user: 1. Edit the following line in the /etc/`sysctl.conf` file: fs.file-max = <value> where <value> is greater than 15000 • Apply the change by running the following command: # /sbin/ sysctl -p **Note** The value specified here is the minimum value to be set for the installation process to go forward. For other modules, this value may depend on the available resources and the number of processes executed in parallel.
Total Number of Process Settings	In the `sysctl.conf` file, set the value to greater than 4096. **Note** The value specified here is the minimum value to be set for the installation process to go forward. For other modules, this value may depend on the available resources and the number of processes executed in parallel.

Configure the Environment Settings

Environment Settings refers to values related to the current environment, like the Operating System or user sessions. To configure the environment settings refer the following topics.

Java Settings

To configure the Java Settings, refer the following table:

> **Note**
>
> In the application, Java 21 is supported.

Table 4-2 Java Settings

Description	Example Value
In the `.profile` file, set JAVA_BIN to include the JDK absolute path.	For example: JAVA 17

Table 4-3 Java Settings

Description	Example Value
In the `.profile` file, set the Java tool options for all versionsJDK-17 and above updates. Ensure that SYMBOLIC links to JAVA installation are not set in the PATH variable.	`JAVA_TOOL_OPTIONS=" -Djdk.util.zip.disableZip64ExtraFieldValidation=true"` `export JAVA_TOOL_OPTIONS`
In the `.profile` file, set JAVA_BIN to include the JDK absolute path.	For example: JAVA_BIN `=/usr/java/jdk-17/bin` `export JAVA_BIN`

Oracle Database Server and Client Settings

To configure the Oracle Database Server and Client Settings, refer to the following table:

Table 4-4 Oracle Database Server and Client Settings

Description	Example Value
In the `.profile` file, set TNS_ADMIN pointing to the appropriate tnsnames.ora file.	`TNS_ADMIN=$HOME/tns`
In the `.profile` file, set ORACLE_HOME pointing to the appropriate Oracle Client installation.	`ORACLE_HOME=/scratch/oraofss/app_client19c/product/ 19.0.0/client_1`
In the `.profile` file, set PATH to include the appropriate $ORACLE_HOME/bin path.	`PATH=$JAVA_HOME/bin:$ORACLE_HOME/bin`

TNS entries in the tnsnames.ora file for Non-TCPS

You must configure the TNS entries in the tnsnames.ora file for Non-TCPS.

Non-TCPS

To configure the TNS entries in the `tnsnames.ora` file for Non-TCPS, refer to the following table:

Table 4-5 TNS entries in the TNSNAMES.ORA file for Non-TCPS

Description	Example Value
Ensure that an entry (with SID or SERVICE NAME) is added in the tnsnames.ora file on the OFSAA server.	<SID_NAME> = DESCRIPTION =(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = <HOST_NAME>.in.oracle.com)(PORT = 1521)))(CONNECT_DATA = (SERVICE_NAME = <SID_NAME>)))<ATOMIC_SCHEMA_NAME> = (DESCRIPTION =(ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = <HOST_NAME>.in.oracle.com)(PORT = 1521)))(CONNECT_DATA = (SERVICE_NAME = <SID_NAME>)))

```
<SID NAME> =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = <HOST NAME>)(PORT = <PORT NUMBER>))

) (CONNECT_DATA =

(SERVICE_NAME = <SID NAME>)

)

)

<ATOMICSCHEMANAME> =

(DESCRIPTION =

(ADDRESS_LIST =

(ADDRESS = (PROTOCOL = TCP)(HOST = <HOST NAME>)(PORT = <PORT NUMBER>))

)

(CONNECT_DATA =

(SERVICE_NAME = <SID NAME>)

)

)
```

Time Zone Settings

In the `.profile` file, set the Time Zone Parameter to indicate the time zone of your region or location.

For more information, see *Oracle Financial Service Stress Testing and Scenario Analytics User's Guide*.

Table 4-6 Time Zone Settings

Description	Example Value
Time Zone	TZ=Asia/Calcutta

Prerequisite for Data Pipeline

The below are prerequisites for Data Pipeline. The steps need to be executed before triggering MMG Installer.

1. Log in as DB user(ex: fccmdb) in your DB server (For example: 1759).
2. Create one directory with preferred reference name (For example: datapipline) at the preferred location.
3. Now execute below commands in putty session of DB server:
   - `mkdir -p ##location of dir created in step 2##/file_store/fs_list/logs` # A directory by the external table to write the logs.
   - `mkdir -p ##location of dir created in step 2##/file_store/fs_list/script` # A directory to hold a pre-processor script used to list the files in a directory. This needs Read-Execute permissions.
   - `mkdir -p ##location of dir created in step 2##/file_store/fs_list/control` # A directory to hold files to control which directories can be listed. This needs Read permissions.
   - `mkdir -p ##location of dir created in step 2##/file_store/fs_list/fccm-data` # A directory to store the csv files.
4. After that login to DB server(For example: 1759) as sysadmin(For example: sys) in sqldeveloper or any preferred application execute below:
   - CREATE OR REPLACE DIRECTORY fs_list_logs_dir AS '##location of dir created in step 2##/file_store/fs_list/logs';
   - GRANT READ, WRITE ON DIRECTORY fs_list_logs_dir TO ##MMG Application Schema##;
   - CREATE OR REPLACE DIRECTORY fs_list_script_dir AS '##location of dir created in step 2##/file_store/fs_list/script';
   - GRANT READ, EXECUTE ON DIRECTORY fs_list_script_dir TO ## MMG Application Schema##;
   - CREATE OR REPLACE DIRECTORY fs_list_control_dir AS '##location of dir created in step 2##/file_store/fs_list/control';
   - GRANT READ ON DIRECTORY fs_list_control_dir TO ##MMG Application Schema##;
   - CREATE OR REPLACE DIRECTORY external_tables_dir AS '##location of dir created in step 2##/file_store/fs_list/fccm-data';
   - GRANT READ ON DIRECTORY external_tables_dir TO ## MMG Application Schema##;
   - GRANT CREATE TABLE TO ##MMG Application Schema##;

   Once the above steps are done, trigger the installation of Compliance Studio build.

> **Note**
>
> - If the new schema is residing on any-other DB server other than the one on which above steps were performed. Please perform all the above steps on that respective server.
> - Once configured the DB server, if new schema is created on it. Need to perform the step for the respective schema.

Setup Password Stores with Oracle Wallet

This section describes the steps to create a wallet and the aliases for the database user accounts. For more information on configuring authentication and password stores, see theOracle Database Security Guide.

As part of an Application Installation, Administrators must set up password stores for Database User Accounts using Oracle Wallet. These password stores must be installed on the Application Database side. The Installer handles much of this process. The Administrators must perform some additional steps.

Setup the Password Stores for Database User Accounts

After the database is installed and the default Database User Accounts are set up, Administrators must set up a password store using the Oracle Wallet. This involves assigning an alias for the username and associated password for each Database User Account. The alias is used later during the application installation. This password store must be created on the system where the Application Server and database client are installed.

> **Note**
>
> In this section, `<wallet_location>` is a placeholder text for illustration purposes. Before running the command, ensure that you have already created the <wallet_location> directory where you want to create and store the wallet.

The wallet is created in the <wallet_location> directory with the auto-login feature enabled. This feature enables the database client to access the wallet contents without using the password. For more information, see Oracle Database Security Guide.

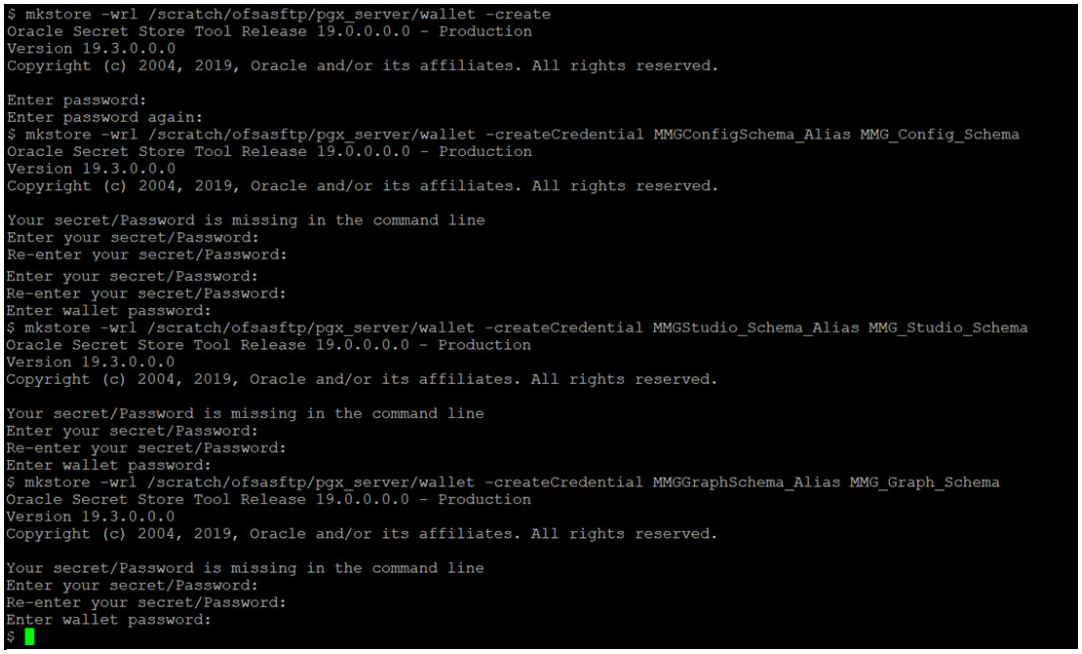
To create a wallet, follow these steps:

1. Login to the server as a Linux user.
2. Create a wallet in the `<wallet_location> using the following command:`
   `mkstore -wrl <wallet_location> -create`

   > **Note**
   >
   > The mkstore utility is included in the Oracle Database Client Installation.

3. After you run the command, a prompt appears. Enter a password for the Oracle Wallet in the prompt.
   A prompt appears to re-enter the password. Re-enter the password.

Figure 4-1 Wallet Creation



4. Create the database connection credentials for the MMG Schema using the following command:
`mkstore -wrl <wallet_location> -createCredential <alias-name> <mmg-schema-name>`

   Here, MMG Schema is the same as explained in Create the MMG Schema section.

   After you run the command, a prompt appears. Enter the password associated with the Database User Account in the prompt. You are prompted to re-enter the password. You are prompted for the wallet password used in Step 1.

   > **Note**
   >
   > In this manner, create a wallet and associated database connection credentials for all the Database User Accounts including Graph and Studio Schema.

   After the wallet is created, go to the `<wallet_location>` directory and click Refresh to view the created wallet folder.

   The wallet folder contains two files: **ewallet.p12** and **cwallet.sso**.

   After you run the command, a prompt appears. Enter the password associated with the Database User Account in the prompt. You are prompted to re-enter the password. You are prompted for the wallet password used in Step 1.

   > **Note**
   >
   > In this manner, create a wallet and associated database connection credentials for all the Database User Accounts including Graph and Studio Schema.

   After the wallet is created, go to the `<wallet_location>` directory and click Refresh to view the created wallet folder.

> **Note**
>
> In addition to creating wallet aliases for MMG Studio and Graph, you must also create wallet aliases for the production schemas; specifically, the config and atomic schemas. Ensure these aliases are added to the `tnsnames.ora` file.

The wallet folder contains two files: **ewallet.p12** and **cwallet.sso**.

5. In the `<wallet_location>` directory, configure the tnsnames.ora file to include the entry for each alias name to be set up. Alias name can be renamed as wallet db alias name.

> **Note**
>
> - You can either update the existing tnsnames.ora file with the above details or create new tnsnames.ora file and add the required entries.
> - <alias-name> is a user-defined value.

6. Create a sqlnet.ora file in the wallet directory using the following content:
```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA =
(DIRECTORY = <Wallet_Location>)) ) SQLNET.WALLET_OVERRIDE=TRUE
SSL_CLIENT_AUTHENTICATION=FALSE
```
Here, `<Wallet_Location>` should be in below format:
```
WALLET_LOCATION = (SOURCE = (METHOD = FILE) (METHOD_DATA =
(DIRECTORY = <WALLET_PATH>) ) )
```

Verify the Connectivity of the Wallet

To verify the connectivity of the wallet, follow these steps:

1. Test the connectivity using the following command:

> **Note**
>
> The ORACLE_HOME used with the wallet must be the same version or higher than the wallet created.

```
$ export WALLET_LOCATION=<wallet_location>

$ export TNS_ADMIN=<tnsnames.ora_location>. If you have created a new
tnsnames.ora file, provide the location of the new file.

$ sqlplus /@<alias_name>
```

The output is similar to:

```
SQL*Plus: Release 11

Connected to:

Oracle Database 12c
```

To verify if you are connected to the correct user:

```
SQL> show user

The output is similar to:

USER is "<database-user-name>"
```

5
Installation

This section provides detailed steps to install the application.

Prerequisites

The list of prerequisites are as follows:

- Port Details
  MMG Studio uses the following ports by default, so make sure these are free:

  Graph Service: 7059

  Server: 7008

  Markdown-Interpreter: 7009, 7029

  Python-Interpreter: 7012, 7032, 6012

  Shell-Interpreter: 7013, 7033

  Plainr-Interpreter 7019, 7039, 6311 (Rserve port- configurable in plainr.Json)

  PGX Server: 7007

  PGX-Interpreter: 7022, 7042

  JDBC-Interpreter: 7011, 7031

  Coherence Cluster: 7574

> **Note**
>
> These are default ports provided in the installation kit. You can modify them in the configuration files if required to avoid conflicts with existing services.

Enhancements for Preinstall Check

Warning messages for below validations:

- To validate whether the given hostname is valid or not.
- To validate whether given hostname and hostname in keystore are matching.
- Error messages and failing the preinstall check for below validations.
- To check sso_token is present or not.
- To check for public and private key are present in the mmghome/conf location.

Download the OFS MMG Installer Kit

To download the software as a .zip folder, download the mandatory minor release patch **38481933** from My Oracle Support (MOS).

Download the installer archive and copy (in Binary Mode) to the download directory that exists in the OFS MMG Installation Setup.

Extract the Software

You must be logged in to the UNIX Operating System as a Non-Root User to perform the following steps. To extract the software, follow these steps:

1. Download the unzip (OS-specific) unzip_<os>.zip and copy it in Binary Mode to the directory that is included in your PATH variable.

   If you already have an unzip utility to extract the contents of the downloaded archive, skip this step. Uncompress the unzip installer file with the command:

   ```
   uncompress unzip_<os>.Z
   ```

   > **Note**
   >
   > If an error message "uncompress: not found [No such file or directory]" is displayed, contact your UNIX Administrator.

2. Assign execute (751) to the file with the following command:

   ```
   chmod 751 unzip_<OS>
   ```

   For example: `chmod 751 unzip_sparc`

3. Extract the contents of the OFS MMG Application Pack Release 8.1.3.3.0 installer archive file in the download directory with the following command:

   ```
   unzip OFS_MMG_8.1.3.3.0_<OS>.zip
   ```

   After unzipping the OFS_MMG_8.1.3.3.0 folder, following zip folders are displayed under OFS MMG folder.

   - mmg-installer.zip
   - mmg-metadata-manager.zip
   - mmg-pgx.zip
   - OFSMMG_8.1.3.3.0_Readme.html

   Unzip mmg-installer.zip and the following components are available under / OFS_MMG/mmg-installer.zip:

   - mmg-ui
   - mmg-studio
   - mmg-service
   - mmg-schema-creator
   - mmg-pipeline
   - mmg-load-to-graph
   - lib
   - bin
   - conf
   - mmg-gateway

- OFSMMG_8.1.3.3.0_Readme.html

4. Navigate to the download directory and assign execute permission to the installer directory with the following command:

   ```
   chmod -R 750 OFS_MMG
   ```

Configure the config.sh file

To configure the `config.sh` file for installing the application, follow these steps:

1. Login to the server as a non-root user.
2. Navigate to the `<installation directory>/OFS_MMG/MMG/OFS_MMG/bin` directory.
3. Configure the applicable `config.sh` attributes as shown here:

   Sample `config.sh` files:

   ```
   #!/bin/sh ##

   ## Common properties

   export APPLICATION_NAME=##APPLICATION_NAME##

   export WALLET_LOCATION=##WALLET_LOCATION##

   export TNS_ADMIN_PATH=##TNS_ADMIN_PATH##

   export WALLET_ALIAS=##WALLET_ALIAS## export LOG_HOME=##LOG_HOME##

   export LOG_HOME=##LOG_HOME##

   export FTPSHARE=##FTPSHARE##

   export LOG_TIMEZONE=##LOG_TIMEZONE##

   ##By default, Data Studio is assumed to run on the same server. If it is
   running on a different host, uncomment the line below and provide the
   appropriate URL. #export DATASTUDIO_URL=##DATASTUDIO_URL##

   export BE_HOSTNAME=##
   BE_HOSTNAME##

   export BE_PORT=##BE_PORT##

   # Add SMTP host and port parameters for email notifications
   export SMTP_HOST=##SMTP_HOST##

   export SMTP_PORT==##SMTP_PORT##

   export UI_PORT=##UI_PORT##

   export SCHEMA_PORT=##SCHEMA_PORT##

   export CONTEXT_PATH=##CONTEXT_PATH##

   export STUDIO_AUTH_TYPE=##STUDIO_AUTH_TYPE##

   export
   RESTRICT_UNMAPPED_DATASTORES_ACCESS=##RESTRICT_UNMAPPED_DATASTORES_ACCESS##

   export SSL_ENABLED=##SSL_ENABLED##
   export SSL_KEYSTORE=##SSL_KEYSTORE##

   export SSL_KS_SECRET=##SSL_KS_SECRET##

   export SSL_KS_TYPE=##SSL_KS_TYPE##
   ```

```
export SSL_KS_ALIAS=##SSL_KS_ALIAS##

export SESSION_TOKEN_CREDENTIALS=##SESSION_TOKEN_CREDENTIALS##

export FCC_API_USER=##FCC_API_USER##

export MMG_DATASOURCE_MAX_POOL_SIZE=##MMG_DATASOURCE_MAX_POOL_SIZE##
export MMG_DATASOURCE_MINIMUM_IDLE=##MMG_DATASOURCE_MINIMUM_IDLE##

export MMG_DATASOURCE_IDLE_TIMEOUT=##MMG_DATASOURCE_IDLE_TIMEOUT##

export MMG_DATASOURCE_CONN_TIMEOUT=##MMG_DATASOURCE_CONN_TIMEOUT##

export EXT_DATASOURCE_MAX_POOL_SIZE=##EXT_DATASOURCE_MAX_POOL_SIZE##

export EXT_DATASOURCE_MINIMUM_IDLE=##EXT_DATASOURCE_MINIMUM_IDLE##

export EXT_DATASOURCE_IDLE_TIMEOUT=##EXT_DATASOURCE_IDLE_TIMEOUT##

export EXT_DATASOURCE_CONN_TIMEOUT=##EXT_DATASOURCE_CONN_TIMEOUT##

export MMG_HTTP_MAX_CONN=##MMG_HTTP_MAX_CONN##
export MMG_HTTP_MAX_CONN_PER_ROUTE=##MMG_HTTP_MAX_CONN_PER_ROUTE##

export MMG_HTTP_CONNECT_TIMEOUT=##MMG_HTTP_CONNECT_TIMEOUT##

export MMG_HTTP_READ_TIMEOUT=##MMG_HTTP_READ_TIMEOUT##

export APPLICATION_ID=##APPLICATION_ID##

## Properties for MMG-ui

export APPLICATION_FAVICON_PATH=##APPLICATION_FAVICON_PATH##

#export UI_AUTH_TYPE=aai

export UI_AUTH_TYPE=##AUTH_TYPE##

export AAI_AUTH_URL=##AAI_AUTH_URL##

export SAML_IDP_URL=##SAML_IDP_URL##

export SAML_SP_ENTITY=##SAML_SP_ENTITY##

export SAML_SRV_URL=##SAML_SRV_URL##

export SAML_LOGOUT_URL=##SAML_LOGOUT_URL##

export SAML_SIGN_AUTHN_REQ=##SAML_SIGN_AUTHN_REQ##
export SAML_PRIVATE_KEY_PATH=##SAML_PRIVATE_KEY_PATH##

export SAML_SP_X509_CERT_PATH=##SAML_SP_X509_CERT_PATH##

export SAML_SIGN_ALGORITHM=##SAML_SIGN_ALGORITHM##

export AUTH_SAML_REQUEST_TYPE=##AUTH_SAML_REQUEST_TYPE##

export AUTH_SAML_INCLUDE_SP_CERT=##AUTH_SAML_INCLUDE_SP_CERT##

export LDAP_URL=##LDAP_URL##
export LDAP_SEARCH_BASE=##LDAP_SEARCH_BASE##

export LDAP_USER_FILTER=##LDAP_USER_FILTER##

export LDAP_USER_SEARCH_FILTER=##LDAP_USER_SEARCH_FILTER##
export LDAP_GROUP_SEARCH_FILTER=##LDAP_GROUP_SEARCH_FILTER##

export LDAP_GROUP_SEARCH_BASE=##LDAP_GROUP_SEARCH_BASE##

export LDAP_GROUP_MEMBER=##LDAP_GROUP_MEMBER##
```

> **Note**
>
> Accepted Values for SAML_SIGN_ALGORITHM Property:
>
> - http://www.w3.org/2001/04/xmldsig-more#rsa-sha512
> - http://www.w3.org/2001/04/xmldsig-more#rsa-sha384
> - http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
> - http://www.w3.org/2000/09/xmldsig#rsa-sha1

```
## Properties for enabling AAI Authorization. Applicable for SAML/LDAP
Profiles
## AAI_AUTH_URL must be set.

export AAI_AUTHZ_ENABLED=##export AAI_AUTHZ_ENABLED##

export AAI_CLIENT_ID=##export AAI_CLIENT_ID##

export AAI_CLIENT_SECRET=<Instance Access Token>

export SERVER_COOKIE_DOMAIN=##SERVER_COOKIE_DOMAIN##
export SERVER_COOKIE_NAME=##SERVER_COOKIE_NAME##

export SERVER_COOKIE_TIMEOUT=##SERVER_COOKIE_TIMEOUT##

export SERVER_COOKIE_IS_SECURE=##SERVER_COOKIE_IS_SECURE##

##Properties for MMG-Service
export BE_AUTH_TYPE=public

export MMG_PYTHON_INTERPRETER=##MMG_PYTHON_INTERPRETER##

#Properties for enabling git based import and export of model pipelines

#set username and PAT(personal access token) to access private remote
repositories

#set proxy url and port if you are behind a firewall

export GIT_ENV_ID=##GIT_ENV_ID##

export GIT_USERNAME=##GIT_USERNAME##

export GIT_PAT_SECRET=##GIT_PAT_SECRET##

export GIT_PROXY_URL=##GIT_PROXY_URL##

export GIT_PROXY_PORT=##GIT_PROXY_PORT##
```

> **Note**
>
> Only tokens (classic) is supported now. Fine grained is not yet supported.

```
##Properties for MMG-Studio
export DATASTUDIO_SCHEMA_WALLET_ALIAS=##DATASTUDIO_SCHEMA_WALLET_ALIAS##

export LOGIN_SHOW=##LOGIN_SHOW##

export SESSION_MODE=##SESSION_MODE##

export STUDIO_REALM=##STUDIO_REALM##
```

```
export OFSAA_URL=##OFSAA_URL##

export API_USERS=##API_USERS##

export VALID_ROLES=##VALID_ROLES##

export STUDIO_LOG_LEVEL=##STUDIO_LOG_LEVEL##

export PYTHON_HOME=##PYTHON_HOME##

#Following are spark specific configurations, Leave as it is if not applicable
export MMG_SPARK_ENABLED=##MMG_SPARK_ENABLED## #will be false by default

export SPARK_HOME=##SPARK_HOME## #required if MMG_SPARK_ENABLED is true

export HADOOP_HOME=##HADOOP_HOME## #only needed if running spark with hadoop
cluster

export SPARK_MASTER=##SPARK_MASTER## #required if MMG_SPARK_ENABLED is true

export SPARK_DEPLOY_MODE=##SPARK_DEPLOY_MODE## #required if MMG_SPARK_ENABLED
is true

export DATASTUDIO_SPARK_INTERPRETER_PORT=##DATASTUDIO_SPARK_INTERPRETER_PORT##
#required if MMG_SPARK_ENABLED is true

export R_ENABLED=##R_ENABLED## export R_PYTHON_HOME=##R_PYTHON_HOME##

export R_PYTHON_HOME=##R_PYTHON_HOME##

# Following are fcc services specific configurations, Leave as it is if not
applicable
export TEMPLATE_CONFIG_PATH=##TEMPLATE_CONFIG_PATH##

export TEMPLATE_DEFAULT_LINK=##TEMPLATE_DEFAULT_LINK##

export AUTH_SERVICE_URL=##AUTH_SERVICE_URL##

export META_SERVICE_URL=##META_SERVICE_URL##

export ER_SERVICE_URL=##ER_SERVICE_URL##

export BATCH_SERVICE_URL=##BATCH_SERVICE_URL##

export SAML_ISSUER=##SAML_ISSUER##

export SAML_DESTINATION=##SAML_DESTINATION##

export SAML_ASSERTION=##SAML_ASSERTION##

export SAML_ROLE_ATTRIBUTE=##SAML_ROLE_ATTRIBUTE##

export SAML_STUDIO_LOGOUT_URL=##SAML_STUDIO_LOGOUT_URL##

export SAML_COOKIE_DOMAIN=##SAML_COOKIE_DOMAIN##

# Following are pipeline services specific configurations, Leave as it is if
not applicable
export DATAPIPELINE_SERVICE_PORT1=##DATAPIPELINE_SERVICE_PORT1##

export DATAPIPELINE_SERVICE_PORT2=##DATAPIPELINE_SERVICE_PORT2##

export
DATAPIPELINE_METADATA_ARCHIVE_PATH=##DATAPIPELINE_METADATA_ARCHIVE_PATH##

export
DATAPIPELINE_METADATA_IMPORT_SERVICE_PORT=##DATAPIPELINE_METADATA_IMPORT_SERVI
CE_PORT##
```

```
export DATAPIPELINE_ERXMLPATH=##DATAPIPELINE_ERXMLPATH##

export DATAPIPELINE_GATEWAY_SERVICE_PORT=##DATAPIPELINE_GATEWAY_SERVICE_PORT##

export PIPELINE_UI_SERVICE_PORT=##PIPELINE_UI_SERVICE_PORT##

export DATA_PIPELINE_UI_SERVICE_PORT=##DATA_PIPELINE_UI_SERVICE_PORT##

export DATAPIPELINE_HAZELCAST_PORT=##DATAPIPELINE_HAZELCAST_PORT##

#URLS for pipeline,ER and matching service. Leave as it is if not applicable.
Will impact the pipeline that could be added to a graph
export MATCHRULE_BASE_URL=##MATCHRULE_BASE_URL##

export LOADGRAPH_BASE_URL=##LOADGRAPH_BASE_URL##

export MATCHSRVC_UI_URL=##MATCHSRVC_UI_URL##

#URLS for index service. Leave as it is if not applicable. Will impact the
pipeline that could be added to a graph
export GRAPH_INDEX_BASE_URL=##GRAPH_INDEX_BASE_URL##

export LOADINDEX_UI_URL=##LOADINDEX_UI_URL##

export MATCHING_MECHANISM=##MATCHING_MECHANISM##

export CANDIDATE_SELECTION_SERVICE_URL=##CANDIDATE_SELECTION_SERVICE_URL##

export LOAD_TO_OS_URL=##LOAD_TO_OS_URL##

#Changes for auth services + mmg keys

export AAI_COOKIE_DOMAIN=##AAI_COOKIE_DOMAIN##

export MMG_KEYS_LOC=##MMG_KEYS_LOC##

#Properties to package Load to Graph (L2G) service inside MMG
## Start of L2G Properties

## export GRAPH_INSTALLATION_PATH=##GRAPH_INSTALLATION_PATH##

export GRAPH_KEYSTORE_PASSWORD=##GRAPH_KEYSTORE_PASSWORD##

export GRAPH_SERVICE_PORT=##GRAPH_SERVICE_PORT##

#Graph HikariCP Configuration Placeholders

export GRAPH_HIKARI_CP_MIN_IDLE=##GRAPH_HIKARI_CP_MIN_IDLE##

export GRAPH_HIKARI_CP_MAX_POOL_SIZE=##GRAPH_HIKARI_CP_MAX_POOL_SIZE##

export GRAPH_HIKARI_CP_IDLE_TIMEOUT=##GRAPH_HIKARI_CP_IDLE_TIMEOUT##

export GRAPH_HIKARI_CP_MAX_LIFETIME=##GRAPH_HIKARI_CP_MAX_LIFETIME##

export GRAPH_HIKARI_CP_TIMEOUT=##GRAPH_HIKARI_CP_TIMEOUT##

## Graph Schema Configurations
export MMG_DB_SERVER_NAME=##MMG_DB_SERVER_NAME##

export MMG_DB_PORT=##MMG_DB_PORT##

export MMG_DB_SERVICE_NAME=##MMG_DB_SERVICE_NAME##

export PGX_SERVER_URLS=##PGX_SERVER_URLS##

#### PGX data memory limits configurations ##
Overall Configuration

export MAX_TOTAL_SHARED_DATA_MEMORY_SIZE=##MAX_TOTAL_SHARED_DATA_MEMORY_SIZE##
```

```
export
MAX_TOTAL_PRIVATE_DATA_MEMORY_SIZE=##MAX_TOTAL_PRIVATE_DATA_MEMORY_SIZE##

export MAX_PER_SESSION_DATA_MEMORY_SIZE=##MAX_PER_SESSION_DATA_MEMORY_SIZE##
## Role wise data memory limits

export MAX_DATA_MEMORY_SIZE_DSUSRGRP=##MAX_DATA_MEMORY_SIZE_DSUSRGRP##

export MAX_DATA_MEMORY_SIZE_DSBATCH=##MAX_DATA_MEMORY_SIZE_DSBATCH##

export MAX_DATA_MEMORY_SIZE_DSINTER=##MAX_DATA_MEMORY_SIZE_DSINTER##

export MAX_DATA_MEMORY_SIZE_DSAPPROVER=##MAX_DATA_MEMORY_SIZE_DSAPPROVER##

export MAX_DATA_MEMORY_SIZE_DSUSER=##MAX_DATA_MEMORY_SIZE_DSUSER##

export MAX_DATA_MEMORY_SIZE_IHUSRGRP=##MAX_DATA_MEMORY_SIZE_IHUSRGRP##

export GRAPH_SERVICE_CACHE_SERVER_PORT=##GRAPH_SERVICE_CACHE_SERVER_PORT##

#end of Properties configurations for L2G

##Schema details for graph service. This is configured as a temporary/target
space for DP to create target tables which will act as input to L2G

export GRAPH_SCHEMA_WALLET_ALIAS=##GRAPH_SCHEMA_WALLET_ALIAS##

export GRAPH_SCHEMA_DB_SCHEMA_NAME=##GRAPH_SCHEMA_DB_SCHEMA_NAME##

#Additional MMG Features
export
MMG_MODEL_PIPELINE_SANDBOX_DEFAULT_VIEW=##MMG_MODEL_PIPELINE_SANDBOX_DEFAULT_V
IEW##

## The following properties are optional and enabled by default. If needed,
you can uncomment them and set them to false.
#export MMG_HTTP2_ENABLED=##MMG_HTTP2_ENABLED##

#export MMG_SERVER_ACCESS_LOG_ENABLED=##MMG_SERVER_ACCESS_LOG_ENABLED##

## The following properties are optional and disabled by default. If needed,
you can uncomment them and set them to true.
#export OJET_CDN_ENABLED=##OJET_CDN_ENABLED##

##The following Properties are related to EST
export EST_ENABLED=##EST_ENABLED##

export EST_UI_URL=##EST_UI_URL##

export EST_SERVICE_URL=##EST_SERVICE_URL##

export DATACATALOG_SERVICE_URL=##DATACATALOG_SERVICE_URL##

##DEFAULT CONNECTION POOLING CONFIGURATION FOR EXTERNAL SCHEMA (Time in
ISO-8601 format)
export GRAPH_EXT_SCHEMA_ENABLE_CP=##GRAPH_EXT_SCHEMA_ENABLE_CP##

export GRAPH_EXT_SCHEMA_CP_MAX_IDLE=##GRAPH_EXT_SCHEMA_CP_MAX_IDLE##

export GRAPH_EXT_SCHEMA_CP_MIN_IDLE=##GRAPH_EXT_SCHEMA_CP_MIN_IDLE##

export GRAPH_EXT_SCHEMA_CP_INITIAL_SIZE=##GRAPH_EXT_SCHEMA_CP_INITIAL_SIZE##

export GRAPH_EXT_SCHEMA_CP_MAX_TOTAL=##GRAPH_EXT_SCHEMA_CP_MAX_TOTAL##

export
GRAPH_EXT_SCHEMA_CP_MAX_WAIT_MILLIS=##GRAPH_EXT_SCHEMA_CP_MAX_WAIT_MILLIS##
```

```
export
GRAPH_EXT_SCHEMA_CP_MIN_EVICTABLE_IDLE_TIME=##GRAPH_EXT_SCHEMA_CP_MIN_EVICTABL
E_IDLE_TIME##

export
GRAPH_EXT_SCHEMA_CP_SOFT_MIN_EVICTABLE_IDLE_TIME=##GRAPH_EXT_SCHEMA_CP_SOFT_MI
N_EVICTABLE_IDLE_TIME##

export GRAPH_DOCUMENT_DOWNLOAD_PATH=##GRAPH_DOCUMENT_DOWNLOAD_PATH##

##Proxy Configurations
export GRAPH_NO_PROXY=##GRAPH_NO_PROXY##

##Data Studio Ports
## Following are the default ports 7008, 7009, 7012,-1. If needed, you can
change the port numbers other than the default values.

export DATASTUDIO_SERVER_PORT=##DATASTUDIO_SERVER_PORT##

export
DATASTUDIO_MARKDOWN_INTERPRETER_PORT=##DATASTUDIO_MARKDOWN_INTERPRETER_PORT##

export
DATASTUDIO_PYTHON_INTERPRETER_PORT=##DATASTUDIO_PYTHON_INTERPRETER_PORT##

export DATASTUDIO_R_INTERPRETER_PORT=##DATASTUDIO_R_INTERPRETER_PORT##

export DATASTUDIO_JDBC_INTERPRETER_PORT=##DATASTUDIO_JDBC_INTERPRETER_PORT##

export
DATASTUDIO_PYTHON_INTERPRETER_REST_SERVER_PORT=##DATASTUDIO_PYTHON_INTERPRETER
_REST_SERVER_PORT##

DATASTUDIO_PGX_PYTHON_INTERPRETER_REST_SERVER_PORT=##DATASTUDIO_PGX_PYTHON_INT
ERPRETER_REST_SERVER_PORT##

export
DATASTUDIO_THRIFT_EVENT_HANDLER_PORT=##DATASTUDIO_THRIFT_EVENT_HANDLER_PORT##

export DATASTUDIO_PGX_INTERPRETER_PORT=##DATASTUDIO_PGX_INTERPRETER_PORT##

export MMG_COHERENCE_CLUSTER_PORT=##MMG_COHERENCE_CLUSTER_PORT##

##MMG Gateway Configuration
export MMG_GATEWAY_PORT=##MMG_GATEWAY_PORT##

The following property can be set to control the pages where MMG can be
embedded:
# Set to 'self' to allow embedding only from the same origin (recommended for
most setups).

# Set to 'all' or '*' to allow embedding from any origin. (less secure)

# Set to a comma-separated list of origins to allow embedding from those
specified origins and from the same origin.

# By default, this is set to 'self'. export
MMG_CSP_FRAME_ANCESTORS=##MMG_CSP_FRAME_ANCESTORS##

# If the Gateway is enabled, this property can be set to control the pages
where Data Studio can be embedded:
# Set to '*' to allow embedding from any origin (less secure).

# Set to a comma-separated list of origins to allow embedding from those
specified origins and from the same origin.
```

```
# By default, this is set to MMG Gateway URL.

# If a load balancer or an external gateway is configured for MMG Gateway, the
URL must be included in the list of origins.

export DATASTUDIO_CSP_FRAME_ANCESTORS=##DATASTUDIO_CSP_FRAME_ANCESTORS##

##MMG Model Serving Configuration
export
MMG_MODEL_ENDPOINT_RESTART_INTERVAL=##MMG_MODEL_ENDPOINT_RESTART_INTERVAL##

##Enhanced Configurability for Data Studio Performance and Session Parameters

# Cleanup timeout for Jobs

export
DATASTUDIO_JOBS_INCOMPLETE_TIMEOUT=##DATASTUDIO_JOBS_INCOMPLETE_TIMEOUT##

export
DATASTUDIO_JOBS_INCOMPLETE_SCHEDULED_CLEANUP_DELAY=##DATASTUDIO_JOBS_INCOMPLET
E_SCHEDULED_CLEANUP_DELAY##

# Studio session timeouts

export
DATASTUDIO_SECURITY_SESSION_TIMEOUT_MS=##DATASTUDIO_SECURITY_SESSION_TIMEOUT_M
S##

export
DATASTUDIO_SECURITY_ABSOLUTE_SESSION_TIMEOUT_MS=##DATASTUDIO_SECURITY_ABSOLUTE
_SESSION_TIMEOUT_MS##

export
DATASTUDIO_INTERPRETER_IDLE_SESSION_TIMEOUT=##DATASTUDIO_INTERPRETER_IDLE_SESS
ION_TIMEOUT##

# Interpreter idle session timeout and cleanup

export
DATASTUDIO_INTERPRETER_IDLE_SESSION_TIMEOUT=##DATASTUDIO_INTERPRETER_IDLE_SESS
ION_TIMEOUT##

export
DATASTUDIO_INTERPRETER_CLEANUP_ENABLED=##DATASTUDIO_INTERPRETER_CLEANUP_ENABLE
D##

export
DATASTUDIO_INTERPRETER_CLEANUP_CRON="##DATASTUDIO_INTERPRETER_CLEANUP_CRON##"

export
DATASTUDIO_INTERPRETER_EXECUTION_RUN_TIMEOUT=##DATASTUDIO_INTERPRETER_EXECUTIO
N_RUN_TIMEOUT##

# Studio Server Thread Pool Size

export
DATASTUDIO_SERVER_ASYNC_THREADPOOL_SIZE=##DATASTUDIO_SERVER_ASYNC_THREADPOOL_S
IZE##

export
DATASTUDIO_SERVER_SCHED_THREADPOOL_SIZE=##DATASTUDIO_SERVER_SCHED_THREADPOOL_S
IZE##

# Hikari Pool Configurations
```

```
export DATASTUDIO_HIKARI_MAXPOOLSIZE=##DATASTUDIO_HIKARI_MAXPOOLSIZE##

export DATASTUDIO_HIKARI_CONN_TIMEOUT_MS=##DATASTUDIO_HIKARI_CONN_TIMEOUT_MS##

export DATASTUDIO_HIKARI_MINIMUM_IDLE=##DATASTUDIO_HIKARI_MINIMUM_IDLE##

export DATASTUDIO_HIKARI_MAX_LIFE_TIME=##DATASTUDIO_HIKARI_MAX_LIFE_TIME##

# Tomcat Threads

export
DATASTUDIO_SERVER_TOMCAT_THREADS_MAX=##DATASTUDIO_SERVER_TOMCAT_THREADS_MAX##

# Zeppelin Scheduler Threadpool Size
```

> **Note**
>
> Ensure that DATASTUDIO_ZPLN_SCHED_THREADPOOL_SIZE is greater than DATASTUDIO_SERVER_TOMCAT_THREADS_MAX+DATASTUDIO_SERVER_ASYNC_THREADPOOL_SIZE

```
export
DATASTUDIO_ZPLN_SCHED_THREADPOOL_SIZE=##DATASTUDIO_ZPLN_SCHED_THREADPOOL_SIZE#
##
```

> **Note**
>
> Gateway is enabled by default and gateway port is mandatory to access gateway enabled UI.

```
#ENDOFFILE#
```

Table 5-1 config.sh file

Parameter	Description	Is Mandatory	Comments
##APPLICATION_NAME##	Title of the application; if not replaced, default is : Model Management and Governance.	YES	It defaults to "Model Management and Governance." **Note:** Provide double quotes for the application name if it is long or contains spaces.
##WALLET_LOCATION##	The wallet is the folder containing the sqlnet.ora, wallet.sso, and .p12 files.	YES	/scratch/users/wallet
##TNS_ADMIN_PATH##	The folder that contains the tnsnames.ora file.	YES	/scratch/users/tns
##WALLET_ALIAS##	The wallet alias name configured for the MMG application config schema.	YES	MMG CONFIG

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##LOG_HOME##	A writable folder designated for storing application and MMG Studio logs.	YES	/scratch/users/logs **Note**: Ensure that log folder is created before installation.
##FTPSHARE##	This can be any writable folder accessible to the process owner.	YES	/scratch/users/ftpshare Ensure that ftpshare folder is created before installation. This should be same as the metadata directory mentioned above.
##LOG_TIMEZONE##	Specifies the timezone used for displaying log timestamps	YES	GMT+05:30, GMT-04:00, Asia/Kolkata
##DATASTUDIO_URL##	URL for MMG Studio.	YES	By default, Data Studio is assumed to run on the same server. If it is running on a different host, uncomment the line below and provide the appropriate URL. https://<hostname/IP>:7008/<contextpath> **NOTE:** The default port for MMG Studio is 7008 and should not be modified.
##BE_HOSTNAME##	Hostname on which the backend service (mmg-service) runs. Use the same hostname wherever applicable.	YES	HostIP or FQDN
##BE_PORT##	Port on which the backend service (mmg-service) needs to run.	YES	7002
##SMTP_HOST##	Hostname of the SMTP server used for email notification service. This value will be populated in the AAICL_SC_COMPONENT_DETAILS table.	YES	Defaults to localhost. Provide SMTP server hostname is external mail server is used.
##SMTP_PORT##	Port number for the SMTP server used for email notification service. This value will be populated in the AAICL_SC_COMPONENT_DETAILS table.	YES	Defaults to 25. Update this if your SMTP service listens on a different port.

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##UI_PORT##	Port on which UI service (mmg-ui) needs to run.	YES	7001
##SCHEMA_PORT##	Port on which Schema Creator service needs to run.	YES	7003
##CONTEXT_PATH##	Context path of the application.	YES	MMG
##STUDIO_AUTH_TYPE##	• FCC_SSO – for SAMLRealm based authentication in FCC Studio • MMG_AAI- AAI based authentication for MMG Studio	YES	Can be either MMG_AAI or FCC_SSO
##SSO_TOKEN##	SSO Token value for Studio authentication. Applicable only when STUDIO_AUTH_TYPE is FCC_SSO and MMG_AAI. For FCC_SSO, refer to the Oracle Financial Services Compliance Studio Installation Guide.	YES	**Note**: Token generation is now automated and stored into a separate file in `"install-helper/tmp/sso-token.log"`.
##SSL_ENABLED##	This enables HTTPS.	YES	Example: `true`
##SSL_KEYSTORE##	Absolute path for the keystore file. **Note**: Run the following command to create a keystore: keytool -genkey -v -alias demoalias -keyalg RSA -keysize 2048 -keystore server.keystore -validity 3650 -keypass secret -storepass secret -storetype PKCS12	YES	../conf/server.keystore. Include the file name in the path. **NOTE:** If ##SSL_ENABLED## is set to false, you must configure keystore for mmg-studio, as it is SSL-enabled by default. MMG application and MMG Studio can share the same SSL configuration if set up on the same server.
##SSL_KS_SECRET##	Keystore secret The value passed in the aforementioned command for -keypass	YES	Example: secret

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##SSL_KS_TYPE##	Keystore type The value passed in the aforementioned command for -storetype Can be either JKS or PKCS12	YES	Example: PKCS12
##SSL_KS_ALIAS##	Keystore alias The value passed in the aforementioned command for -alias	YES	Example: demoalias
##SESSION_TOKEN_CREDENTIALS##	The password used to generate the Authorization header token to communicate with mmg-services.	YES	**NOTE:** If not applicable, enter NA
##FCC_API_USER##	API user for FCC Studio.	YES	**NOTE**: If not applicable, enter NA
##MMG_DATASOURCE_MAX_POOL_SIZE##	Maximum connection pool size allowed for Config Datasource.	YES	It defaults to 10. You can edit it if required.
##MMG_DATASOURCE_IDLE_TIMEOUT##	Idle timeout for config Datasource.	YES	It defaults to 30000. You can edit it if required.
##MMG_DATASOURCE_CONN_TIMEOUT##	Connection timeout for Config Datasource.	YES	It defaults to 80000. You can edit it if required.
##EXT_DATASOURCE_MAX_POOL_SIZE##	Maximum connection pool size allowed for meta/data schemas.	YES	It defaults to 10. You can edit it if required.
##EXT_DATASOURCE_IDLE_TIMEOUT##	Idle timeout for meta/data schemas.	YES	It defaults to 30000. You can edit it if required.
##EXT_DATASOURCE_CONN_TIMEOUT##	Connection timeout for meta/data schemas.	YES	It defaults to 80000. You can edit it if required.
##MMG_HTTP_MAX_CONN##	The maximum number of connections allowed across all routes.	YES	It defaults to 20.
##MMG_HTTP_MAX_CONN_PER_ROUTE##	The maximum number of HTTP connections allowed for a route.	YES	It defaults to 2.
##MMG_HTTP_CONNECT_TIMEOUT##	The connection timeout for HTTP connection. A timeout value of 0 specifies an infinite timeout.	YES	It defaults to 30000.

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##MMG_HTTP_READ_TIMEOUT##	The socket read timeout for HTTP connection. A timeout value of 0 specifies an infinite timeout.	YES	It defaults to 120000.
##APPLICATION_ID##	The id will be stored as app_id and must be the same as mentioned in the APP_ID column of MMG_PATCHES table. Currently the UI displays the MMG Version <version number of application> and last applied MMG version.	YES	The APPLICATION_ID should be without spaces.
##APPLICATION_FAVICON_PATH##	Icon for the application. If not specified, it will default to the icon at the following location: `css/images/favicon.ico`	NO	`css/images/favicon.ico`
##UI_AUTH_TYPE##	aai – if using an existing AAI instance as the identity provider. saml – for saml based authentication ldap – for ldap based authentication **NOTE:** This is case sensitive.	YES	Can be one of the following: aai, or saml or ldap.
##AAI_AUTH_URL##	Base URL of the AAI instance. Will be used for ##UI_AUTH_TYPE## = aai **Note:** If the target AAI is https, then it is necessary to import the AAI host certificate into the MMG server Java keystore.	YES	http(s)://whfxxxxx.in.oracle.com:7110/mmg
##AUTH_SAML_REQUEST_TYPE##	This can be set to REDIRECT (default) / POST.	NA	NA
##AUTH_SAML_INCLUDE_SP_CERT##	This can be set to True/ False (default).	NA	NA

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##SAML_IDP_URL##	This is the endpoint on the IDP side where SAML requests are posted. The Service Provider (SP) needs to obtain this information from the Identity Provider (IDP).	YES	http(s)://idcs-xxxx.com/fed/v1/idp/sso This is used only if ##UI_AUTH_TYPE## is SAML.
##SAML_SP_ENTITY##	Enter a globally unique name for SAML entity. It typically takes the URL of an identity provider or a service provider as a value.	YES	http(s)://<UI_HOST>:<UI_PORT>/mmg This is used only if ##UI_AUTH_TYPE## is SAML.
##SAML_SRV_URL##	UI Landing Page URL.	YES	http(s)://<UI_HOST>:<UI_PORT>/mmg/home This is used only if ##UI_AUTH_TYPE## is SAML.
##SAML_LOGOUT_URL##	Initiated SAML Single Logout URL.	YES	http(s)://idcs-xxxx.com/sso/v1/user/logout This is used only if ##UI_AUTH_TYPE## is SAML.
##LDAP_URL##	LDAP URL Will be used for ##UI_AUTH_TYPE## = LDAP	YES	ldap://whf00xyz:3060/
##LDAP_SEARCH_BASE##	LDAP Search Base Will be used for ##UI_AUTH_TYPE## = LDAP	YES	"cn=Users,dc=oracle,dc=com"
##LDAP_USER_FILTER##	LDAP User Filter Will be used for ##UI_AUTH_TYPE## = LDAP	YES	"cn={0}"
#LDAP_USER_SEARCH_FILTER##	LDAP User Search Filter Will be used for ##UI_AUTH_TYPE## = LDAP	YES	NA
##LDAP_GROUP_SEARCH_FILTER##	LDAP Group Search Filter Will be used for ##UI_AUTH_TYPE## = LDAP	YES	NA

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##LDAP_GROUP_SEARCH_BASE##	LDAP Group Search Base Will be used for ##UI_AUTH_TYPE## = LDAP	YES	NA
##LDAP_GROUP_MEMBER#	LDAP Group Member Will be used for ##UI_AUTH_TYPE## = LDAP	YES	NA
##SERVER_COOKIE_DOMAIN##	The domain name.	YES	This should be the domain name of the host server. Example: .in.xyz.com **Note**: If the MMG application is configured with the IP address, then provide the same.
##SERVER_COOKIE_NAME##	The name for the cookie.	YES	If not set it will default to ORA_OLDS_SESSION
##SERVER_COOKIE_TIMEOUT##	Timeout/expiry duration in seconds.	YES	If not set, it defaults to 999999
##SERVER_COOKIE_IS_SECURE##	Specifies if we are using cookies to add an additional security layer to prevent cross-origin requests. Can be either **true** or **false**	YES	If not set, it defaults to true.
##AAI_AUTHZ_ENABLED##	This is disabled and is set to *False* by default. Set AAI_AUTHZ_ENABLED to *True* to enable.	YES	
##AAI_CLIENT_ID##	Create this in the OFSAA application in the location where you had configured the AAI Auth URL. a. Login as sysadmn in OFSAA b. From **System Configuration**, navigate to **Configure Instance Access Token** c. Add a new access token.	YES	

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##AAI_CLIENT_SECRET##		YES	
##SERVER_COOKIE_DOMAIN##		YES	
##SERVER_COOKIE_NAME##		YES	
##SERVER_COOKIE_TIMEOUT##		YES	
##SERVER_COOKIE_IS_SECURE##			
##BE_AUTH_TYPE##	Auth Type on which the backend service (mmg-service) runs.	YES	It defaults to public.
##MMG_PYTHON_INTERPRETER##	A comma separated value without whitespaces that specifies python interpreter python,fcc-ml4aml	YES	If not set, it defaults to python.
##DATACATALOG_SERVICE_URL##	Only used when EST application is integrated with MMG.	NO	NA
##STUDIO_WALLET_ENABLED##	Set as true when using a wallet for the MMG Studio Schema. Can be either **true/TRUE** or **false/FALSE (all caps or all small)**	YES	true/TRUE
##LOGIN_SHOW##	Can be either **true/TRUE** or **false/FALSE (all caps or all small)** **Note:** Set as true when the login screen of Studio is required. This property should be set as true if MMG application is non-SSL.	YES	It defaults to true.
##SESSION_MODE##	Can be either NOTEBOOK or NOTEBOOK_USER.	YES	If not set, it defaults to NOTEBOOK.
##STUDIO_REALM##	Can be either OFSAARealm or saml.OFSAASamlRealm	YES	OFSAARealm – the default realm for studio auth type FCC_AAI, MMG_AAI. .auth.saml.OFSAASamlRealm – for SAML specific studio authentication

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##OFSAA_URL##	AAI login IDM Service URL. This is applicable only if ## STUDIO_AUTH_TYPE ## is " MMG_AAI ".	YES	**Format**: http://<ofsa-web-host>:<port>/<context>/rest-api For example, http://ABC00abc:4325/LLFP/rest-api The /rest-api is mandatory for OFSAA URL.
##API_USERS##	This is the API user with which the token is generated; if not set, it defaults to MMG_API_USER. **Note**: Use the same <API_USER> as given in the ##SSO_TOKEN##	YES	MMG_API_USER
##VALID_ROLES##	MDLUSR,MDLREV,MDLAPPR The comma separated values for Studio-related roles in USER-ROLE mapping.	YES	MDLBATCHUSR, DSUSRGRP, DSREDACTGRP
##DATASOURCE_URL##	The connection address to the database where the MMG Studio Schema is created. When ##WALLET_ENABLED## is false- jdbc:oracle:thin:@<Host>:<Port>/<Service_Name> When ##WALLET_ENABLED## is true- jdbc:oracle:thin:@<DS ALIAS> where <DS_ALIAS> is the wallet alias configured for the MMG Studio Schema.	YES	NA
##DATASOURCE_USERNAME##	MMG Studio Schema/ User name; required only when ##WALLET_ENABLED## is false	YES	dsschema

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##DATASOURCE_PASSWORD##	MMG Studio Schema/ User Password; required only when ##WALLET_ENABLED## is false	YES	password
##DATASOURCE_DRIVER##	Database Driver used in connection	YES	oracle.jdbc.OracleDriver
##JPA_DB_PLATFORM##	Hibernate Class or SQL Dialect used in Database	YES	org.hibernate.dialect.Oracle12cDialect
##STUDIO_LOG_LEVEL##	Logging level for logs.	YES	info, warn, debug or error logs
##PYTHON_HOME##	Home Path of Python Library. It defaults to python3 during installation. For a custom installation of python3 where the soft link is not configured, you can mention the complete path up to python3.	YES	python3
##SPARK_HOME##	Absolute path of Apache Spark Library.	NO	NA
##R_ENABLED##	This can be set to TRUE/true or FALSE/ false depending on which R interpreter will be started and will be present in the interpreters list. **Note**: If you are using an older Studio schema with an R-interpreter already present and then install with **R_ENABLED** set as FALSE; the R-interpreter will remain in the interpreter's menu of Studio and must be deleted from there.	YES	The default is False.
##RS_CONF_PATH##	Absolute path to Rserve.conf file for running Rserve.	YES	/scratch/users/ datastudio/conf/ Rserve.conf
##RS_KEYSTORE##	Absolute path for the Keystore file made for Rserve.conf.	YES	/scratch/users/ datastudio/conf/ rinterpreterkeystore
##RS_KS_SECRET##	Keypass for rinterpreterkeystore.	YES	Example: changeit

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##TEMPLATE_CONFIG_PATH##	Configuration path of the Template.	NO	NA
##TEMPLATE_DEFAULT_LINK##	Default link of the template.	NO	NA
##AUTH_SERVICE_URL##	The AUTH service URL that is activated after the fccstudio.sh file runs.	NO	Example: https://<hostname>:7041/authservice
##META_SERVICE_URL##	The metaservice URL that is activated after the fccstudio.sh file runs.	NO	Example: https://<hostname>:7045/metaservice
##ER_SERVICE_URL##	Used for the entity resolution service.	NO	Example: https://<hostname>:<port>
##BATCH_SERVICE_URL##	Used for the batch service.	NO	Example: https://<hostname>:<port>/batchservice
##SAML_ISSUER##	The SAML entity ID (Studio URL) configured in the IDP.	YES	https://<hostname>.xyz.com:7008
##SAML_DESTINATION##	The SAML IDP URL that the Identity Provider provides after creating the SAML application.	YES	https://idcs-xyzgvh.com/fed/v1/idp/sso
##SAML_ASSERTION##	The SAML Consume URL (Studio/URL/saml/consume) that is configured in IDP.	YES	https://<hostname>.xyz.com:7008/saml/consume
##SAML_ROLE_ATTRIBUTE##	The SAML client identifier provided by the SAML Administrator for the role and attributes information while creating the SAML application for MMG Studio. The attribute will contain the role required for the application.	YES	Example: group
##SAML_STUDIO_LOGOUT_URL##	The SAML client identifier provided by the SAML Administrator for the Logout URL information, while creating the SAML application for MMG Studio.	YES	https://idcs-xyzgvh.com/sso/v1/user/logout
##SAML_COOKIE_DOMAIN##	Domain of the server.	YES	Example: in.xyz.com

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
Pipeline Services Specific Configurations			
##DATAPIPELINE_SERVICE_PORT1##	The port where the pipeline service resides.	YES	By default, it is set as 18005.
##DATAPIPELINE_SERVICE_PORT2##	The port where the data pipeline service resides.	YES	By default, it is set as 18006.
##DATA PIPELINE_METADATA_ARCHIVE_PATH##	The dump path for the pipeline service.	YES	/OFS_MMG/mmg-pipeline/pipeline/pipeline-service-x.x.x.x.x.
##DATA PIPELINE_METADATA_IMPORT_SERVICE_PORT##	Meta data import service port.	YES	By default, it is set as 18007.
##DATAPIPELINE_ERXMLPATH##	The XML path in which the schema details are stored.	NO	NA
##DATAPIPELINE_GATEWAY_SERVICE_PORT##	Data pipeline gateway service port.	YES	NA
##PIPELINE_UI_SERVICE_PORT##	Pipeline UI service port.	YES	NA
##DATA_PIPELINE_UI_SERVICE_PORT##	Data pipeline UI service port	YES	NA
##MATCHRULE_BASE_URL##	The host and port where the match rule service resides.	NO	http(s)://abc.in.xyz.com:7051
##LOADGRAPH_BASE_URL##	The host and port where the load graph service resides.	YES	http(s)://abc.in.xyz.com:7059/graph-service
##MATCHSRVC_UI_URL##	Matching Service UI resource path.	NO	NA

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##GRAPH_INDEX_BASE_URL##	Indicates the Graph Index resource path.	NO	NA http(s)://<hostname>xyz.com:7053/load-to-elastic-search
##LOADINDEX_UI_URL##	Indicates the Graph Load Index UI resource path.	NO	NA

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##MATCHING_MECHANISM##	Indicates the matching mechanism for Entity Resolution and Graph Enter the value that is used for matching mechanism for Entity Resolution and Graph. The options are OS, and OT. Where OS refers to OpenSearch 2.7.0, and OT refers to Candidate Selection with Oracle Text matching. By default, the value of the parameter is OT. For MATCHING_MECHANISM = OT, no additional installation or configuration is required as it is part of the Oracle Database. **NOTE:** This is applicable for Compliance Studio.	NO	NA
##CANDIDATE_SELECTION_SERVICE_URL##	Enable this service for Entity Resolution and Graph use cases when MATCHING_MECHANISM is set to OT. **NOTE:** This is applicable for Compliance Studio.	NO	NA
##LOAD_TO_OS_URL##	Provide the URL for Entity Resolution and Graph use cases when MATCHING_MECHANISM is set to OT. **NOTE:** This is applicable for Compliance Studio.	NO	NA
##AAI_COOKIE_DOMAIN##	The domain of the server.	YES	Example: in.xyz.com
##MMG_KEYS_LOC##	Indicates public and private key location.	YES	Example:< MMG Installation Path> / OFS_MMG/conf
##GRAPH_INSTALLATION_PATH##	The installation path of the Graph.	YES	<MMG Installation Path>/ OFS_MMG/mmg-load-to-graph/graph-service
##GRAPH_KEYSTORE_PASSWORD##	Graph Keystore Password.	YES	Password

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##GRAPH_SERVICE_PORT##	Graph Service Port	YES	By default, it is set as 7059. You should not modify the Graph Service Port if graphs are already created and executed.
##GRAPH_HIKARI_CP_MIN_IDLE##	Defines the minimum number of idle connections that HikariCP maintains in the pool.	YES	
##GRAPH_HIKARI_CP_MAX_POOL_SIZE##	Specifies the maximum number of total connections (active + idle) allowed in the pool.	YES	
##GRAPH_HIKARI_CP_IDLE_TIMEOUT##	Sets the maximum time a connection can remain idle before being removed from the pool.	YES	
##GRAPH_HIKARI_CP_MAX_LIFETIME##	Determines the maximum lifetime of a connection before it is retired and replaced.	YES	
##GRAPH_HIKARI_CP_TIMEOUT##	Defines how long HikariCP waits for a connection from the pool before timing out.	YES	
##MMG_DB_SERVER_NAME##	Name of the MMG Database Server.	YES	NA
##MMG_DB_PORT##	The port of the MMG database server.	YES	NA
##MMG_DB_SERVICE_NAME##	Name of the MMG Database Service.	YES	NA
##PGX_SERVER_URLS##	Indicates the pgx server resource path. Skip this if not installing pgx.	YES	http(s)://<hostname>.xyz.com:<pgx port>/<pgx context name>
##MAX_TOTAL_SHARED_DATA_MEMORY_SIZE##	Maximum total shared data memory size.	YES	Edit if required; default value is 20 GB.
##MAX_TOTAL_PRIVATE_DATA_MEMORY_SIZE##	Maximum total private data memory size.	YES	Edit if required; default value is 8 GB.
##MAX_PER_SESSION_DATA_MEMORY_SIZE##	Maximum per session data memory size.	YES	Edit if required; default value is 700 MB.
##MAX_DATA_MEMORY_SIZE_DSUSRGRP##	Maximum data memory size allowed for DSUSRGRP.	YES	Edit if required; default value is 10 GB.

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##MAX_DATA_MEMORY_SIZE_DSBATCH##	Maximum data memory size allowed for DSBATCH.	YES	Edit if required; default value is 10 GB.
##MAX_DATA_MEMORY_SIZE_DSINTER##	Maximum data memory size allowed for DSINTER.	YES	Edit if required; default value is 5 GB.
##MAX_DATA_MEMORY_SIZE_DSAPPROVER##	Maximum data memory size allowed for DSAPPROVER.	YES	Edit if required; default value is 5 GB.
##MAX_DATA_MEMORY_SIZE_DSUSER##	Maximum data memory size allowed for DSUSER.	YES	Edit if required; default value is 5 GB.
##GRAPH_SCHEMA_WALLET_ALIAS##	Wallet alias created for the Graph Schema.	YES	NA
##GRAPH_SCHEMA_DB_SCHEMA_NAME##	Name of the Graph schema.	YES	NA
##EST_ENABLED##	Only used when EST application is integrated with MMG.	YES	The default value is FALSE.
##EST_UI_URL##	The URL of EST application. This is set based on ##EST_ENABLED## property.	NO	NA
##DATASTUDIO_SERVER_PORT##	The port of the Data Studio server.	NO	The default value is 7008.
##DATASTUDIO_MARKDOWN_INTERPRETER_PORT##	The port of the Data Studio Markdown Interpreter.	NO	The default values are 7009, 7029.
##DATASTUDIO_PYTHON_INTERPRETER_PORT##	The port of the Data Studio Python Interpreter.	NO	The default values are 7012, 7032, 6012.
##DATASTUDIO_JDBC_INTERPRETER_PORT##	The port of the Data Studio JDBC Interpreter.	NO	The default values are 7011, 7031.
##DATASTUDIO_PYTHON_INTERPRETER_REST_SERVER_PORT##	The port of the Data Studio Python Interpreter Rest server.	NO	The default value is 6012
##DATASTUDIO_PGX_PYTHON_INTERPRETER_REST_SERVER_PORT##	The port of the Data Studio PGX Python Interpreter Rest server.	NO	The default value is 6022
##DATASTUDIO_THRIFT_EVENT_HANDLER_PORT##	The port of the Data Studio Thrift Event handler.	NO	The default value is 8432
##DATASTUDIO_PGX_INTERPRETER_PORT##	The port of the Data Studio PGX Interpreter.	NO	The default value is 7022.

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##MMG_COHERENCE_CLUSTER_PORT##	This is the port where the Coherence Cluster for the Notification service runs. By default, this value is 7574.	YES	NA
MMG Gateway Configuration			
##MMG_GATEWAY_ENABLED##	Enables or disables the MMG Gateway. By default, this is set to 'self'. **Note** Gateway is enabled by default and gateway port is mandatory to access gateway enabled UI.	YES	NA
##MMG_GATEWAY_PORT##	Specifies the port on which the MMG Gateway will run.	YES	NA
##MMG_CSP_FRAME_ANCESTORS##	Configure this attribute to all or the AAI origin when MMG has to be embedded from AAI. By default, MMG pages cannot be embedded if Gateway is enabled. This is to prevent CLickjacking vulnerability.	YES	NA

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##DATASTUDIO_CSP_FRAME_ANCESTORS##	If the Gateway is enabled, this property can be set to control the pages where Data Studio can be embedded: • Set to '*' to allow embedding from any origin (less secure). • Set to a comma-separated list of origins to allow embedding from those specified origins and from the same origin. By default, this is set to MMG Gateway URL.	YES	NA
##SAML_SIGN_AUTHN_REQ##	NA	YES	NA
##SAML_PRIVATE_KEY_PATH##	NA	NA	/scratch/mmg8133/config/sp-privatekey.pem
##SAML_SP_X509_CERT_PATH##	NA	NA	/scratch/mmg8133/config/sp-certificate.cer
##SAML_SIGN_ALGORITHM##	NA	NA	##SAML_SIGN_ALGORITHM##
##AUTH_SAML_REQUEST_TYPE##	NA	NA	##AUTH_SAML_REQUEST_TYPE##
##AUTH_SAML_INCLUDE_SP_CERT##	NA	NA	##AUTH_SAML_INCLUDE_SP_CERT##
##GIT_ENV_ID##	This variable represents the environment in which your GIT operations or deployments are being performed. Common environment names include dev (development), QA (quality assurance), SIT (system integration testing), PROD (production), and so on. It helps you to understand with which branch you need to work.	NA	##GIT_ENV_ID## A new group GIT ADMIN is created which is mapped to GIT_ADV role which is mapped to below functions. Kindly map the above group to the user to access the required GIT functionality: • GITVIEW • GITPUSH • GITPULL
##GIT_USERNAME##	This is your GitHub username.	NA	##GIT_USERNAME##

Table 5-1 (Cont.) config.sh file

Parameter	Description	Is Mandatory	Comments
##GIT_PAT_SECRET##	This variable stores your GitHub Personal Access Token (PAT). A PAT is a secret token used for authenticating API requests or GIT operations over HTTPS instead of a password.	NA	##GIT_PAT_SECRET##
##GIT_PROXY_URL##	If your environment requires you to access external resources like GitHub through a proxy server, this variable defines the proxy server's hostname or IP Address.	NA	##GIT_PROXY_URL##
##GIT_PROXY_PORT##	This defines the port number on which the proxy server listens. It works together with GIT_PROXY_URL to route your Git traffic through the proxy.	NA	##GIT_PROXY_PORT##
##RESTRICT_UNMAPPED_DATASTORES_ACCESS##	Restriction of users access to Data Stores from a workspace for unmapped datasources: If the value is set as True, only the current workspace attached data stores will get connection in notebook sessions using get_conn(). This will only work if the session-mode in application.yml is set to NOTEBOOK_USER. Only True/False is allowed.	NA	The default value is False.

Note

If changes are done directly on UI and then restart is done then in that case overwrite will not be triggered so configuration would not change, it is only dependent on file system JSON).

Note

- In case of ##OFSAA_URL## and ##MMG_SVC_URL##, do not add any ending '/' in the URLs
- If pool size, connection timeout and idle timeout are not configured, then it will proceed with default Hikari Configurations.
- The default session timeout is 3600 seconds (60 mins). You can configure timeout using server.servlet.session.timeout property.
- If the AUTH type specified is AAI, make sure the AAI System has appropriate user groups mapped for the users. WKSPADMIN, IDNTYADMN, IDNTYAUTH need minimally to be present for a successful subsequent logins.
- The name for MMG Studio cookie is ORA_OLDS_SESSION.
- If the ##SSL_ENABLED## is set to false, keystore configuration must be done for mmg-studio as it is SSL enabled by default. MMG application and MMG Studio can use the same SSL configuration if configured in the same server.
- The wallet is same for all the MMG services including MMG Studio. So, if you want to use the MMG Studio with wallet configurations, then configure in the same wallet.
- If the MMG Studio is remotely configured, then the MMG Application Configuration Schema wallet alias and `tnsnames.ora` file entries need to be added to the MMG Studio configured wallet and `tnsnames.ora` file.
- If MMG application is Non SSL, set the below property to "false" in the `application.yml` file inside the MMG Studio and restart the services.

```
security:

cookies:

secure: false
```

Note

Setting both Timeouts

Data Studio user Login sessions to be logged out automatically after a set amount of time:

It is recommended to set both the relative and absolute session timeouts.

With the following properties, a user session will expire after one hour of inactivity or the maximum duration of 24 hours.

Studio-Server:

Security:

Absolute-session-timeout-ms: 86400000 // 24 hours

Session-timeout-ms: 3600000 // 1 hour

Paragraph Execution Timeout

Automatically cancel a paragraph execution after a set amount of time:

Studio-server:

Interpreter:

Execution:

Run-timeout: 24 hours

You can set the timeout to 0 to disable the timeout entirely. When the timeout is disabled, paragraphs will run either until they finish or are being manually cancelled.

With the following properties, a paragraph will be timed out after 24 hours.

Synchronous Run Timeout

Timeout for synchronous code running (this is used for pipeline executions in MMG):

Studio-server:

Rest:

Code:

Maximum Runtime Sync: 20 minutes

With the following configuration, the synchronous run will be timed out after 20 minutes.

Note

The flag `*IS_SELF_USR_GRP_AUTH_ALLOWED*` in NEXTGENEMF_CONFIG table is used to check if the Self Groups Approvers/Reviewers are allowed or not. If it is **'Y'**, the Requested User as well Approvers and Reviewers can belong to the same user group. Otherwise, **'Self User Group Authorization is not allowed'** is returned.

Support of Alphanumeric Name in SAML Entity ID for SAML Configuration

MMG Application: `mmg_ui/application.properties` file contains SAML entity `saml.auth.sp.entity` placeholder that contains value like `https://<FQDN of compliance server>:7001/cs`

However, the requirement is to have this parameter value to `amlcs8126` i.e. support of SAML Entity ID as alphanumeric name.

The expected value from SAML team:
`saml.auth.sp.entity="amlcs8126":`

Entity ID should not contain below special character as per ForgeRock MF Authentication (Policy sets in the UI: ForgeRock Identity Cloud docs).

The regex allowed is: `^[A-Za-z0-9,.\\/:@&?\\-_]+$`, and it should be passed in string.

Import Server Certificate to Java Keystore

You must import the server certificate (`.cer`) file to the Java keystore.

To import the server certificate, perform the following steps:

1. Create a `.cer` file from the `server.keystore`.

   ```
   keytool -export -alias <alias>-file <filename>.cer -keystore
   <path_to_Keystore>/server.keystore -storepass secret
   ```

   Example:

   ```
   keytool -export -alias demo_alias -file server.cer -keystore OFS_MMG/config/
   server.keystore -storepass secret
   ```

2. Import `.cer` file generated from the above step to java keystore.

   ```
   keytool -import -file "<path_to_Keystore>/<filename>.cer" -alias <alias>-
   keystore "<java_home>/lib/security/cacerts" -storepass "changeit"
   ```

> **Note**
>
> The above step should be performed by the Root user.

Run the MMG Installer

To run the MMG Installer, follow these steps:

1. Navigate to following path:

   Go to `<MMG_INSTALLATION_PATH>/bin` directory.

2. Run the following command:

   ```
   ./install.sh
   ```

> **Note**
>
> When ./install.sh command is triggered, pre-installation utility validates install configurations such as availability of ports, ftpshare/log folders, database connections, and so on.

This step will install the configurations and has to be executed only once per deployment. This will also bring up the Schema Creator Service in nohup mode.

A message similar to following means a successful startup:

<MMG_INSTALLATION_PATH>/OFS_MMG/bin>./install.sh

PIPELINE_HOME: <MMG_INSTALLATION_PATH>/OFS_MMG/mmg-pipeline/pipeline

<MMG_INSTALLATION_PATH>/OFS_MMG/mmg-pipeline/pipeline

PIPELINE_HOME: <MMG_INSTALLATION_PATH>/OFS_MMG/mmg-pipeline/pipeline

Installing Pipeline Data Model. Please Wait ...

Pipeline Data Model installation finished.

Starting Gateway ...

Starting Pipeline UI Service ...

Starting Pipeline Service ...

Starting Data Pipeline UI Service ...

Starting Data pipeline services ...

Inserting DataMeta Data ...

********** Data Pipeline Deployment Done **********

Stopping Graph-Service service...

Graph-Service stopped.

Schema Creator executed successfully for config schema

Schema Creator for config executed successfully.

If Graph Schema is configured, the below message is displayed.

Now triggering for graph-schema

./../mmg-schema-creator/bin/startup.sh: line 70: 126438 Killed nohup java -jar -Doracle.net.tns_admin=/scratch/ofsaadb -Doracle.net.wallet_location=/scratch/ofsaadb/wallet -Dspring.config.location=../conf/ -Dspring.datasource.url=jdbc:oracle:thin:@conf_als -Dspring.liquibase.change-log=file:../scripts/changelog-master.xml $JAVA_OPTS ../lib/mmg-schema-creator.war > nohup.out 2>&1

Schema Creator executed successfully for graph schema

nohup: ignoring input and redirecting stderr to stdout

Stopping Graph-Service service...

Graph-Service stopped.

nohup: ignoring input and redirecting stderr to stdout

You can check mmg-schema-creator/bin/nohup.out to check if the service comes up properly.
Started BuildSchemaCreatorApplication in 20.317 seconds (JVM running for 21.26)

> **Warning**
>
> If you notice any errors, do not proceed further. Contact My Oracle Support (MOS) and provide the applicable error code and log files.

3. Execute `shutdown.sh` and trigger `startup.sh` for the services to come up. For more details, refer to the below sections.

> **Note**
>
> The MMG Application is installed with or without OFSAA, depending on the configuration provided in the `config.sh` file.

Modify the Logging Level

The log rotation policy in the default OFS MMG setup is set to create a new log file whenever the size crosses 5MB and also to retain the last 5 log files. This feature is customizable. Additionally, the logging level can be modified to mmg-service, mmg-ui, mmg-studio etc.

The log config file for each service is present at the following locations. A backup can be taken and updates can be performed on the same:

- <MMG_INSTALLATION_PATH>/mmg-service/conf/log4j2.xml
- <MMG_INSTALLATION_PATH>/mmg-ui/conf/log4j2.xml
- <MMG_INSTALLATION_PATH>/mmg-studio/conf/log*.xml
- <MMG_INSTALLATION_PATH>/mmg-gateway/conf/log4j2.xml

Sample 1 (Size based rolling)

```
<Appenders>
 <RollingFile name="RollingFile">
  <FileName>##LOG_HOME##/services/mmg-service.log</FileName>
  <FilePattern>##LOG_HOME##/services/mmg-service.%i.log.zip</FilePattern>
  <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p [%c{1}] - %m%n"/>
  <Policies>
   <!-- New log once the current size crosses 10MB -->
   <SizeBasedTriggeringPolicy size="10MB"/>
  </Policies>
   <!-- keep last 20 log files -->
  <DefaultRolloverStrategy max="20"/>
 </RollingFile>
 <RollingFile name="HealthCheckFile">
  <FileName>##LOG_HOME##/services/mmg-health.log</FileName>
  <FilePattern>##LOG_HOME##/services/mmg-health.%i.log.zip</FilePattern>
  <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p - %m%n"/>
  <Policies>
   <!-- New log once the current size crosses 12MB -->
   <SizeBasedTriggeringPolicy size="12MB"/>
  </Policies>
  <!-- keep last 15 log files -->
  <DefaultRolloverStrategy max="15"/>
 </RollingFile>
 <Console name="Console" target="SYSTEM_OUT">
  <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p [%c{1}] - %m%n"/>
```

```
 </Console>
</Appenders>
```

Sample 2 (Day based logs rolling at midnight)

```
<Appenders>
 <RollingFile name="RollingFile"
              fileName="##LOG_HOME##/services/mmg-service.log"
              filePattern="##LOG_HOME##/services/mmg-service-%d{yyyy-MM-
dd}.log.gz">
 <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p [%c{1}] - %m%n"/>
 <Policies>
  <!-- rollover at midnight -->
  <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
 </Policies>
 <!-- keep last 10 daily logs -->
 <DefaultRolloverStrategy max="10"/>
 </RollingFile>
 <RollingFile name="HealthCheckFile"
              fileName="##LOG_HOME##/services/mmg-health.log"
              filePattern="##LOG_HOME##/services/mmg-health-%d{yyyy-MM-
dd}.log.gz">
  <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p - %m%n"/>
  <Policies>
  <!-- rollover at midnight -->
  <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
 </Policies>
  <!-- keep last 15 daily logs -->
  <DefaultRolloverStrategy max="15"/>
 </RollingFile>
 <Console name="Console" target="SYSTEM_OUT">
   <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p [%c{1}] - %m%n"/>
 </Console>
</Appenders>
```

Sample 3 (Hourly log files)

```
<Appenders>
  <RollingFile name="RollingFile"
               fileName="##LOG_HOME##/services/mmg-service.log"
               filePattern="##LOG_HOME##/services/mmg-service-%d{yyyy-MM-dd-
HH}.log.gz">
    <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p [%c{1}] - %m%n"/>
    <Policies>
      <!-- rollover every hour on the clock -->
      <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
    </Policies>
    <!-- keep last 10 hourly logs -->
    <DefaultRolloverStrategy max="10"/>
  </RollingFile>
  <RollingFile name="HealthCheckFile"
```

```
                fileName="##LOG_HOME##/services/mmg-health.log"
                filePattern="##LOG_HOME##/services/mmg-health-%d{yyyy-MM-dd-
HH}.log.gz">
    <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p - %m%n"/>
    <Policies>
      <!-- rollover every hour on the clock -->
      <TimeBasedTriggeringPolicy interval="1" modulate="true"/>
    </Policies>
    <!-- keep last 15 hourly logs -->
    <DefaultRolloverStrategy max="15"/>
  </RollingFile>
  <Console name="Console" target="SYSTEM_OUT">
    <PatternLayout pattern="%d{dd-MM-yyyy HH:mm:ss.SSS}{##LOG_TIMEZONE##}
[%18.18t] %-5p [%c{1}] - %m%n"/>
  </Console>
</Appenders>
```

Starting MMG Services

To start the MMG service, run the following command:

- Navigate to `<MMG_INSTALLATION_PATH>/bin` directory. `./startup.sh`

  A message similar to following means a successful startup:

  Starting MMG UI...

  MMG UI started successfully.

  Starting MMG Service...

  MMG Service started successfully.

  Starting Data Studio...

  Data Studio started successfully.

  Starting Gateway ...

  Starting Pipeline UI Service ...

  Starting Pipeline Service ...

  Starting Data Pipeline UI Service ...

  Starting Data pipeline services ...

  You may check `<MMG_INSTALLATION_PATH>/mmg-ui/bin/nohup.out` to check if the UI service comes up properly.

  A message similar to following means a successful startup:

  `Started BuildUIServiceApplication in 27.981 seconds (JVM running for 29.365)`

  You can check `<MMG_INSTALLATION_PATH>/mmg-service/bin/nohup.out` to check if the backend service comes up properly.

  A message similar to following means a successful startup:

  `Started BuildServiceBuildApplication in 20.317 seconds (JVM running for 21.26)`

  You can check `<MMG_INSTALLATION_PATH>/mmg-studio/bin/nohup.out` to check if the backend service comes up properly.

A message similar to following means a successful startup:

```
05:06:02.155 Thread-9] INFO oracle.datastudio.starter.App - Data Studio Server is ready to use
```

This will start the successful installation of application.

> **Warning**
>
> If you notice any errors, do not proceed further. Contact My Oracle Support (MOS) and provide the applicable error code and log files.

> **Note**
>
> Unset the https/http proxy details before starting the services.
> OR
>
> Add the relevant entries in no_proxy with mmg hosted server details.

Stopping MMG Services

To stop the MMG services, run the following command: `./shutdown.sh`

A message similar to following means a successful shutdown:

Stopping Graph-Service service...

Graph-Service stopped.

MMG UI shutdown is complete.

MMG Service shutdown is complete.

MMG Schema Creator shutdown is complete.

Data Studio shutdown is complete.

Data Pipeline Service shutdown is complete.

Generate GRAPH-KEYSTORE.P12

Graph services should be up and running.
To generate GRAPH-KEYSTORE.P12 file, perform the below steps:

> **Note**
>
> The Keystore generation fails if graph service is down.

1. Execute `graph-keystore-generator.sh` using PUTTY.
2. Enter the values as below when prompted.

   Enter Wallet Alias : <GRAPH_SCHEMA_WALLET_ALIAS> as given in the `config.sh` file.

   Enter Password: <GRAPH_SCHEMA_DB_SCHEMA> password

Enter Keystore alias: <GRAPH_SCHEMA_DB_SCHEMA_NAME> as given in the `config.sh` file.

Check the below location for the graph-keystore.p12

<mmg installation path>/OFS MMG/mmg-load-to-graph/graph-service/conf/

Installing Conda

Conda as a package manager helps you to find and install packages. With the capability of environment manager, you can set up a totally separate environment to run different versions of Python. In addition, you can continue to run your usual version of Python in your normal environment.

> **Note**
>
> The supported version is 4.14.0.

To install the Conda, perform the following:

1. Download the miniconda.
2. Copy it to your server where the Conda needs to be installed.
3. Grant execute permission to the Conda folder.
4. Execute the following command: `$ ./Miniconda3-latest-Linux-x86_64.sh`
5. Update the PATH variable with miniconda installation path:

   `<install_path>/miniconda3/bin`

> **Note**
>
> In the current release, the Conda feature is not supported in Solaris Operating System.

For more details on the Roles and privileges, see *MMG User Guide*.

Installing Python Library

This section provides detailed steps to install the Python Library.

Prerequisites

- Python 3.8.x and above

> **Note**
>
> Ensure the libraries, bzip2-devel, sqlite-devel, ncurses-devel, and xz-devel, libffi-devel are installed before you install the Python package.

For Example:

> **Note**
>
> Install the below libraries as a root user.
>
> - **bzip2-devel:** Execute the command `yum install bzip2-devel`
> - **sqlite-devel:** Install as a root user using the command `yum install sqlite-devel`
> - **ncurses-devel:** Install as a root user using the command `yum install ncurses-devel`
> - **xz-devel:** Install as a root user using the command `yum install xz-devel`
> - **libffi-devel:** Install as a root user using the command `yum install libffi-devel`

> **Note**
>
> In order to download the latest python packages after MMG upgrade, the pip cache in the server user home directory needs to be cleared.
> Path: `$HOME/.cache/pip`

> **Note**
>
> Documentation of Mandatory Parameter to be passed as part of Python environment install.sh Installation
> Now, parameters is supported as below for python-env-install.sh file:
>
> The user needs to be passed in case the packages has to be installed at a user level. The above is not required in case of CONDA environment. In this case make sure the Python3 is pointed to proper Conda environments.

Procedure

1. Set system python3 to the one that is to be used. Navigate to bin folder.
2. To install the mmg library with dependencies from `conf/requirements.txt`, execute the following command:

   ```
   ./python-env-install.sh
   ```

   The following parameter is supported for python-env-install.sh file:

   --user need to be passed in case the packages has to be installed at user level.

   This is not required in case of Conda environment but python3 must be configured to the appropriate Conda environments.

> **Note**
>
> Expose Python variables/functions to derive File Path when saving files directly from Notebook script (without using Save Widget).

3. To install the mmg library with flexible dependencies or using already installed dependent packages, execute the following command:

   ```
   ./python-env-install.sh -S
   ```

   OR

   ```
   ./python-env-install.sh --skip
   ```

   This will skip the installation of dependency based on the version mentioned in the conf/requirements.txt. The installation will be with whatever version available in the pypi server.

4. To install the Apache Flink packages, execute the following command:

   ```
   ./python-env-install.sh --include-flink
   ```

Note

Ignore the below error message during Apache Flink package installation.

```
ERROR: pip's dependency resolver does not currently take into account
all the

packages that are installed. This behavior is the source of the
following dependency conflicts.

modin 0.19.0 requires pandas==1.5.3, but you have pandas 1.3.5 which
is incompatible.

Successfully installed numpy-1.21.4 pandas-1.3.5 python-dateutil-2.8.0

Installing with dependencies

ERROR: pip's dependency resolver does not currently take into account
all the

packages that are installed. This behavior is the source of the
following dependency conflicts.

pemja 0.2.6 requires numpy==1.21.4, but you have numpy 1.24.2 which is
incompatible.

apache-flink 1.16.1 requires numpy<1.22.0,>=1.21.4;
python_full_version >=

"3.7", but you have numpy 1.24.2 which is incompatible.

apache-flink 1.16.1 requires pandas<1.4.0,>=1.3.0; python_full_version
>=

"3.7", but you have pandas 1.5.3 which is incompatible.

apache-flink 1.16.1 requires python-dateutil==2.8.0, but you have

python-dateutil 2.8.2 which is incompatible.

apache-beam 2.38.0 requires

numpy<1.23.0,>=1.14.3, but you have numpy

1.24.2 which is incompatible.
```

Setting up the Environment for Hive Data Sourcing

This section is applicable if you want to use Hive Data Source.

In the MMG Home directory, a lib folder is available for the Hive specific jars and a conf folder is available for the Kerberos Configuration and Keytab files.

Hive Source Connection Requirements:

MMG_HOME/conf: kbank.keytab and krb5.conf files

MMG_HOME/lib: hive-jdbc-uber-2.6.3.0-235.jar

MMG-Studio/conf: kbank.keytab, krb5.conf and hive-jdbc-driver.jar

> **Note**
>
> The Datastudio placement of jars are for creating a connection from python library and the other is from java, for data sourcing.

Configure the Hive Jars and Configuration Files

For Hadoop version 3.1.1 and hive version 3.1.2, below is the list of jar files that needs to be copied into the `OFS_MMG/lib location`:

zookeeper-3.4.9.jar

woodstox-core-5.0.3.jar

stax2-api-3.1.4.jar

slf4j-log4j12-1.7.25.jar

slf4j-api-1.7.25.jar

re2j-1.1.jar

log4j-1.2.17.jar

libthrift-0.9.3.jar

libfb303-0.9.3.jar

httpcore-4.4.4.jar

httpclient-4.5.2.jar

htrace-core4-4.1.0-incubating.jar

hive-service-3.1.2.jar

hive-metastore-3.1.2.jar

hive-jdbc-3.1.2.jar

hive-exec-3.1.2.jar

hadoop-hdfs-client-3.1.1.jar

hadoop-common-3.1.1.jar

hadoop-auth-3.1.1.jar

curator-client-2.12.0.jar

commons-logging-1.0.4.jar

commons-io-2.4.jar

commons-configuration2-2.1.1.jar

commons-collections-3.2.2.jar

commons-cli-1.2.jar

The MMG service requires a restart after copying the Hive jars and configuration files. For more information, see the MMG User Guide.

Remote MMG Studio Configuration

For Solaris Operating System, the MMG Studio has to be configured in Linux machine remotely. The MMG Studio URL must be the same as that of the remote studio during MMG Application Installation.

In the `OFS_MMG/bin/config.sh`, update the following properties with the remote server where the MMG Studio will be running:

Copy the mmg-studio folder to the remote machine where you want to configure the same.

Navigate to `mmg-studio/bin` and update the config.sh file with respect to studio server values. For more details, see the Configure the config.sh file section.

```
export DATASTUDIO_URL=##DATASTUDIO_URL##

export SSL_KEYSTORE=##SSL_KEYSTORE##

export SSL_KS_SECRET=##SSL_KS_SECRET##

export SSL_KS_TYPE=##SSL_KS_TYPE##

export SSL_KS_ALIAS=##SSL_KS_ALIAS##
```

> **Note**
>
> The keystore must be generated for the remote machine and the path must be present in the remote server.

```
export DS_TNS_ADMIN_PATH=##DS_TNS_ADMIN_PATH##

export DS_WALLET_LOCATION=##DS_WALLET_LOCATION##
```

TNS admin and wallet must be configured in the remote server and the wallet must contain the mmg config schema wallet configurations.

```
export MMG_TNS_ADMIN_PATH=##MMG_TNS_ADMIN_PATH##

export MMG_LIB_WALLET_ALIAS=##MMG_LIB_WALLET_ALIAS##
```

Note

The Self signed certificate needs to be generated and imported to the Java keystore. In case self-signed certificate is being used, perform the below step:

1. Delete the already imported certificate from the Java keystore using the below command keytool:
   `-delete -noprompt -alias <alias_name> -keystore "<java_home>/lib/security/cacerts" -storepass "changeit"`
2. Run the following command to re-create a Java keystore:
   `keytool -genkey -v -alias <alias_name> -keyalg RSA -keysize 2048 -keystore server.keystore -validity 3650 -keypass secret -storepass secret -storetype PKCS12`
3. Create a `.cer file from the server.keystore. keytool -export -alias <alias>-file <filename>.cer -keystore <path_to_Keystore>/server.keystore -storepass secret`
   Example: `keytool -export -alias demo_alias -file server.cer -keystore OFS_MMG/config/server.keystore -storepass secret`
4. Import `.cer file generated from the above step to java keystore. keytool -import -file "<path_to_Keystore>/<filename>.cer" -alias <alias>-keystore "<java_home>/lib/security/cacerts" -storepass "changeit"`

Note

The above step should be performed by the Root user.

Note

The token is automated. Once the token is generated, ignore '-e' character present in the `token.out` file.

PGX Installation

Note

PGX Installation is recommended to be installed in a different server other than the MMG Installation Server.

To install the PGX, follow these steps:

1. Copy the mmg-pgx.zip file from MMG Server and copy it to the target server where PGX has to be installed remotely to MMG.
2. Unzip the mmg-pgx.zip file.

   For Example: unzip -a mmg-pgx.zip.

The below files will be displayed:

- bin
- conf
- pgx-25.4.0

3. Give 0755 permission to mmg-pgx folder.
4. Configure the config.sh of pgx. For more details, see Configure the config.sh File of PGX section.
5. Copy the graph-keystore.p12 from MMG Installation server to <pgx installation path>/mmg-pgx/conf. For more details, see Generate GRAPH-KEYSTORE.P12 section.
6. Copy the below key files from <MMG Installation path>/OFS_MMG/conf to <pgx installation path>/mmg-pgx/conf.
   - public.key
   - private.key
7. Run the install.sh from <pgx installation path>/mmg-pgx/bin
8. Update the pgx-server URL in config.sh for ##PGX_SERVER_URLS## in the <MMG Installation path>/bin and run the install.sh -u command and restart the MMG services. For more details, see Configure the config.sh File of PGX section.
9. Start the Server. For more details, see Starting PGX Server section.
10. Stop the Server. For more details, see Stopping PGX Server section.

Configure the config.sh File of PGX

To configure the config.sh file for installing PGX with MMG, follow these steps:

1. Login to the server as a non-root user.
2. Navigate to the `<OFS_MMG>/mmg-pgx/bin` directory.
3. Configure the applicable config.sh attributes as shown in the following table:

   Sample Config.sh file

```
#!/bin/sh
export PGX_PORT=##PGX_PORT##
export PGX_CONTEXT_PATH=##PGX_CONTEXT_PATH##
export PGX_SSL_ENABLED=##PGX_SSL_ENABLED##
export PGX_SSL_KEYSTORE=##PGX_SSL_KEYSTORE##
export PGX_SSL_KS_SECRET=##MMG_SSL_KS_SECRET##
export PGX_SSL_KS_TYPE=## PGX_SSL_KS_TYPE ##
export PGX_SSL_KS_ALIAS=## PGX_SSL_KS_ALIAS##
export GRAPH_SERVICE_URL=## GRAPH_SERVICE_URL##
export GRAPH_KEYSTORE_PASSWORD=## GRAPH_KEYSTORE_PASSWORD##
export LOG_HOME=##LOG_HOME##
export LOG_LEVEL=##LOG_LEVEL##
```

Table 5-2 Config.sh file of PGX

Parameter	Description	Is Mandatory	Comments
##PGX_PORT##	Port on which PGX server needs to be run.	YES	If not set, Port defaults to 7007.
##PGX_CONTEXT_PATH##	Context path of PGX server	YES	If not set, Context path defaults to PGX.
##PGX_SSL_ENABLED##	The values can be true /false. If true, follow the below steps if Self Signed is being used: • Import PGX server.cer file to MMG server java keystore • Import MMGserver.cer file to PGX server java keystore For more details, see Import Server Certificate to Java Keystore section.	YES	
Properties if ##PGX_SSL_ENABLED## is set to true.			
##PGX_SSL_KEYSTORE##	Absolute path for the keystore file. This is applicable only if ##PGX_SSL_ENABLED## is set to true. NOTE: Run the following command to create a keystore: keytool -genkey -v -alias demoalias -keyalg RSA -keysize 2048 -keystore server.keystore -validity 3650 -keypass secret -storepass secret -storetype PKCS12	YES	/conf/server.keystore. Include the file name in the path.
##PGX_SSL_KS_SECRET##	Value passed in above command for keypass. This is applicable only if ##PGX_SSL_ENABLED## is set to true.	YES	Keystore password
##PGX_SSL_KS_TYPE##	The type of the PGX keystore. This is applicable only if ##PGX_SSL_ENABLED## is set to true.	YES	PKCS12

Table 5-2 (Cont.) Config.sh file of PGX

Parameter	Description	Is Mandatory	Comments
##PGX_SSL_KS_ALIAS##	The Alias of the PGX keystore. This is applicable only if ##PGX_SSL_ENABLED## is set to true.	YES	password123
Properties for graph service			
## GRAPH_SERVICE_URL ##	Graph Service URL. The value is same as ##LOADGRAPH_BASE_URL## in the MMG.config.sh	YES	http(s)://<MMG Host>:<Graph service port>/graph-service
## GRAPH_KEYSTORE_PASSWORD ##	Graph Keystore password. The value is same as ##GRAPH_KEYSTORE_PASSWORD## in the MMG.config.sh	YES	password123
Properties for setting log path			
##LOG_HOME##	A writable folder that stores PGX logs.		/scratch/users/logs
##LOG_LEVEL##			The values can be DEBUG/INFO/WARN

Starting PGX Server

To start the PGX Server, run the following command:

- Navigate to `<MMG_INSTALLATION_PATH>/bin` directory. ./startup.sh

  You may check `<mmg-pgx/pgx-<pgx-version/bin/nohup.out` to check if the UI service comes up properly.

  A message similar to following means a successful startup:

  ```
  INFO: Starting ProtocolHandler ["http-nio-7007"]
  ```

  This will start the successful installation of PGX Server.

Stopping PGX Server

To stop the PGX Server, run the following command:

```
./shutdown.sh
```

A message similar to following means a successful shutdown:

PGX Server shutdown is complete.

R Interpreter

You can configure the R Interpreter support either with ORD-3.6.1 or R 4.1.2.

Installing Oracle R Distribution

https://docs.oracle.com/en/database/oracle/machine-learning/oml4r/2.0.0/oread/install-oracle-r-distribution-linux.html

- Check installation:
  1. R --version

R 4.1.2 Installation

> **Note**
>
> This setup might update some of the older root level files and using Non-Oracle Yum Repository for getting R rpm files.

To install R 4.1.2, follow these steps:

1. Set Proxy, (pseudo user):
   a. curl -O https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm
   b. yum install epel-release-latest-7.noarch.rpm
   c. curl -O https://cdn.rstudio.com/r/centos-7/pkgs/R-${R_VERSION}-1-1.x86_64.rpm
   d. sudo yum install R-${R_VERSION}-1-1.x86_64.rpm
   e. sudo ln -s /opt/R/${R_VERSION}/bin/R /usr/bin/R
2. Check installation:
   a. R --version

MMG Connection Objects Library Setup

This section describes the MMG Connection Objects Library Setup.

For detailed description on installation and installation of Oracle R Enterprise Client libraries (ROracle, and so on), check following link:

https://docs.oracle.com/cd/E83411_01/OREAD/installing-ORE-client.htm#OREAD167

Installing ROracle Library

Prerequisites

DBI is one of the dependencies for using this library.

- **Installing DBI**
  1. curl - O https://cran.r-project.org/src/contrib/Archive/DBI/DBI_1.1.1.tar.gz
  2. R CMD INSTALL DBI_1.1.1.tar.gz

Procedure

To install ROracle Library, follow these steps:

- For ORD 3.6.1/R 4.1.2

1. curl - O https://cran.r-project.org/src/contrib/ROracle_1.3-1.1.tar.gz
2. Install Oracle Instant Client Sdk Package. This is required for additional header files and an example makefile for developing Oracle Applications with Instant Client.
3. Oracle client lib must be present in PATH. In the .profile file, set PATH to include the appropriate `$ORACLE_HOME/bin` path.
   For example:

   ```
   PATH=$JAVA_HOME/bin:$ORACLE_HOME/bin
   ```

4. R CMD INSTALL `--configure-args='--with-oci-lib=<absolute-path-to-oracle-client-lib> --with-oci-inc=<absolute path to instantclient_21_5>/include' ROracle_1.3-1.1.tar.gz`
   For example:

   R CMD INSTALL --configure-args='--with-oci-lib=/scratch/users/oracle/app/oracle/product/19.3.0/client_1/lib --with-oci-inc=/scratch/users/oracle/instantclient-sdk/instantclient_21_5/sdk/include' ROracle_1.3-1.1.tar.gz

Installing RODBC Library

- **For ORD 3.6.1**
  1. curl -O https://cran.r-project.org/src/contrib/Archive/RODBC/RODBC_1.3-16.tar.gz
  2. R CMD INSTALL RODBC_1.3-16.tar.gz

  > **Note**
  >
  > It needs write permission to `'/usr/lib64/R/library'` or similar root directory for system installation.

- **For R 4.1.2**
  1. curl -O https://cran.r-project.org/src/contrib/Archive/RODBC/RODBC_1.3-16.tar.gz
  2. R CMD INSTALL RODBC_1.3-16.tar.gz

> **Note**
>
> LD_LIBRARY_PATH should contain path to $ORACLE_HOME/lib and check that file 'libsqora.so.19.1' exists in $ORACLE_HOME/lib. Now, set an environment variable named RODBC_DRIVER with value 'libsqora.so.19.1' whichever is present in $ORACLE_HOME/lib/ directory based on the Oracle Client Version Installation. Now for RODBC Connection to work for Sandbox, check the TNS_ADMIN path set, and then in tnsnames.ora, add the connection string details with alias as Sandbox Name. For example, if Sandbox Name is SAND1 for which the datasource is on host abc.in.oracle.com , port 1234 and service name – ABCXYZ, then in tnsnames.ora file add the following entry-
>
> ```
> SAND1 =
> (DESCRIPTION =
> (ADDRESS_LIST =
> (ADDRESS = (PROTOCOL =
> TCP)(HOST=abc.in.oracle.com)(PORT=1234))
> )
> (CONNECT_DATA =
> (SERVICE_NAME = ABCXYZ)
> )
> )
> ```
>
> If this only does not resolve the connections, then configure odbcinst.ini. / odbc.ini files as well as mentioned in Oracle Client Installation and Setup (figured by: > odbcinst -j)

Installing the IRkernel

To configure IRkernel, follow these steps:

1. The IRkernel requires jupyter installed. Create a conda environment with the IRkernel dependencies:

   ```
   conda create -p <PATH TO YOUR CONDA ENV> -y --override-channels -c
   <CHANNEL URL> jupyter_client==7.2.2 grpcio protobuf
   ```

2. Install the IRkernel in R: Open your R REPL (R does not have to be installed with conda) and run:

   ```
   install.packages('IRkernel')
   ```

3. Softlink jupyter and register IRkernel:

   ```
   ln -s <PATH TO YOUR CONDA ENV> /usr/bin/jupyter && R -e
   "IRkernel::installspec(user = FALSE)"
   ```

Configuring R Interpreter

Make sure to configure the interpreter config and set the correct path for R binary (`zeppelin.R.cmd`) and the path to Python environment which has the IRkernel installed (`zeppelin.python`).

In config.sh there is a new parameter:

export R_PYTHON_HOME=##R_PYTHON_HOME##

This is the Python Conda environment path till `<PATH TO YOUR CONDA ENV>/bin/python` created following steps mentioned in 5.9, which will set the `zeppelin.python property` in `mmg-studio/server/builtin/interpreters/r.json` file.

The other property mentioned for R binary (`zeppelin.R.cmd`) is also set in above path the default value is R so make sure that the user starting the services has access to R and alias is set or change the property according to the path of R binary.

> **Note**
>
> **Steps to enable TLS during R Interpreter Installation**
> Steps to enable TLS during R Interpreter Installation:

1. Download `Rserve_1.8-10.tar.gz` from this link: https://cran.r-project.org/src/contrib/Archive/Rserve/
2. Install openssl-devel - [yum install openssl-devel]
3. Execute below cmd:
   ```
   R CMD INSTALL --configure-args="--with-ssl-headers=/usr/include --with-ssl-libraries=/usr/lib64" Rserve_1.8-10.tar.gz
   ```

The below steps can be executed from R session to check that Rserve configuration is working before starting the services:

Start R from Terminal/Putty and execute the below commands:

- `library(Rserve)`
- `Rserve(args="--RS-conf <Rserve.conf file location including file name> --no-save")`

Using MMG Studio to Oracle Connection Objects

This section describes the Using MMG Studio to Oracle Connection Objects.

Workspaces

1. `mmg.list_workspaces()`: Used to fetch a vector of all workspaces.
   For example: `vec <- mmg.list_workspaces() vec will be vector object`
2. `mmg.attach_workspace("workspace_name ")`: A method used to set workspace.
   Sets a global `mmg_attached_WS` variable with value of `workspace_name`

   Sets a `mmg_DS_Vec` Vector Object with name and order of all datasources for attached workspace.

Sets a `mmg_WL_Vec` Vector Object with name and wallet of all datasources for attached workspace.

For example:

```
mmg.attach_workspace("SB1")
```

Connections

Following is the list of datasources related to workspace using:

- `mmg.list_datasources("SB1", 1)`: will list datasources related to SB1 workspace with order 1 as passed in second argument `mmg.list_datasources("workspace_name",order)` order is integer for specific order or null for all datasources.

  For example:

  df <- mmg.list_datasources("workspace_name",order) df will be Data.Frame Object.

  From the datasource name or order for the attached workspace, we can get the **ROracle** or **RODBC** Connection Object.

- `mmg.get_connection()`:
  datasource_name is the string name of the datasource, order is integer, library is one of **"RODBC" or "ROracle"**

  conn <- mmg.get_connection(datasource=order,conn_type="library");

  conn <- mmg.get_connection(datasource="datasource_name",conn_type="library");

  conn <- mmg.get_connection(datasource="datasource_name","library");

  conn <- mmg.get_connection(datasource=order,"library");

  conn <- mmg.get_connection("datasource_name",conn_type="library");

  conn <- mmg.get_connection(order,conn_type="library");

  conn <- mmg.get_connection("datasource_name","library");

  conn <- mmg.get_connection(order,"library");

  sets the conn variable to connection object of relevant library

Multi Level Approval

Model Pipeline deployment process by default requires one level of approval for every stage including model pipeline acceptance, promotion to production, and so on.
The requestor is allowed to select Reviewer and Approver user groups. All the user groups with MDLREVIEW function mapped to them are displayed in the Reviewers selector field. Similarly, the user groups with the MDLAPPROVE function mapped to them are displayed in the Approvers selector field. Applicable Pending requests are shown in the Reviewer/Approver tabs.

To add multi level approvers or reviewers, perform the following:

1. Navigate to `<installation directory>/MMG/OFSMMG/conf/workflow/model-pipeline/default.yml`

   Following are the default values:
   workflow:

   workflow-name: Default Workflow

num-approver-levels: 1

levels:

- level: 1

approvers:

escalation-approvers:

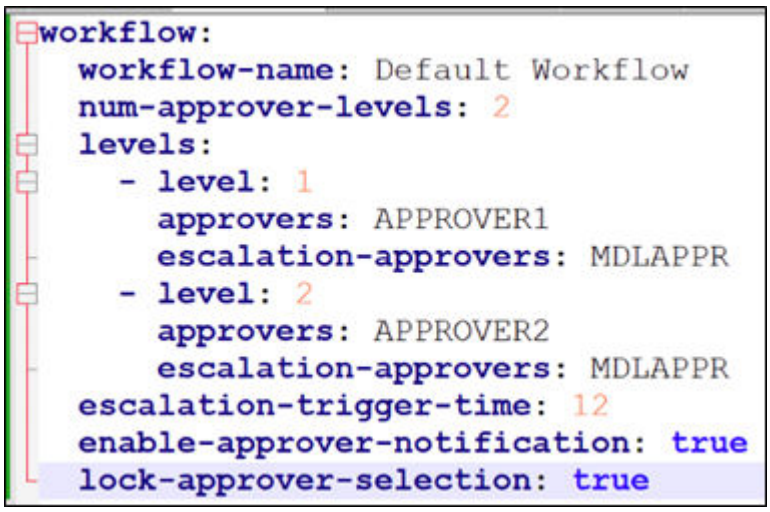
escalation-trigger-time: 0

lock-approver-selection: false

enable-approver-notification: true

2. Modify the approver levels based on your requirements as shown below.

Figure 5-1 Multi level approval



Kafka Installation

Windows

Apache Zookeeper (Download the bin tar):
https://shaaslam.medium.com/installing-apache-zookeeper-on-windows-45eda303e835#.fgofwm6n6

Apache Kafka:
https://shaaslam.medium.com/installing-apache-kafka-on-windows-495f6f2fd3c8

Linux
Method 1(Using separate zookeeper)

Apache Zookeeper

1. Download Apache Zookeeper bin files from https://www.apache.org/dyn/closer.lua/zookeeper/zookeeper-3.7.1/apache-zookeeper-3.7.1-bin.tar.gz
2. Untar `apache-zookeeper-3.7.1-bin.tar.gz tar -zxvf apache-zookeeper-3.7.1-bin.tar.gz`
3. In conf folder, rename `zoo-sample.cfg to zoo.cfg`

4. Make a folder named `data in /scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin`
5. Change the value of `dataDir in zoo.cfg to dataDir=/scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin/data`
6. Open a terminal and set the below values:
   `export ZOOKEEPER_HOME=/scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin`

   `export PATH=/scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin/bin:$PATH`
7. Start the zookeeper server:`cd /scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin/bin ./zkServer.sh start`

A few commands for zookeeper server:

- `./zkServer.sh start`
- `./zkServer.sh start-foreground`
- `./zkServer.sh stop`
- `./zkServer.sh status`

Apache Kafka

- Download Apache Kafka bin files
  `wget https://downloads.apache.org/kafka/3.3.1/kafka_2.13-3.2.0.tgz`
- Untar `kafka_2.13-3.2.0.tgz`
  `tar -zxvf kafka_2.13-3.2.0.tgz`
- Make a folder "`logs`" in the directory `/scratch/users/ofsaa/kafka/kafka_2.13-3.2.0/`.
- Go to config folder in Apache Kafka and edit `server.properties` using any text editor. Change value of `logs.dir` to `log.dirs=/scratch/users/ofsaa/kafka/kafka_2.13-3.2.0/logs in server.properties`
- Go to `config` folder in Apache Kafka and edit `zookeeper.properties` using any text editor. Change value of `dataDir dataDir=/scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin/data`
- Start Apache Kafka broker
  `cd /scratch/users/ofsaa/kafka/kafka_2.13-3.2.0/bin/`

  `./kafka-server-start.sh ../config/server.properties`

Method 2(Using zookeeper shipped with Apache Kafka)

Apache Zookeeper and Kafka

- Download Apache Kafka bin files
  `wget https://downloads.apache.org/kafka/3.3.1/kafka_2.13-3.2.0.tgz`
- Untar `kafka_2.13-3.2.0.tgz`
  `tar -zxvf kafka_2.13-3.2.0.tgz`
- Go to config folder in Apache Kafka and edit `zookeeper.properties` using any text editor. Change value of `dataDir` to `dataDir=/scratch/users/ofsaa/zookeeper/apache-zookeeper-3.7.1-bin/data` or any folder where the snapshot of zookeeper metadata can be stored.
- Make a folder "`logs`" in the directory `/scratch/users/ofsaa/kafka/kafka_2.13-3.2.0/`.

- Go to config folder in Apache Kafka and edit server.properties using any text editor. Change value of logs.dir to log.dirs=/scratch/users/ofsaa/kafka/kafka_2.13-3.2.0/logs in server.properties.
- **Start Apache Zookeeper**
  cd /scratch/users/ofsaa/kafka/kafka/kafka_2.13-3.2.0

  bin/zookeeper-server-start.sh config/zookeeper.properties
- **Start Apache Kafka Broker**
  cd /scratch/users/ofsaa/kafka/kafka_2.13-3.2.0

  bin/kafka-server-start.sh /config/server.properties

Create Keys and Certificates

- keytool -keystore kafka.server.keystore.jks -alias localhost -keyalg RSA -validity 365 -genkey -storepass oracle123 -keypass oracle123 -ext SAN=DNS:ofss-mum-1035.snbomprshared1.gbucdsint02bom.oraclevcn.com
- openssl req -new -x509 -keyout ca-key -out ca-cert -days 365
- keytool -keystore kafka.client.truststore.jks -alias CARoot -storepass oracle123 -keypass oracle123 -importcert -file ca-cert keytool -keystore kafka.server.truststore.jks -alias CARoot -storepass oracle123 -keypass oracle123 -importcert -file ca-cert
- keytool -keystore kafka.server.keystore.jks -alias localhost -storepass oracle123 -keypass oracle123 -certreq -file cert-file
- openssl x509 -req -CA ca-cert -CAkey ca-key -in cert-file -out cert-signed -days 365 -CAcreateserial -passin pass:oracle123
- keytool -keystore kafka.server.keystore.jks -alias CARoot -storepass oracle123 -keypass oracle123 -importcert -file ca-cert keytool -keystore kafka.server.keystore.jks -alias localhost -storepass oracle123 -keypass oracle123 -importcert -file cert-signed

Converting Keystore file from jks to pem

- keytool -exportcert -alias localhost -keystore kafka.server.keystore.jks -rfc -file certificate.pem -storepass oracle123
- keytool -v -importkeystore -srckeystore kafka.server.keystore.jks -srcalias localhost -keystore cert_and_key.p12 -storetype PKCS12 -storepass oracle123 -srcstorepass oracle123
- openssl pkcs12 -in cert_and_key.p12 -nodes -nocerts -out key.pem -passin pass:oracle123
- keytool -exportcert -alias CARoot -keystore kafka.server.keystore.jks -rfc -file CARoot.pem -storepass oracle123

Inside the Kafka/config path we have to create client-ssl.properties and server-ssl.properties.

The client-ssl.properties file contains the following:

- security.protocol=SSL
- ssl.truststore.location=/scratch/mmg8133/kafka/ssl/kafka.server.truststore.jks
- ssl.truststore.password=oracle123
- ssl.keystore.location=/scratch/mmg8133/kafka/ssl/kafka.server.keystore.jks
- ssl.keystore.password=oracle123

- `ssl.key.password=oracle123`

server-ssl.properties contains the same content as server.properties, but we have to update the ssl parameters in it.

Configuring Interpreters

This section provides information on configuring interpreters.

Spark Interpreter

> **Note**
>
> Java 17 supports the Spark3.4+ version, excluding the 4.0.0 series.

Interpreter Configuration

For JDBC

Update the below proeprty in jdbc.json underOFS_MMG/mmg-studio/server/builtin/interpreters

"propertyName": "default.url"

"defaultValue": "<JDBC_URL>"

For example: jdbc:oracle:thin:@ofss-mum-1033.snbomprshared1.gbucdsint02bom.oraclevcn.com:15 21/MMG19PDB

"propertyName": "default.user",

"defaultValue": "<schameusername>"

For example: The schema user to which you want to connect, for example: datastudio schema name,

"propertyName": "default.password",

"defaultValue": "<schemapassword>"

For example: Password of the provided schema user.

Start the jdbc interpreter by executing below command under //OFS_MMG/mmg-studio/interpreter-server/jdbc-interpreter-22.4.3/bin ./

jdbc-interpreter

If the jdbc interpreter needs to be included in the datastudio startup script remove the below entry from /OFS_MMG/mmg-studio/bin/startup.sh --jdbc -1.

For Spark

Configuration with Kerberos enabled remote spark cluster:

1. Copy the configured Spark directory from hadoop cluster to <MMG Studio>/interpreter-server/spark-interpreter/extralibs. For example: spark-2.4.8-bin-hadoop2.7
2. Copy the below files to the <MMG Studio>/interpreter-server/spark-interpreter/extralibs krb5.conf <keytabfile>.keytab

3. To run Spark in yarn-client mode, configure the following parameters in this file OFS_MMG/mmg-studio/server/builtin/interpreters/spark.json
spark.master = yarn-client

   spark.driver.bindAddress = 0.0.0.0

   spark.driver.host = <host> -> Apache Spark host

   > **Note**
   >
   > When using the Kubernetes interpreter lifecycle, <host> can be the IP address or hostname of any node in your Kubernetes cluster. When using the Host interpreter lifecycle, <host> should be the IP address or hostname of the node that runs the Spark interpreter.

   > **Note**
   >
   > When connecting to a YARN cluster, the Spark driver authenticates as the UNIX user that runs the Spark interpreter. You can set the HADOOP_USER_NAME environment variable to make the Spark driver authenticate as a different user. If you use the Host interpreter lifecycle, then you can do this by exporting the HADOOP_USER_NAME environment variable before starting the Spark interpreter process. If you us the Kubernetes interpreter lifecycle, then you can do this by setting the HADOOP_USER_NAME environment variable in the resource manifest (spark.yml).

4. Update file spark-defaults.conf keytab location to the location where <keytabfile>.keytab file is copied
5. Update file spark-env.sh with the krb5.conf location to the location where krb5.conf file is copied.
   For example: Djava.security.krb5.conf=/OFS_MMG/mmg-studio/interpreter-server/spark-inte rpreter-22.4.2/extralibs/krb5.conf".

6
Post-installation Steps

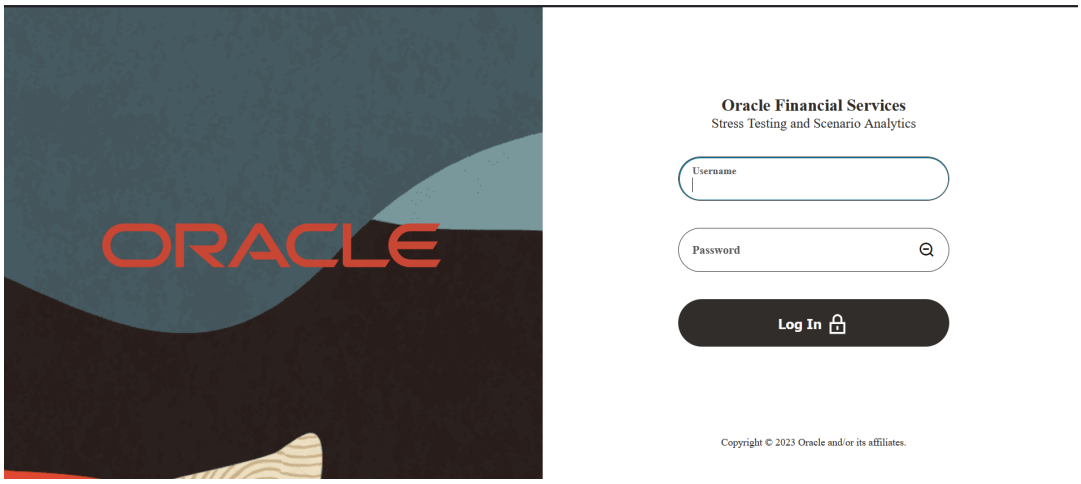
On successful installation of the application, refer to the below topics for post-installation procedures.

Access the Application

To access the application, follow these steps:

- Open a browser and enter the URL in the following format:

Figure 6-1 Login window – AAI Authentication



For more information, see the User Access and Permissioning Management section.

Create Application Users

Create the application users in the setup before starting to use the application. For more information, see the User Access and Permissioning Management section.

Map Application User(s) to User Group

User Groups seeded with the OFS MMG Application Pack are listed in the Seeded User Groups table.

Note

Run the following scripts manually for the user user-group mapping/unmapping in OFS AAI:
The following two flags must be added in the configuration table of the config schema:

```
MERGE INTO CONFIGURATION a USING (SELECT 1 FROM DUAL) b ON ( a.PARAMNAME=
'ENB_CSTM_GRP' ) WHEN NOT MATCHED THEN INSERT ( a.PARAMNAME,
a.PARAMVALUE,a.DESCRIPTION) VALUES ('ENB_CSTM_GRP','true','Enable custom
group creation during JIT')

/

MERGE INTO CONFIGURATION a USING (SELECT 1 FROM DUAL) b ON ( a.PARAMNAME=
'ENB_GRP_SYNC' ) WHEN NOT MATCHED THEN INSERT ( a.PARAMNAME,
a.PARAMVALUE,a.DESCRIPTION) VALUES ('ENB_GRP_SYNC','true','Enable
unmapping operation during JIT')

/
```

Two new flags are introduced in configuration table:

- **ENB_CSTM_GRP** - If this flag is set as *true*, then custom groups can be created. If this flag is set as *false*, then the user will not be able to create custom groups and assign it to the user.
- **ENB_GRP_SYNC** - If this flag is set as *true*, then the unmapping of users and groups are enabled. If this flag is set as *false*, then the user will not be able to unmap groups and users.

Based on these flags, unmapping of users with groups on user login is enabled.

Table 6-1 Seeded User Groups

User Group Name	User Group Description
MDLREV	The Modeling Reviewer Group. Users mapped to this group have access to the menu items in the OFS MMG Application that are related to model review activities.
MDLAPPR	The Modeling Approver Group. Users mapped to this group have the rights to approve models created by the users.
MDLBATCHUSR	The Modeling Batch User. Scheduler can use this Group for executing batches.
WKSPADMIN	The Workspace Administrator Group. Users mapped to this group have access to all the menu items in the OFS MMG Application. Additionally, they have authorization rights to create and populate workspaces.
MDLUSR	The Modeling User Group. Users mapped to this group have access to all the menu items in the OFS MMG Application that is related to model creation.

Table 6-1 (Cont.) Seeded User Groups

User Group Name	User Group Description
DSUSRGRP	General Role Users mapped to this group have permission to access/modify MMG Studio Interpreter Configurations.
DSREDACTGRP	Roles for applying redaction in graph. This group will be applicable to only those users for whom graph redaction is required.
OBJMIGADMIN	Users mapped to this group have access to Object Migration links and UI to perform import or export of objects.
GRPADMIN	The Graph Administrator Group. Users mapped to this group have access to all the menu items in the OFS MMG Application related to graph and Pipeline/Refresh graphs related health services.
GRPUSR	The Graph User Group. Users mapped to this group have access to all the menu items in the OFS MMG Application related to graph and Pipeline/Refresh graphs related health services.

Note

Admin link in the application home page is accessible only if the below seeded groups are mapped to the user:

- IDNTYADMN
- IDNTYAUTH

User Access and Permissioning Management

The application uses a realm based on unique authentication and authorization for its users. Realm indicates the functional grouping of Database Schemas and roles that must be secured for an application.

Realms protect data from access through system privileges and do not provide its owner or participants additional privileges. Realm based authorization establishes a set of database accounts and roles that can manage, or access objects protected in realms and are authorized to use its system privileges. It provides a runtime mechanism to check logically if a user's command can access objects specified in the command and proceed with its execution. Realms (AAIRealm, SAMLRealm) are selected based on the Identity Provider (IDP) during the installation. For more information, see the OFS MMG Installation Guide. After you select the realms, you can register a set of schema objects or roles (secured objects) for realm protection and authorize a set of users or roles to access the secured objects. The Application is accessed using the following realms that you have selected during the installation of the Application:

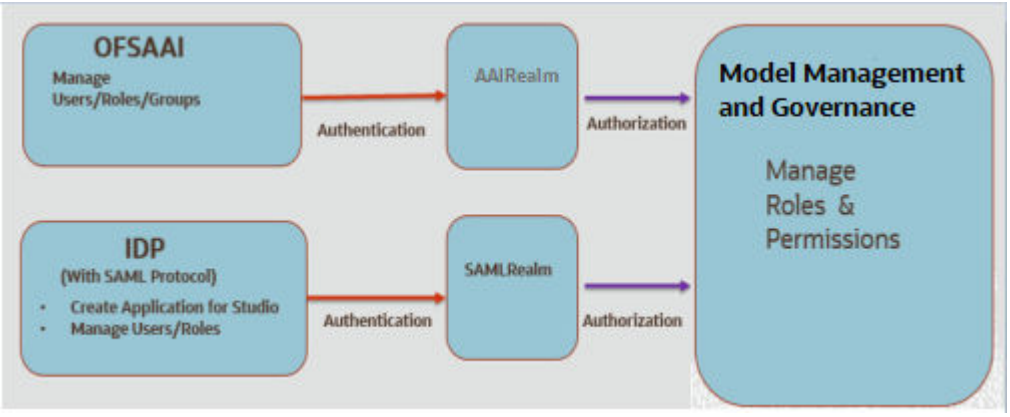
- **AAIRealm:** This uses Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Identity Management System for User Authentication. Users, Roles, and Groups

are created in the OFSAAI. The OFSAAI facilitates System Administrators to provide access, monitor, and administer users along with the infrastructure metadata operations. The required permissions to roles or groups are authorized in the application by using the Permission feature.

- **SAMLRealm:** The SAMLRealm uses an identity provider (IDP) Identity Management System for User Authentication. Security Assertion Markup Language (SAML) is an open standard that allows Identity Providers (IDP) to pass authorization credentials to Service Providers (SP). IDP acts as the Single Sign-On (SSO) service. Users and Roles are created in the IDP. The required permissions to Users and Roles are authorized in the application by using the Permission feature.

The following image illustrates the authentication and authorization process in the application.

Figure 6-2 Authentication and Authorization process in the application



Mapping User Groups

Users must be mapped to User Groups that are mapped to access the application. The following subsections provide information about the user groups and roles required in addition to the information about configuring the user groups.

User Groups

Table 6-2 User Groups

User Group	Description
IDNTYADMN	Identity Administrator group
IDNTYAUTH	Identity Authorizer group
MDLREV	The Modeling Reviewer Group. Users mapped to this group have access to the menu items in the application that are related to model review activities
MDLAPPR	The Modeling Approver Group. Users mapped to this group have the rights to approve models created by the users.
MDLBATCHUSR	The Modeling Batch User. Scheduler can use this Group for executing batches.

Table 6-2 (Cont.) User Groups

User Group	Description
WKSPADMIN	The Workspace Administrator Group. Users mapped to this group have access to create and populate workspaces. For viewing the landing page this group is required.
MDLUSR	The Modeling User Group. Users mapped to this group have access to all the menu items in the application that is related to model creation.
DSUSRGRP	Data Studio User Group This User Group provide access to modify Interpreter configurations.
GRPADMIN	The Graph Administrator Group Users mapped to this group have access to all the menu items in the application related to graph as well as Pipeline/Refresh graphs related health services.
GRPUSR	The Graph User Group Users mapped to this group have access to all the menu items in the application related to graph as well as Pipeline/Refresh graphs related health services.
DSREDACTGRP	Roles for applying redaction in graph. This group will be applicable to only those users for whom graph redaction is required. **Note**: This group has to be created manually in AAI and map it to the users.
FILEADMINUSER	User Group for admin level access to files
FILEREADUSER	User Group for read access to files
FILEWRITEUSER	User Group for read and write access to files

Note

- At the first-time login, User Group mappings are initialized from AAI/IDCS for the newly provisioned users. These will be reflected in Admin Console in the next login.
- If User Group mappings are deleted in AAI/IDCS, it would not delete in the Admin Console. Admin needs to delete this in the Identity screens too.
- Only the group with MDLSUMM role will be displayed in the Workspace provisioning steps.
  MDLSUMM function is mapped to the MDLACCESS role.

User Group - Role Mapping

Map the user groups in the application to the roles in the following table to enable access to the OFS CS application.

Table 6-3 User Group to Role Mapping

Group Name	Role Name
DSREDACTGRP	DSREDACT
IDNTYADMN	Batch Advance Role

Table 6-3 (Cont.) User Group to Role Mapping

Group Name	Role Name
IDNTYADMN	Batch Write Role
IDNTYADMN	Admin Link Role
IDNTYADMN	User Advanced Role
IDNTYADMN	Group Advanced Role
IDNTYADMN	Role Advanced Role
IDNTYADMN	Function Advanced Role
IDNTYAUTH	Group Authorize Role
IDNTYAUTH	User Authorize Role
IDNTYAUTH	Group Read Role
IDNTYAUTH	Admin Link Role
IDNTYAUTH	Function Read Role
IDNTYAUTH	Role Read Role
IDNTYAUTH	Role Authorize Role
MDLAPPR	DSINTER
MDLAPPR	Model Authorize
MDLAPPR	Model Deployment
MDLAPPR	Workspace Read
MDLAPPR	Model Read
MDLAPPR	Model Access
MDLAPPR	Workspace Access
MDLAPPR	DSAPPROVER
MDLBATCHUSR	DSBATCH
MDLREV	Workspace Read
MDLREV	Model Review
MDLREV	Model Access
MDLREV	Workspace Access
MDLREV	DSUSER
MDLREV	Model Read
MDLUSR	Model Advanced
MDLUSR	Model Write
MDLUSR	Model Read
MDLUSR	Batch Advance Role
MDLUSR	Model Execute
MDLUSR	DSUSER
MDLUSR	Model Access
MDLUSR	Workspace Access
MDLUSR	Workspace Read
MDLUSR	Datastore Access
MDLUSR	Datastore Write
MDLUSR	Datastore Read
WKSPADMIN	Workspace Access
WKSPADMIN	DSADMIN
WKSPADMIN	Identity MGMT advanced

Table 6-3 (Cont.) User Group to Role Mapping

Group Name	Role Name
WKSPADMIN	Workspace Authorize
WKSPADMIN	Workspace Read
WKSPADMIN	Workspace Write
DSUSRGRP	DSADMIN
GRAPHUSER	Graph Administrator
GRAPHUSER	Graph Read Role
GRAPHUSER	Graph Read Role
GRAPHUSER	Graph Execute Role
GRAPHADMINISTRATOR	Graph Administrator Role

Functions and Roles required to perform CRUD operations for Conda

The following table provides details about the Functions and Roles required to perform CRUD operations for Conda in the application.

Table 6-4 Functions and Roles

Function	Role	Groups Mapped	Access
CONDAENVSUMM	CONDAENVACCESS	• MDLUSR • MDLREV • MDLAPPR	Summary view
CONDAENVVIEW	CONDAENVREAD	• MDLUSR • MDLREV • MDLAPPR	Read
CONDAENVEXP	CONDAENVREAD	• MDLUSR • MDLREV • MDLAPPR	Export yml file
CONDAENVEXP	CONDAENVWRITE	• MDLREV • MDLAPPR	Export yml file
CONDAENVDEL	CONDAENVWRITE	• MDLREV • MDLAPPR	Delete a registered conda environment
CONDAENVEDIT	CONDAENVWRITE	• MDLREV • MDLAPPR	Edit a conda environment
CONDAENVADD	CONDAENVWRITE	• MDLREV • MDLAPPR	Add a conda environment

Access MMG using AAI Realm

This section provides information on creating users who can access MMG using the AAI Realm method of authentication through Oracle Financial Services Analytical Applications Infrastructure (OFSAAI). The users with SYSADMN and SYSAUTH roles in OFSAAI can create and authorize users, respectively.

Identity Management in the OFSAAI facilitates System Administrators to provide access, monitor, and administer users along with the infrastructure metadata operations. The Security Management System (SMS) component is incorporated with Password Encryption, Single

Logon, Role and DataBased Security, Access Control, and Audit Trail feature to provide a highly flexible security envelope. Administrators can create, map, and authorize users defining a security framework that can restrict access to the data and meta-data in the warehouse, based on a fine-grained access control mechanism. These activities are done at the initial stage and then on a required basis. For more information on creating and authorizing users in OFSAAI, see the Oracle Financial Services Analytical Applications Infrastructure User Guide. The following table describes the ready-to-use roles and the corresponding user groups who can access MMG using AAIRealm.

> **Note**
>
> Only in AAIRealm, users are mapped to user groups. The default permissions mapped to these users and user groups are available in the Permission section. However, these permissions can be added or modified.

Prerequisites

1. Configuring WebLogic for REST Services Authorization.

   To enable REST API authorization by OFSAA in WebLogic server, perform the following steps:

   a. Open the config.xml file located in the domain where OFSAA is deployed that is: `<domain_home>/config/config.xml`.

   b. Add the following in the security-configuration tag: `<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-authcredentials>`.

2. In OFSAA Application, **Allow user to log in from multiple machines** option should be enabled.

   REST API Calls
   There is need for Model Scripts to access input parameters passed by REST API Calls:

   a. The REST API optional parameters are mapped to runtime parameters , this mapping needs to be documented. It is not apparent from runtime parameter section.

   b. The mapped simple params can be accessed with script segment like below , but it is not supporting base64 encoding and/or json hp = '${hyperparams}' cnf = '${conf}' 3.the enhancement against json is retracted as we can do base64 encrypted value passing
   {

   ```
   "RUN_ID": 2,

   "RUN_CODE": 200440,

   "PARAM_SET": {

   "CALL_TYPE": "PASS_THROUGH_RATE",

   "BOLLINGER_PERIOD": 3,

   "BOLLINGER_COEFFICIENT": 1,

   "DATA_SNAP_FREQ": 1,

   "DATA_SNAP_FREQ_MULTIPLIER": "D",

   "MISSING_VALUE_TECH": "inpol",

   "MISSING_THRESHOLD_PERCENTAGE": 30,
   ```

```
"START_DATE": 20230131,
"END_DATE": 20230431,
"EXECUTED_BY": "TESTUSR",
"REFRESH_DATA": "Yes"
},
"SLICE": [
{
"SEQ": 1,
"LOGICAL_PARTITION_NAME": "DEFAULT",
"TABLE_NAME": "FSI_I_NM_MLSTAGE"
}
],
"OUTPUT_DATA": {
"PLOTS_TAB": "FSI_O_NM_PLOTS",
"MILESTONE_TAB": "FSI_O_NM_MILESTONE",
"SCORE_TAB": "FSI_O_NM_MODEL"
}
}
```

For example: base64 encryption

Access MMG using SAMLRealm

This section provides information on managing users who can access MMG with Identity Provider (IdP or IDP). An Identity Provider (IdP) is a service that stores and verifies user identity. IdPs are cloud-hosted services, and they often work with single sign-on (SSO) providers to authenticate users. An Identity Provider stores and manages users' digital identities. An IdP checks user identities via username-password combinations and other factors, or it may simply provide a list of user identities that another Service Provider (like an SSO) checks. The following are the ready-to-use roles that can access MMG using SAMLRealm. To integrate MMG with IdP as the SSO Provider, follow these steps:

1. Create the following roles in the IDP System:

   For MMG:

   - IDNTYADMN
   - IDNTYAUTH
   - MDLREV
   - MDLAPPR
   - MDLBATCHUSR
   - WKSPADMIN
   - MDLUSR
   - DSUSRGRP
   - DSREDACTGRP

- GRPADMIN
- GRPUSR

> **Note**
>
> IDNTYADMN role is required only if you need the Admin Access.

2. Map the user groups to the respective user based on the user roles. The default permissions mapped to these users are available in the Permission section. However, these permissions can be added or modified.

> **Note**
>
> It is recommended to use AAIRealm or SAMLRealm.

Access the Application by Using SAML Realm

This section provides information on managing users who can access the application with Identity Provider (IdP or IDP). The IdP acts as the Single Sign-On (SSO) service provider for implementations between the application, Investigation Toolkit, and Enterprise Case Management. This configuration prevents separate login for each application.

An identity provider (IdP) is a service that stores and verifies user identity. IdPs work with single sign-on (SSO) providers to authenticate users. An identity provider (IdP or IDP) stores and manages users' digital identities. An IdP checks user identities via username-password combinations and other factors, or it may simply provide a list of user identities that another service provider (like an SSO) checks.

See the User Groups section for Pre-configured Groups to access the application using SAML Realm.

> **Note**
>
> You can configure SAML in the following ways:
>
> - SAML for Authentication and AAI for Authorization
> - SAML for Authentication and SAML for Authorization
>
> For more information, see the respective sections in the Installation Guide.

To integrate the application with IDP as the SSO provider, follow these steps:

1. Create the following Group in the IDP system. For more information on creating groups in IDP, see the OFS Admin Console User Guide.
   - Create the new groups with the same name as the pre-configured groups. For more information, see the User Groups section.
2. Create a SAML application in IDP.
3. Configure the SAML application. Key configurations in the SAML application is as follows:
   - **Entity ID:** `https://<FQDN of <application name> Linux Server>:7001/<application name>`

- **Assertion Consumer URL:** `http://<FQDN of <application name> Linux Server>:7001/<application name>/home`

  > **Note**
  >
  > – If the application Gateway service is enabled, then the value of this port will be the GATEWAY PORT. For example, 7071.
  >
  > – Response in SAML response must be signed.

- **Include Signing Certificate in Signature**: Enabled
- **Signature hashing algorithm**: SHA-256
- **Enable Single Logout**: Enabled
- **Logout Binding**: POST
- **Single Logout URL (SAML_LOGOUT_URL)**: `http://<FQDN of <application name>>:7001/<application name>/signoff`

  > **Note**
  >
  > If the application Gateway service is enabled, then the value of this port will be the GATEWAY PORT. For example, 7071.

- **Logout Response URL**: `http://<FQDN of <application name>>:7001/<application name>/signoff`

  > **Note**
  >
  > If the application Gateway service is enabled, then the value of this port will be the GATEWAY PORT. For example, 7071.

- **Encrypt Assertion**: Disabled
- SAML Attribute Configuration
  Update the SAML attribute configuration as described in the following table.

Table 6-5 Attribute Configuration

Name	Format	Type	Value	Condition
ofs_mapped_groups	Basic	User Attribute	Group Member	All Groups
email	Basic	User Attribute	Primary Email	-
username	Basic	User Attribute	Last Name	-

4. Create a user and map the user groups to the respective user based on the user roles.

Access Data Studio Using SAML Realm

This section provides information on managing users who can access Data Studio with Identity Provider (IdP or IDP). The IdP acts as the Single Sign-On (SSO) service provider for

implementations between the application, MMG and Data Studio. This configuration prevents separate login for each application.

An identity provider (IdP) is a service that stores and verifies user identity. IdPs work with single sign-on (SSO) providers to authenticate users. An identity provider (IdP or IDP) stores and manages users' digital identities. An IdP checks user identities via username-password combinations and other factors, or it may simply provide a list of user identities that another service provider (like an SSO) checks.

Users should map the following user groups to access the Data Studio and Investigation Toolkit:

- **DSUSRGRP**: Grants admin privileges for Data Studio

To integrate Data Studio with IDP as the SSO provider, follow these steps:

1. Create the following Group in the IDP system. For more information on creating groups in IDP, see the OFS Admin Console User Guide.
   - Create the new groups with the same name as the pre-configured groups. For more information, see the User Groups section.
2. Create a SAML application in IDP for Data Studio.
3. Configure the SAML application. Key configurations in the SAML application is as follows:
   - **Entity ID:** `https://<Hostname>:7008/<application name>`
   - **Assertion Consumer URL:** `http://<Hostname>:7008/<application name>/saml/consume`
   - **Include Signing Certificate in Signature**: Enabled
   - **Signature hashing algorithm**: SHA-256
   - **Enable Single Logout**: No
   - **Require encrypted assertion**: No
4. Update the SAML attribute configuration as described in the following table.

Table 6-6 SAML Attribute Configuration

Name	Format	Type	Value	Condition
ofs_mapped_groups	Basic	User Attribute	Group Member	All Groups
email	Basic	User Attribute	Primary Email	-
username	Basic	User Attribute	Last Name	-
group	Basic	User Attribute	Group Member	All Groups

5. Create a user and map the user groups to the respective user based on the user roles.
6. After creating the application, download the “Signing Certificate” of the SAML application of the Data Studio and rename it to “key.cert” file and place in the following location:
   - `<<application name>_PATH>/mmg-home/mmg-studio/conf`
7. Restart the application.

Model Techniques/ Model Library

Following are the prerequisites to use the model techniques from the older version when you upgrade to 8.1.2.4.0 version.

> **Note**
>
> MMG_TECHNIQUE_MASTER table had no V_WORKSPACE_ID column, which has been added in this release and then the primary key is updated to (V_TECHNIQUE_ID, V_WORKSPACE_ID).

To use the existing Techniques in the upgraded setup, perform the below:

The V_WORKSPACE_ID column will have the value set as ##WORKSPACE## for the existing records by default. If the same records has to be used in the latest version of MMG, you must update the table MMG_TECHNIQUE_MASTER with relevant Workspace ID.

.PEM file creation for Model Service

You must create **server.pem** file from **server.keystore** in the same path where server.keystore file is present using the below command:

```
openssl pkcs12 -in <Path_To_server.keystore> -out <Path_To_Server.pem> -nodes
```

For Example:

```
openssl pkcs12 -in

/scratch/users/ofsaa/dev_home/config/server.keystore -out

/scratch/users/ofsaa/dev_home/config/server.pem -nodes
```

AAI User Provisioning SQL Scripts Generator Utility

This utility allows you to use AAI for authN in MMG. Identity administrators can create new user groups or roles, perform appropriate roles, usergroup and domain mapping, and so on.

This is provided as a SQL generator utility. This SQL scripts is executed in the AAI's configuration schema to create the required metadata.

Ensure that you run this script multiple times against each username. Additionally, generate the merge scripts accordingly.

Execute the following command from `<mmg-home>/bin` folder

```
./userprovisioning-script-generator.sh <user> <comma separated listof user groups
or ALL> <infodom> <segment>
```

Sample Commands:

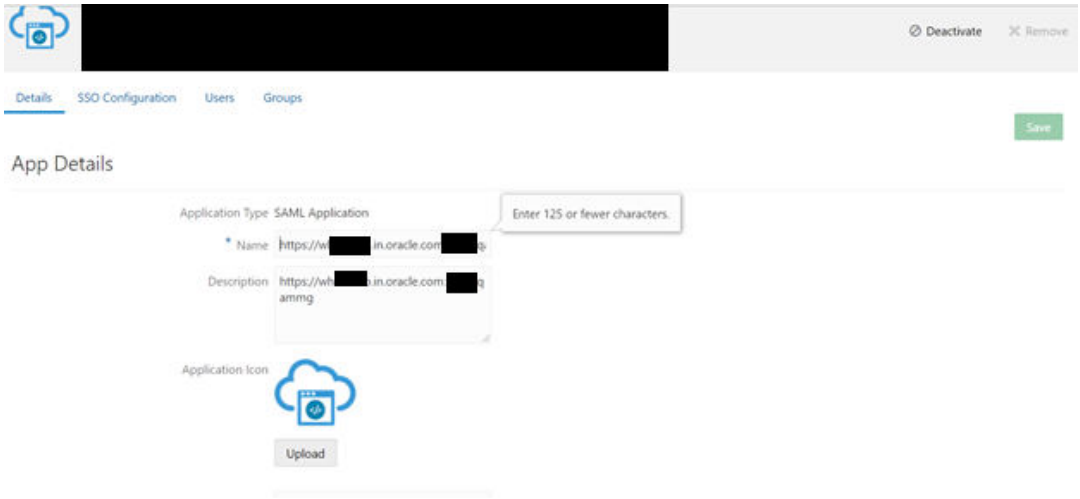
```
./userprovisioning-script-generator.sh SCRIPTUSER ALL OFSAAAIINFO EMFLD

./userprovisioning-script-generator.sh SCRIPTUSER MDLREV,MDLUSR,IDENTITY_ADMIN

OFSAAAIINFO EMFLD
```

IDCS Server Configuration

To perform IDCS Server Configuration, follow these steps:

1. Navigate to SAML IDCS Admin.
2. Navigate to Details section and add the app details in IDCS Server as shown below:

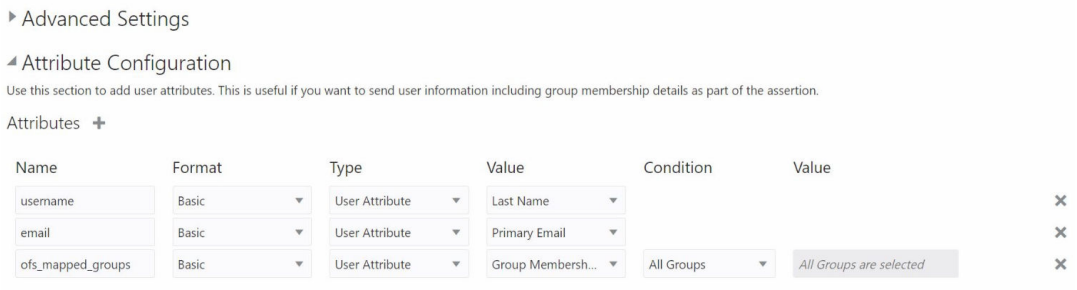
Figure 6-3 IDCS Server



3. Navigate to SSO Configuration section and add the app details in IDCS Server as shown below:

Figure 6-4 SSO Configuration section

Figure 6-5 IDCS Server



> **Note**
>
> The following attributes such as username, email, and ofs_mapped_groups needs to configured as shown in the above image.

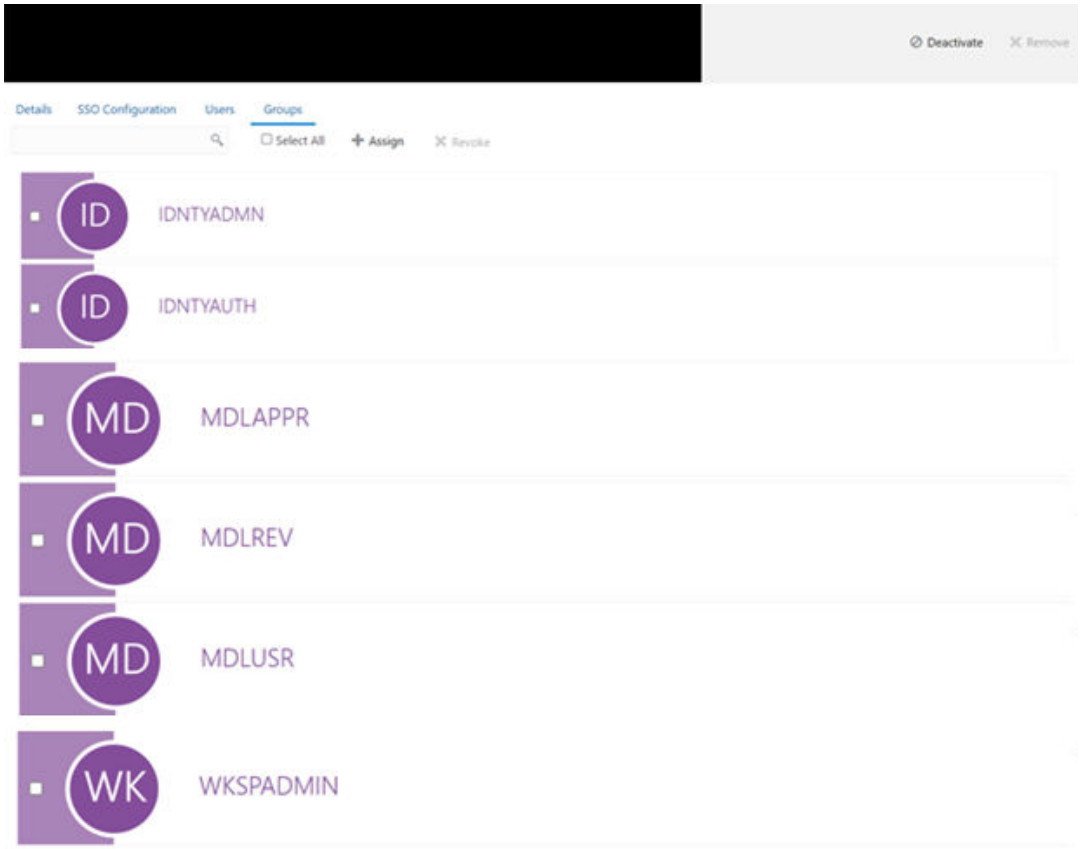
The user has to upload the certificate.

The user can generate the certificate by following these steps:

- Create .cer and .pem file from the below command.
- Upload the same .cer to the idcs Signing certificate option as well:
  openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout sp-privatekey.pem -out sp-certificate.cer

4. Navigate to Group section and Configure User Groups.

Figure 6-6 Configure User Groups in Group section



Conda Environment Migration and Restoration

Use this process when migrating or restoring Conda environments from a reliable source (such as production) to a sandbox instance

Source Environment (Export Steps)

Perform the following steps:

1. Prepare the backup directory:
   - Create a directory to store the exported environments by executing the following command:
     `mkdir conda_export`
   - Ensure that the mount point has sufficient storage.
2. Activate the environment.
   Activate the environment you wish to export:

   ```
   conda activate <conda_env_name>

   # or, for older conda:

   source activate <conda_env_name>
   ```

3. Export the Environment.
   Pack the active conda environment (ignore missing files):

```
conda pack -n <conda_env_name> -o <conda_env_name>.tar.gz --ignore-missing-
files
```

> **Note**
>
> Do not use this method for custom environments without prior validation.

4. Deactivate the Environment.
   Deactivate the environment after packing:

   ```
   conda deactivate

   # or, for older conda:

   source deactivate
   ```

5. **Pre-requisite** - Install the conda-pack (if it is missing).
   - If the conda-pack is not installed, run:
     `pip install conda-pack`
   - Use index-url or extra-index-url if your repository requires them.

Target Environment (Import Steps):

Perform the following steps to transfer the environment archive:

1. Transfer the environment archive.
   Copy the exported .tar.gz file to the target server (For example, use scp or another file transfer tool).
2. Prepare the target directory.
   a. Log in to the target server.
   b. Navigate to the <<miniconda3>>/envs directory:
      `cd <<miniconda3>>/envs`
   c. Create a directory for the environment to be imported:
      `mkdir <Env_to_be_imported>`

      # For example:

      `mkdir ml4aml_8.1.2.6.0`
   d. Rename the existing tar files if present, to avoid overwrites.
3. Untar the environment.
   Extract the archive contents into the newly created environment directory:

   ```
   tar -xzf <Env_to_be_imported>.tar.gz -C <Env_to_be_imported>
   ```

4. Test the python executable.
   a. Before activating, you can test the environment's python:
      `./<Env_to_be_imported>/bin/python`
   b. You must see the python prompt; exit with quit().
5. Activate the environment.
   - Activate the imported environment:
     `source <Env_to_be_imported>/bin/activate`

> **Note**
>
> Your prompt must now reflect the activated environment.

6. Verify the python version.
   Confirm you are running python from within the activated environment:

   ```
   python
   ```

7. Cleanup prefixes.
   Fix any hardcoded prefixes in the environment:

   ```
   conda-unpack
   ```

   This step can be performed either within the activated environment or by specifying the path to the python binary.

8. Register the imported environment.
   Add the environment path to your environment tracking file (optional but recommended for automation):

   ```
   echo "<<miniconda3>>/envs/<Env_to_be_imported>" >> ~/.conda/Environment.txt
   ```

9. List the conda environments.
   Confirm the new environment appears in your environment list:

   ```
   conda env list
   ```

10. Enroll or use the environment as needed.
    Proceed with any application-specific enrollment steps.

Manual Configuration of Email Notification for Scheduler Batch Execution

This chapter provides information how to manually configure the SMTP settings.

Prerequisites

- Database access privileges to update the configuration schema.
- Knowledge of your SMTP server's host name and port.

To manually enable email notifications for OFS MMG, administrators must populate the SMTP configuration details into the `AAICL_SC_COMPONENT_DETAILS` configuration table present in the OFS MMG configuration schema. This is required for environments running versions of OFS MMG released prior to the automated installer enhancement.

Execute the Insert Statements

Replace `{{<smtphost>}}` and `{{<smtpport>}}` with your actual SMTP server host name and port number.

```
INSERT INTO AAICL_SC_COMPONENT_DETAILS

(V_COMPONENT_ID, V_PROPERTY_ID, V_PROPERTY_VALUE, V_CREATED_BY, D_CREATED_DATE,
V_MODIFIED_BY, D_MODIFIED_DATE)

VALUES

('AAICL_EMAIL', 'host', '<smtphost>', NULL, NULL, NULL, NULL);
```

Validate the Configuration

Verify the entries by using a SELECT statement:

```
SELECT * FROM AAICL_SC_COMPONENT_DETAILS WHERE V_COMPONENT_ID = 'AAICL_EMAIL';
```

7

Upgrade Installation

This chapter describes the Upgrade Installation.

Upgrading to 8.1.3.3.0

To update an already installed MMG Application, perform the following steps:

Prerequisite:

- A valid working setup should be available before performing the upgrade.
- Use the MMG Config Schema, MMG Graph Schema, and MMG Datastudio Schema, from the existing version, along with the wallet configurations.
- Shutdown all the services of the existing installation using `shutdown.sh`.
- Backup the existing MMG Installation to a backup folder.

Upgrade:

Follow steps mentioned in the Installation section.

> **Note**
>
> Compare and copy the placeholder values from the existing installed MMG.config.sh to the new one mentioned in the Installation section.

8

Update Utility to Reconfigure Installation Parameters

If you need to update any of the existing configuration related values, perform the following steps:

> **Note**
>
> This Utility is applicable from 8.1.2.3.0 version onwards.

Procedure:

1. Shut down all the services using shutdown.sh command.
2. Reconfigure the config.sh file with the required changes.
3. Execute the command `install.sh -u` from the following path: <mmg installation path>/OFS MMG/bin
   A successful update message as follows:

```
nohup: ignoring input and redirecting stderr to stdout
PIPELINE_HOME: /scratch/ofsaaapp/OFS_MMG/mmg-pipeline/pipeline
/scratch/ofsaaapp/OFS_MMG/mmg-pipeline/pipeline
PIPELINE_HOME: /scratch/ofsaaapp/OFS_MMG/mmg-pipeline/pipeline
Installing Pipeline Data Model. Please Wait ...
Pipeline Data Model installation finished.
Starting Pipeline Service...
Starting Data pipeline services...
Inserting DataMeta Data...
**************** Data Pipeline Deployment Done ****************
Stopping Graph-Service service...
Graph-Service stopped.
Stopping Graph-Service service...
Graph-Service stopped.
nohup: ignoring input and redirecting stderr to stdout
```

4. Start all the MMG services using `startup.sh` command.

9

Cloning the MMG Instance

There is a consistent requirement for a faster and effective approach of replicating an existing MMG Instance for further project developments. The approach is to set up the MMG Instances that are exact copies of the current MMG Instance.

Copying the Directories

The Installation Directory structure in the base environment has to be replicated in the clone environment.

- Copy the MMG base directory (OFS_MMG, by default) in the base environment with all of its contents to the clone environment.
  The base directory in the clone environment will have the following folders upon copying:
  - mmg-ui
  - mmg-studio
  - mmg-service
  - mmg-schema-creator
  - mmg-pipeline
  - lib
  - bin
  - conf

> **Note**
>
> You need to copy LOG and FTPSHARE directories to the cloned environment.

Copying the Database Schemas

To copy the Database Schemas:

1. Create a copy each of the MMG Config Schema and the Data Studio Schema.
   You may use Oracle Data Pump Export/Import or the Database Copy feature of Oracle SQL Developer. For more details, see Database Copy using Oracle SQL Developer.

   The Cloned Schemas can be created either in the same database instance or in a different one.
2. Similarly, create copies of Workspace Schemas or other Data Source Schemas as required.

Configuring Password Store with Oracle Wallet

To configure the password store with Oracle Wallet:

- Setup an Oracle wallet in the clone environment. For more details, see Setup Password Stores with Oracle Wallet.

> **Note**
>
> It is recommended to use the same wallet aliases used in the base environment.

Updating the WALLET_LOCATION and TNS_ADMIN_PATH

Update the WALLET_LOCATION and TNS_ADMIN_PATH values in config.sh file present in the following path: `OFS_MMG/bin` with configured corresponding values of the cloned environment.

Updating the Host Details

Update the HOST and PORT values in `config.sh` file present in the following path: `OFS_MMG/bin` with configured corresponding values of the cloned environment.

> **Note**
>
> It is recommended to use the same ports and context used in the base environment.

Replace the placeholders and update the host name in the MMG Config schema using the following command:

```
update NEXTGENEMF_CONFIG set V_VALUE =

'http(s)://##HOST_NAME##:##BE_PORT##/##CONTEXT##' where V_NAME in ( '

BASE_URL', ‘EMFSTUDIO_SERVICE_URL’)

/

update NEXTGENEMF_CONFIG set V_VALUE =

'http(s)://##HOST_NAME##:7008/##CONTEXT##' where V_NAME = ‘DATASTUDIO_URL’

/

update AAICL_SS_BATCH_URL set V_URL =

'http(s)://##HOST_NAME##:##BE_PORT##/##CONTEXT##' whereV_URL_NAME in

('CS_SERVICE_URL', 'MMG_SERVICE_URL', 'WORKSPACE_URL')

/
```

Update LOG_HOME and FTPSHARE

Update the LOG_HOME and FTPSHARE values in `config.sh` file present in the following path:

`OFS_MMG/bin` with configured corresponding values of the cloned environment.

Replace the ##LOG_HOME## and ##FTPSHARE## placeholders and update the LOG_HOME and FTPSHARE values in the MMG Config Schema using the following command:

```
update NEXTGENEMF_CONFIG set V_VALUE = '##LOG_HOME##' where V_NAME = 'LOG_HOME'

/

update NEXTGENEMF_CONFIG set V_VALUE = '##FTPSHARE##' where V_NAME = 'FTPSHARE'

/
```

Setting up the SSL Keystore

To run on HTTPS, you must create a Keystore for MMG Application. For more details, see the SSL Keystore in the Configure the config.sh file.

Update the Keystore path, Password and Storetype values in config.sh file present in the following path: `OFS_MMG/bin` with configured corresponding values of the cloned environment.

Updating Wallet Aliases for Oracle Schemas

> **Note**
>
> It is recommended to use the same wallet aliases used in the base environment.

In case if the same wallet aliases cannot be used, perform the following:

1. Update the MMG Config Schema Wallet Alias values in `config.sh` file present in the following path: `OFS_MMG/bin` with configured corresponding values of the cloned environment.
2. Replace the placeholders and update the wallet alias for Workspace Schemas or other Oracle datasources using the following command:

   ```
   update MMG_DB_MASTER set V_PROPERTY_VALUE = '##WALLET_ALIAS##' where
   V_PROPERTY_NAME = 'WALLET_ALIAS' and V_DB_NAME = '##DATASOURCE NAME##'

   /
   ```

Updating Context and Ports

> **Note**
>
> It is recommended to use the same context and ports used in the base environment.

In case if the same context and ports aliases cannot be used, perform the following:

1. Update the references of context path and port values in config.sh file present in the following path: `OFS_MMG/bin` with configured corresponding values of the cloned environment.
2. Replace the ##CONTEXT## and ##BE_PORT## placeholders.

For more details, see Updating the Host Details.

> **Note**
>
> For MMG authentication, port access should be enabled for MMG ports to access OFSAA ports.

Starting MMG Services

Post updating all the required parameters in the new `config.sh.`file, start the services by using the following command: `./install.sh -u`

Cloning the Environment

We will not have to manually update the entries or Database details/service URLs for most of the tables.

This can be done in two parts:

Part I

- Execute a utility that updates most of the entries in config tables and configuration/ properties in the file system.
- In the DR setup, after cloning, configure the <base>/mmg-home/bin/config.sh with new values of:
  - WALLET_LOCATION (pointing to DR db)
  - TNS_ADMIN_PATH (pointing to DR db)
  - FTPSHARE (Ensure contents of ftpshare are also synced up between Prod and DR instances).

> **Note**
>
> All other values, as applicable (If they differ between Prod and DR environments).

- Execute the update script: `<base>/mmg-home/bin/install.sh -u`

> **Note**
>
> This is not going to install again. Instead this will just update the configuration values in config tables and files.

Part II

For rest of the tables, additionally we can execute sql to update the following:

Replace the PARAMVALUE for PARAMNAME = ¿ER_SERVICE_URL¿

`fcc_pgx_m_config|V_GRAPH_PATH|`: Replace this if it is not null; as this is applicable only for IN-Memory graphs.| |MMG_MENU|V_MENU_URL|For example:

Table 9-1 Param Value for Param Name with Service URL

V_MENU_CODE	V_MENU_NAME	V_MENU_URL
C003	Match Rules	https://ofss-mum-1750.snbomprshared1.gbucdsint02bom.oraclevcn.com:7061/fcc/matchrulesetsummary.jsp
C004	Merge Rules	https://ofss-mum-1750.snbomprshared1.gbucdsint02bom.oraclevcn.com:7061/fcc/mergerulesetsummary.jsp
C005	Data Survival	https://ofss-mum-1750.snbomprshared1.gbucdsint02bom.oraclevcn.com:7061/fcc/datasurvivalsummary.jsp
C001	Manual Decisioning	https://ofss-mum-1750.snbomprshared1.gbucdsint02bom.oraclevcn.com:7061/fcc/manualdecisioning.jsp
C006	Merge and Split Global Entities	https://ofss-mum-1750.snbomprshared1.gbucdsint02bom.oraclevcn.com:7061/fcc/mergeandsplit.jsp

10

Frequently Asked Questions (FAQs) and Error Dictionary

This section consists of resolution to the Frequently Asked Questions and Error Codes noticed during the installation.

Topics:

Related Topics

- Frequently Asked Questions
- Frequently Asked Questions (FAQs) and Error Dictionary

Frequently Asked Questions

You can refer to the Frequently Asked Questions, which is developed with the interest to help you resolve some of the installation and configuration issues. This intends to share the knowledge of problem resolution to a few of the known issues. This is not an official support document and just attempts to share the knowledge of problem resolution to a few of the known issues.

Frequently Asked Questions

1. Why does my console show an unsuccessful message during wallet creation?
   Please check if you have run the following commands correctly. For more information on wallet creation, see Setup Password Stores with Oracle Wallet.
   a. `mkstore -wrl <wallet_location> -create` //creates a wallet in the specified location.
   b. `mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>` //creates an alias in the Studio Schema.
   c. `mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>` //creates an alias in the Atomic Schema.
   d. `mkstore -wrl <wallet_location> -createCredential <alias-name> <database-user-name>` //creates an alias in the configuration schema.

   If your issue is still not resolved, contact My Oracle Support (MOS).
2. Where can I find my created wallet?
   Your wallet will be in the directory you have set as your wallet location.

   If your issue is still not resolved, contact My Oracle Support (MOS).
3. When should I create a Database link, and if yes, how do I do it?
   Create a Database link to connect the Atomic and Configuration Database Schemas to the Studio Database Schema if the databases are different. You must create the link in the Studio Database.

   In the following example, a link has been created from the Configuration Schema to the Atomic Schema by running the following script:

```
create public database link <studio database link> connect to <Config Schema>
identified by password using ' (DESCRIPTION = ADDRESS_LIST = (ADDRESS =
(PROTOCOL = TCP) (HOST =<host name> (PORT = <port number>)) (CONNECT_DATA =
(SERVICE_NAME = <service name>))) ';
```

```
Config Schema : <Config Schema>/password ' (DESCRIPTION = ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP) (HOST =<host name> (PORT = <port number>))
(CONNECT_DATA = (SERVICE_NAME = <service name>))) ';
```

After running the script, run the FCDM Connector and ICIJ Connector jobs.

4. Why does my installed studio setup not have any notebooks?
   Some default notebooks are ready to use when you install Compliance Studio. If you do not see any notebooks when you log in to the application, you may not be assigned any roles. Check the <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs directory to see if you have been assigned any roles, and if not, contact your Administrator. If your issue is still not resolved, contact My Oracle Support (MOS).
5. What can I do if the Schema Creation fails?
   If the Atomic Schema creation fails, login to the BD and ECM Atomic Schemas and run the following query: select * from fcc_orahive_datatypemapping; The fcc_orahive_datatypemapping table must not have duplicate data types. If the Studio schema creation fails, login as a Studio user and run the following query: select * from fcc_datastudio_schemaobjects Run the following query to replace all Y values with '': update fcc_datastudio_schemaobjects set SCHEMA_OBJ_GENERATED='' After the schema creation is successful, the value of the SCHEMA_OBJ_GENERATED attribute changes to Y. You can also check for errors in the application log file in the <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs directory. If your issue is still not resolved, contact My Oracle Support (MOS).
6. What can I do if the Import_training_model batch execution fails?
   Batch Execution Status always displays success in case of success or failure.

   You can also check for errors in the application log file in the <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs directory. You can fix the failure according to the log details and run the same batch again.
7. Why is the sqoop job not successful?
   The Sqoop job may fail if some of the applicable values are null or if the service name or SID value is not provided. Do one of the following:
   - Check if there are any null values for the applicable configurations in the config.sh and FCC_DATASTUDIO_CONFIG tables. If there are any null values, add the required value.
   - Check for any errors in the application log file in the <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs directory. If your issue is still not resolved, contact My Oracle Support (MOS).
8. Why am I getting the following error when I run the sqoop job:
   `Error: Could not find or load main class com.oracle.ofss.fccm.studio.batchclient.client.BatchExecute`

   Set the `FIC_DB_HOME` path in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb` directory.

   You can also check for any errors in the application log file in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs` directory.
9. 11. Why is the PGX server is not starting even though the graph service is up and running?
   Grant execution rights to the PGX folder to start the PGX server.
10. Why is the PGX Server not starting?

The PGX server starts only after the FCDM tables are created after the FCDM Connector Job is run. Check if all FCDM tables are created and then start the PGX Server. You can also check for any errors in the application log file in the <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs directory. If your issue is still not resolved, contact My Oracle Support (MOS).

11. Why is the ICIJ Connector job failing?
This can happen because of a missing `csv` file path in the FCC_STUDIO_ETL_FILES table. Add the `CSV` file path. You can also check for any errors in the application log file in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs` directory. If your issue is still not resolved, contact My Oracle Support (MOS).

12. What should I do if there is a below Error while selecting edges in manual Decision UI?

```
java.lang.IllegalStateException: Unable to create
PgxSessionWrapperjava.lang.IllegalStateException: Unable to create
PgxSessionWrapper at
oracle.datastudio.interpreter.pgx.CombinedPgxDriver.getOrCreateSession(Combine
dPgxDriver.java:147) at
oracle.pgx.graphviz.driver.PgxDriver.getGraph(PgxDriver.java:334) at
oracle.pgx.graphviz.library.QueryEnhancer.createEnhancer(QueryEnhancer.java:22
3) at
oracle.pgx.graphviz.library.QueryEnhancer.createEnhancer(QueryEnhancer.java:20
9) at oracle.pgx.graphviz.library.QueryEnhancer.query(QueryEnhancer.java:150)
at oracle.pgx.graphviz.library.QueryEnhancer.execute(QueryEnhancer.java:136)
at
oracle.pgx.graphviz.interpreter.PgqlInterpreter.interpret(PgqlInterpreter.java
:131) at
oracle.datastudio.interpreter.pgx.PgxInterpreter.interpret(PgxInterpreter.java
:120) at
org.apache.zeppelin.interpreter.LazyOpenInterpreter.interpret(LazyOpenInterpre
ter.java:103) at
org.apache.zeppelin.interpreter.remote.RemoteInterpreterServer$InterpretJob.jo
bRun(RemoteInterpreterServer.java:632) at
org.apache.zeppelin.scheduler.Job.run(Job.java:188) at
org.apache.zeppelin.scheduler.FIFOScheduler$1.run(FIFOScheduler.java:140) at
java.base/
java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:515) at
java.base/java.util.concurrent.FutureTask.run(FutureTask.java:264) at
java.base/
java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Sched
uledThreadPoolExecutor.java:304) at java.base/
java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1128
) at java.base/
java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:628
) at java.base/java.lang.Thread.run(Thread.java:834)Caused by:
java.util.concurrent.ExecutionException:
oracle.pgx.common.auth.AuthorizationException: PgxUser(FCCMDSADMIN) does not
own session 6007f00a-8305-4576-9a56-9fa0f061586f or the session does not exist
code: PGX-ERROR-CQAZPV67UM4H at java.base/
java.util.concurrent.CompletableFuture.reportGet(CompletableFuture.java:395)
at java.base/
java.util.concurrent.CompletableFuture.get(CompletableFuture.java:1999) at
oracle.pgx.api.PgxFuture.get(PgxFuture.java:99) at
oracle.pgx.api.ServerInstance.getSession(ServerInstance.java:670)
oracle.datastudio.interpreter.pgx.CombinedPgxDriver.getOrCreateSession(Combine
dPgxDriver.java:145) ... 17 moreCaused by:
```

```
oracle.pgx.common.auth.AuthorizationException: PgxUser(FCCMDSADMIN) does not
own session 6007f00a-8305-4576-9a56-9fa0f061586f or the session does not exist
code: PGX-ERROR-CQAZPV67UM4H at
oracle.pgx.common.marshalers.ExceptionMarshaler.toUnserializedException(Except
ionMarshaler.java:107) at
oracle.pgx.common.marshalers.ExceptionMarshaler.unmarshal(ExceptionMarshaler.j
ava:123) at
oracle.pgx.client.RemoteUtils.parseExceptionalResponse(RemoteUtils.java:130)
at
oracle.pgx.client.HttpRequestExecutor.executeRequest(HttpRequestExecutor.java:
198) at
oracle.pgx.client.HttpRequestExecutor.get(HttpRequestExecutor.java:165) at
oracle.pgx.client.RemoteControlImpl$10.request(RemoteControlImpl.java:313) at
oracle.pgx.client.RemoteControlImpl$ControlRequest.request(RemoteControlImpl.j
ava:119) at
oracle.pgx.client.RemoteControlImpl$ControlRequest.request(RemoteControlImpl.j
ava:110) at
oracle.pgx.client.AbstractAsyncRequest.execute(AbstractAsyncRequest.java:47)
at oracle.pgx.client.RemoteControlImpl.request(RemoteControlImpl.java:107) at
oracle.pgx.client.RemoteControlImpl.getSessionInfo(RemoteControlImpl.java:296)
at
oracle.pgx.api.ServerInstance.lambda$getSessionInfoAsync$14(ServerInstance.jav
a:490) at java.base/
java.util.concurrent.CompletableFuture.uniComposeStage(CompletableFuture.java:
1106) at java.base/
java.util.concurrent.CompletableFuture.thenCompose(CompletableFuture.java:2235
) at oracle.pgx.api.PgxFuture.thenCompose(PgxFuture.java:158)
```

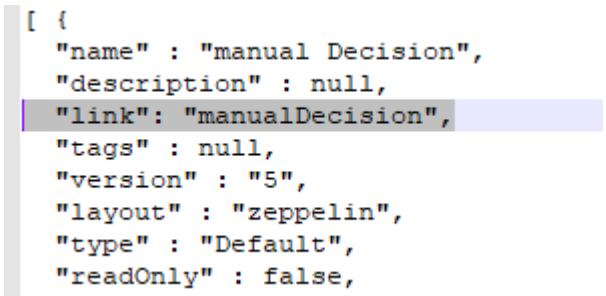
Then, perform the below steps as a workaround -

Export the "Manual Decision" Notebook

Add the link parameter just below Description

for Ex - "link": "manualDecision",

Figure 10-1 Manual Decision

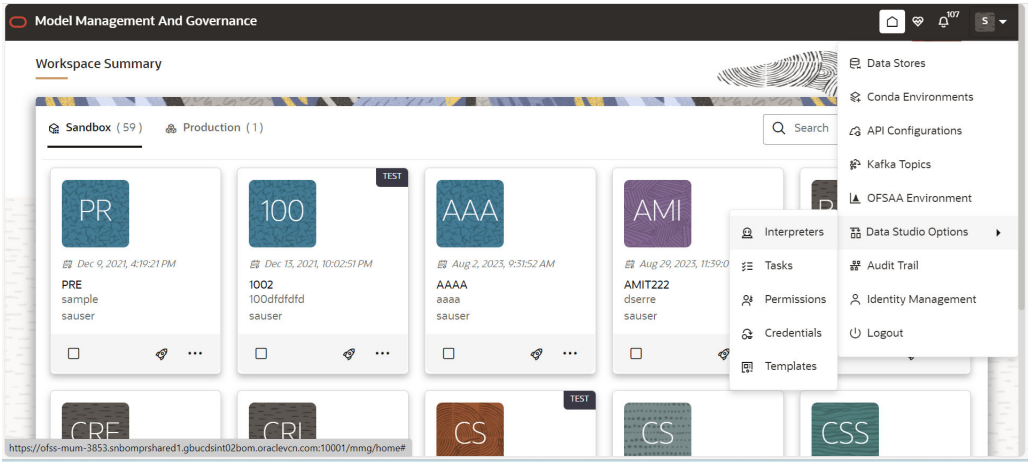


Truncate the table “fcc_er_paragraph_manual" in Studio Schema. Import the modified notebook again.

13. Data Extraction is truncated to default limit (Approx 197 records) in Python paragraph widget output in MMG. **Setting the ZEPPELIN_LIMIT_INTERPETER_OUTPUT in Python Interpreter**

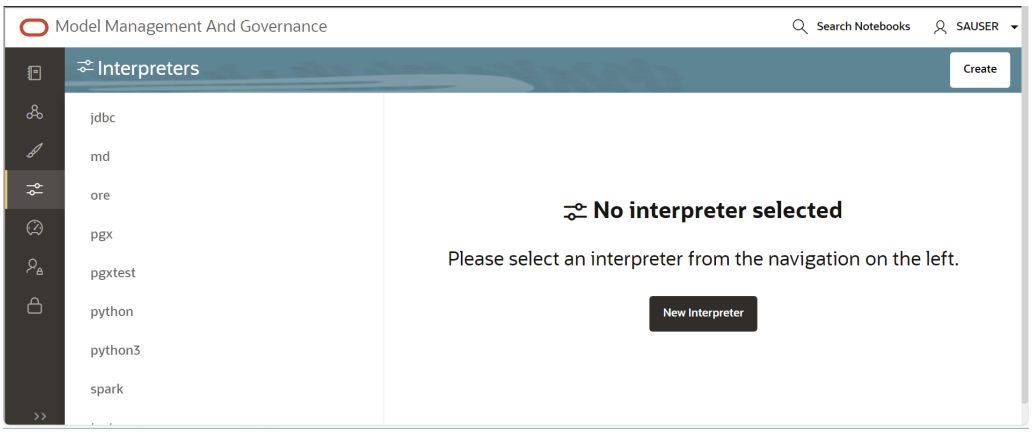
    a. From UI: Using Wizard screen
    Go to Interpreters screen in MMG-Studio from Datastudio Options tab.

Figure 10-2 Datastudio Options tab



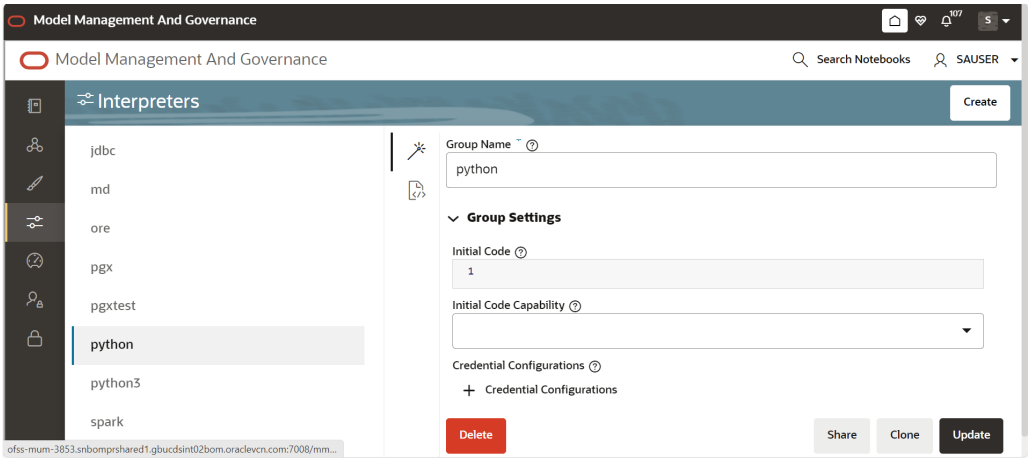
b. Once on the interpreters option screen select the Python Interpreter for which we want to configure the zeppelin.limit.interpreter.output.

Figure 10-3 Interpreter screen



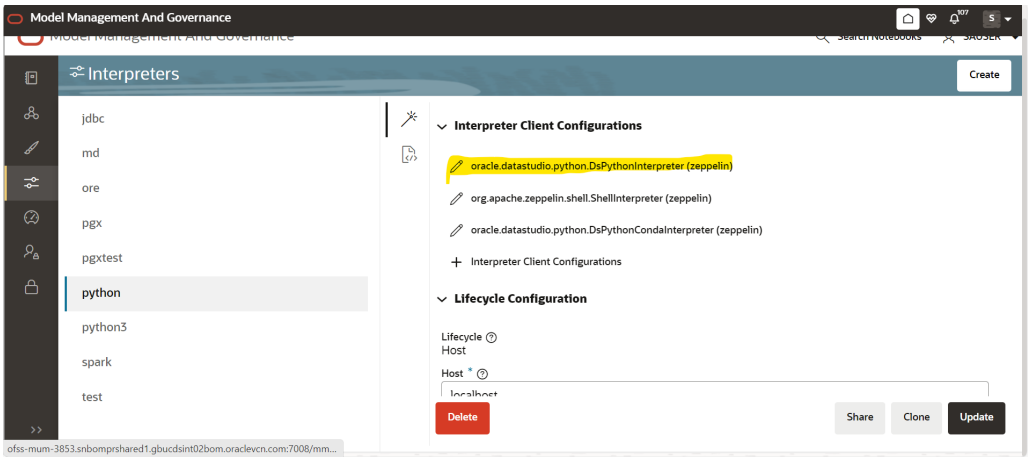
c. Select python from the LHS options.

Figure 10-4 Python Interpreter



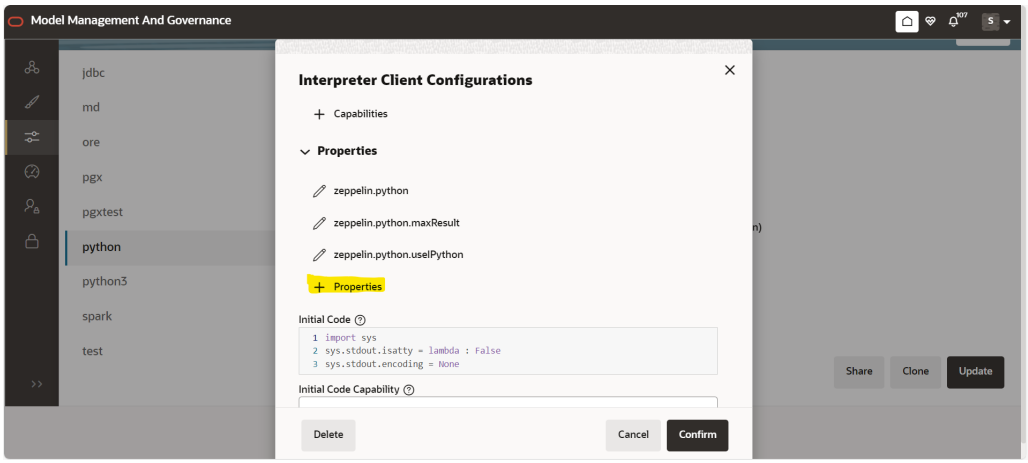
d. Now scroll down in the RHS side and click on the oracle.datastudio.python.DsPythonInterpreter under Interpreter Client Configurations it will open a popup.

Figure 10-5 Interpreter Client Configurations



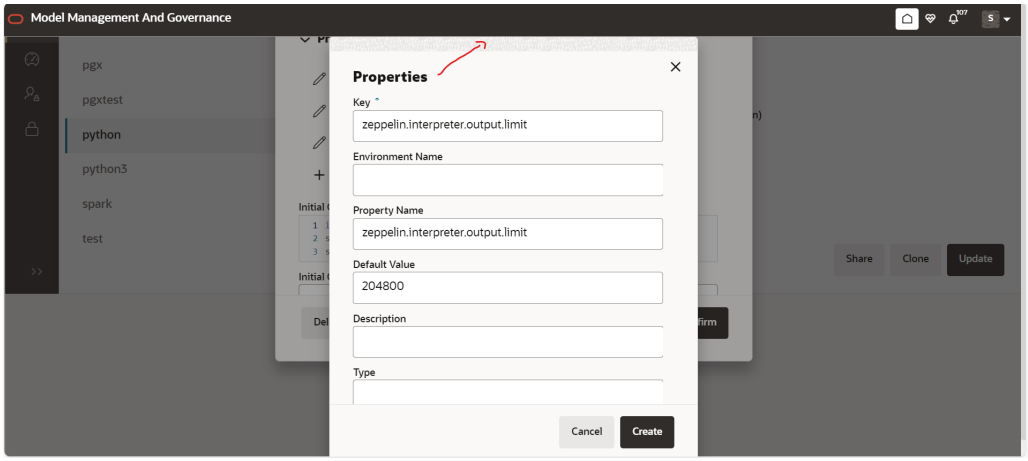
e. In the popup scroll down and click on + Properties under Properties as shown:

Figure 10-6 Properties screen



f. Another popup will open fill the options as shown and set the default value according to your needs if you are not able to see the Create and Cancel button, click on the part of the popup pointed by red arrow. The default value for zeppelin.interpreter.output.limit if not set is 102400 (in bytes).

Figure 10-7 Popup box



> **Note**
>
> Increasing the default option from 102400 to some bigger value will slow down the rendering of outputs of python paragraphs.

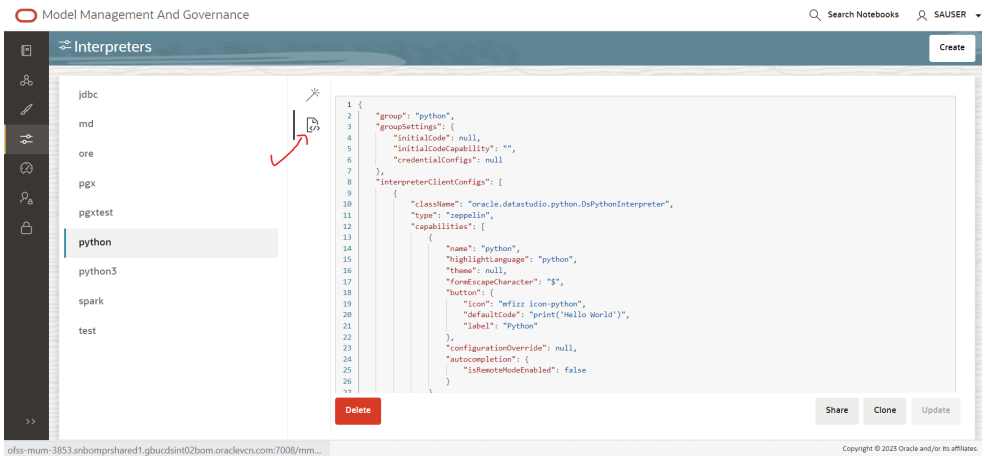
g. Once filled click on Create (you will see zeppelin.interpreter.output.limit under the Properties section), then click on **Confirm** (if you are not able to see the Confirm button on the UI, either click on the same shaded area on popup as highlighted in above image or zoom out in UI of browser) and then click on Update in the lower right side of the screen.

h. After following all the above steps, restart the MMG-Studio for changes to reflect.

a. Using JSON screen

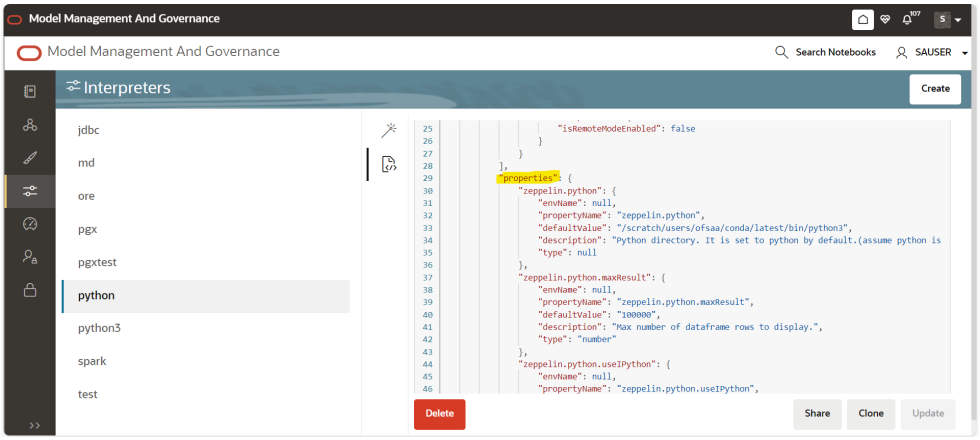
i. Follow the steps i, ii and iii from above, then click on the following icon on UI pointed by red arrow and following json config view will open.

Figure 10-8 JSON Config View



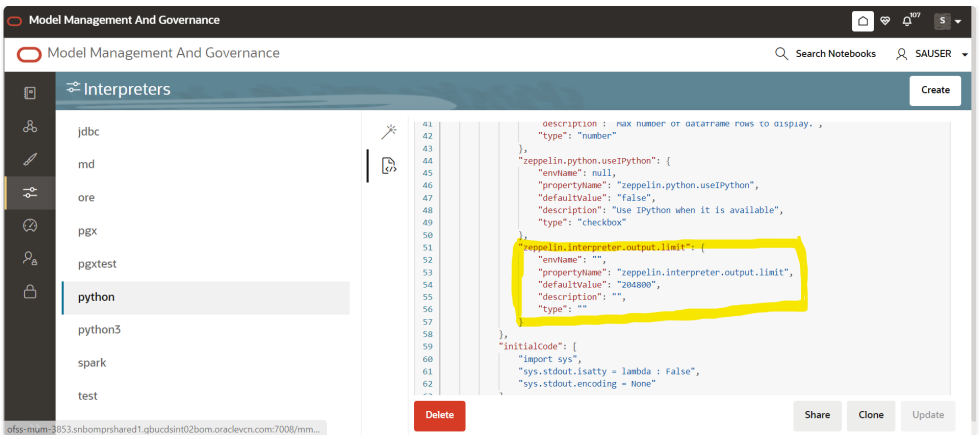
ii. Scroll down under interpreterClientConfigs with className oracle.datastudio.python.DsPythonInterpreter you will find following properties section with bunch of zeppelin configurations.

Figure 10-9 Interpreter Client Config



iii. After the last entry in properties add the zeppelin.interpreter.output.limit also as shown in the following image:

Figure 10-10 Properties screen



iv. After doing the change the Update button will get enabled in the bottom right corner click on it, you will get a message as “python interpreter updated”.

v. Now restart the MMG-Studio service for changes to reflect.

b. From filesystem: (Datastudio version 23.4.x onwards)

i. Go to the Python Interpreter option as pointed out in From UI using wizard screen option above, if you have already ran the MMG services before you will see the python interpreter listed there. Delete it, if you are running the MMG Application for the first time on a fresh schema then you do not need to do this step.

ii. After deleting the Python Interpreter or if start has not been done yet, go to filesystem inside mmg-home/mmg-studio/server/builtin/interpreters, open python.json in a text editor.

iii. Scroll down under interpreter ClientConfigs with className oracle.datastudio.python.DsPythonInterpreter you will find following properties section with bunch of Zeppelin configurations. After the last entry in properties add the Zeppelin Interpreter.output.limit also as shown in step iii) of 1) From UI b) using JSON screen (last image of From UI way). Save the python.json with the desired default value and the changes done.

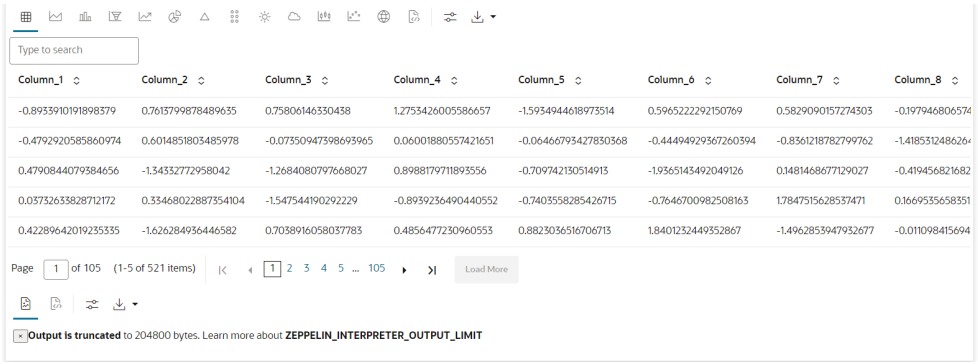
iv. Now restart/start the MMG-Studio for your changes to reflect.

> **Note**
>
> If you have configured the python environment for MMG-Studio (basically you have installed pandas and numpy which are subset of libraries required by MMG as pre-req), you can run the below script on python paragraph.

```
%python
import pandas as pd
import numpy as np
# Create 1000 rows of random data for 20 columns
data = np.random.randn(1000, 20)
# Create column names columns = [f"Column_{i+1}" for i in range(20)]
# Create DataFrame df = pd.DataFrame(data, columns=columns)
# Display the DataFrame
z.show(df)
```

Output in table view

Figure 10-11 Output in table view



You can see the ZEPPELIN_INTERPRETER_OUTPUT_LIMIT value as warning if the table content is more than the set default value for zeppelin.interpreter.output.limit and accordingly you can modify the default value for same.

14. What should I do when the result set is truncated if the size goes above '102400' bytes? Perform the following steps:
    a. Login to Compliance Studio.
    b. Navigate to interpreter zeppelin.interpreter.output.limit.

Figure 10-12 Zeppelin Interpreter

c. Set the value to the required size.

d. Restart the Studio Application.

15. What should I do if there is a below KubernetesClientException in load-to-elastic-search.log, matching-service.log files after Compliance Studio installation?

```
configServicePropertySourceLocator - Could not locate PropertySource: I/O
error on GET request for "http://localhost:8888/<Service Name>/default":
Connection refused (Connection refused); nested exception is
java.net.ConnectException: Connection refused (Connection
refused)onfigServicePropertySourceLocator - Could not locate PropertySource:
I/O error on GET request for "http://localhost:8888/<Service Name>/default":
Connection refused (Connection refused); nested exception is
java.net.ConnectException: Connection refused (Connection refused)20:04:55.686
[ main] WARN .cloud.kubernetes.config.ConfigMapPropertySource - Can't read
configMap with name: [<Service Name>] in namespace:[null].
Ignoring.io.fabric8.kubernetes.client.KubernetesClientException: Operation:
[get] for kind: [ConfigMap] with name: [<Service Name>] in namespace: [null]
failed. at
io.fabric8.kubernetes.client.KubernetesClientException.launderThrowable(Kubern
etesClientException.java:64) ~[kubernetes-client-4.4.1.jar!/:?] at
io.fabric8.kubernetes.client.KubernetesClientException.launderThrowable(Kubern
etesClientException.java:72) ~[kubernetes-client-4.4.1.jar!/:?] at
io.fabric8.kubernetes.client.dsl.base.BaseOperation.getMandatory(BaseOperation
.java:229) ~[kubernetes-client-4.4.1.jar!/:?] at
io.fabric8.kubernetes.client.dsl.base.BaseOperation.get(BaseOperation.java:162
) ~[kubernetes-client-4.4.1.jar!/:?] at
org.springframework.cloud.kubernetes.config.ConfigMapPropertySource.getData(Co
nfigMapPropertySource.java:96) ~[spring-cloud-kubernetes-
config-1.1.3.RELEASE.jar!/:1.1.3.
```

You can ignore the error when the following message is displayed at the end of the log; if you do not see this message, contact My Oracle Support (MOS) and provide the applicable error code and log:

```
13:52:57.698 [main] INFO org.apache.catalina.core.StandardService - Starting
service [Tomcat] 13:52:57.699 [ main] INFO
org.apache.catalina.core.StandardEngine - Starting Servlet engine: [Apache
Tomcat/9.0.43]
```

16. What happens if a new sandbox workspace is created?
When a new sandbox workspace is created, the folders of the older workspace are by default being copied into the new workspace. Here, folder means the Model Objectives. The Model Objectives are global objects and will be visible across the workspaces. However, the models created within those objectives will be private. This has been done purposely as you expect multiple modelers working on the common objective in their private workspaces.

17. Not able to access any models in the copied folders in the new workspace – the folders are being copied as empty folders?
    Yes, you should not be able to access other workspace's private models. Also, as long as other users are working on the objective and have their models in there, you will not be able to delete the objectives.
18. What should you do when UI pages does not load due to less network speed?
    The default time to load all the modules of OJET/REDWOOD page is 1 minute. Reload the page to view the UI pages.
19. What are the Workspace parameters used in MMG Python Scripts?
    The following parameters are used:
    - **workspace.list_workspaces():** Used to fetch a list of all workspaces. This list is populated in the dropdown menu of datastudio.
    - **workspace.check_aif():** A method used to check if AIF is enabled or not
    - **workspace.attach_workspace("SANDBOX123"):** A method used to set workspace
    - **workspace.get_workspace():** Used to fetch the selected workspace (for example, SB1)
    - **get_mmg_studio_service_url():** Used to fetch the base URL (for example, http://whf999yyy:0000/mmg)
    - **get_user():** Used to fetch current user (for example, mmguser)
20. How to take connections for Data access?
    You need access to the data to work on it. For the workspace, there are some underlying Data Schemas. You can also create a workspace that allows to select multiple underlying Data Schemas. You can use or remove multiple Data Schemas like multi combo box, where 1, 2, 3, and 4, 5 are schemas underlying. When you work with the models, you can access the notebook to fetch data for all these Data Schemas and create some data frames out of it. That can be used for model reading or other purposes.

    This happens in workspace of the sandbox where you are building a Notebook. The same Notebooks gets promoted to production workspace. Therefore, the workspace production has its own set of underlying Data Schemas. When you build the model with getting connection for the underlying Schema 1 and 2, and getting the data and building, it makes rules work and will not be affected if the same Notebooks gets promoted to production or deployment is cloned.

    Therefore, the Notebook needs to run which should not be fetching this data because it will be working on any 1 and 2 Schemas.

    To avoid this issue, you can use connection feature to connect with a schema. This is a wrapper function where you can specify which workspace you are connecting to.

    You can enter the workspace details to get the connection and that starts fetching the data.

    When you create the Notebook to production, a script runs to not to connect the workspace. This also uses overloaded methods. This method tells how to get the connection. Simple get connection gets the primary connection as first Data Schema which you are using without any overload.

    The second connection gets an ID as the name the Data Source which you are using and for the current one will passes as get connection 1.

    In the sandbox, this script looks for 1 and it creates a connection and moves to production.

    It will again look for an equivalent 1 and tries to get a connection.

    Therefore, whatever you select first, becomes the first Data Schema, Second Schema, Third Schema, therefore, Primary, Secondary, Tertiary and so on. You can also pass the

number while getting the connection to get the first primary Data Schema as a secondary Data Schema. Therefore, when it runs in sandbox, it gets the Secondary Schema. When it runs in the production, it fetches a Secondary Data Schema of production.

21. What are parameters to establish the Connection for data access?
The following section lists the connection details such as the Data Sources and so on: workspace.get_connection(): fetches connection object for the Primary Data Source of the workspace. This is equivalent to executing workspace.get_connection(1). workspace.get_connection('id'): fetches connection for the Data Source by name. For example, workspace.getconnection ('ws_data_1') – here 'ws_data_1' is one of the underlying Data Source for the workspace. workspace.get_connection(n): fetches connection for the Data Source by order. For example, workspace.getconnection(2) – this will fetch connection for the Secondary Data Source. The following section lists the workspace details: After a workspace is attached, we can list Data Sources related to that using: workspace.list_datasources(): will list Data Sources related to attached workspace with default order 1 For example, {'Data Source': [{'name': 'newdatasource1', 'order': '1'}]} workspace.list_datasources("SB1"): will list Data Sources related to SB1 workspace with default order 1 For example, {'Data Source': [{'name': 'ds1', 'order': '1'}]} workspace.list_datasources("SB1", 1): will list Data Sources related to SB1 workspace with order 1 as passed in second argument For example, {'Data Source': [{'name': 'ds1', 'order': '1'}]} Note: This is applicable for Python and Python variants interpreters, and not on any other interpreters.

22. What should I do if the Python installation displays the following error message, " If ModuleNotFoundError: No module named '_lzma'"?
You must install xz-devel library before installing the Python. For more details, see Install MMG Python Library section.

    To install, perform the following step:

    ```
    $yum install -y xz-devel.
    ```

23. What should I do to reconfigure DS Studio server port and its interpreter's default port to available ports?
To reconfigure port numbers:

    a. Run the command `install.sh -u` to change the current studio port to the desired port number in the configuration files/tables.

    b. Run the t `startup.sh` script of Studio at the location: OFS_MMG/mmg-studio/bin/ and modify the line numbers 24/25 of OFS-MMG/mmg-studio/bin/startup.sh to specify the interpreter name and port number.
    DS version 22.4.3

    nohup "$DIR"/datastudio --jdbc -1 --eventjdbc -1 --shell -1 --eventshell -1 --graalvm -1 --eventgraalvm -1 --pgx -1 --eventpgx -1 --external --port 8008 --jdbc 3011 --eventjdbc 3031 --python 3012 --eventpython 3032 --markdown 3009 --eventmarkdown 3029 --spark 3014 --eventspark 3034 &> "$DIR"/nohup.out &

    For PGX Interpreter, modify: OFS_MMG/mmg-studio/interpreter-server/pgx-interpreter-22.4.3/bin/pgx-interpreter file "${1:-7022}" "${2:-7042}" values to "${1:-3022}" "${2:-3042}"

    DS version 23.3.5

    nohup "$DIR"/datastudio --jdbc -1 --shell -1 --external --port 8008 --jdbc 3011 --python 3012 --markdown 3009 --spark 3014 --pgx 3022 &> "$DIR"/nohup.out

    For event ports in DS 23.3.5

Set the environment variables DS_EVENT_HANDLER_HOST and DS_EVENT_HANDLER_PORT before launching the interpreters, else, default values will be used. You can modify these ports in the startup.sh of the Studio.

Example:

export DS_EVENT_HANDLER_HOST=localhost

export DS_EVENT_HANDLER_PORT=3432

To change the ports configured for events in the Data Studio server, modify the following server configuration:

studio-server:

thrift-server:

enabled: true

port: <desired port -defaulted to 8432>

mode: TCP

NOTE:

****

Python Interpreter

Beginning with Data Studio 21.4.0, 6012 is default port on which the REST server for the Python interpreter listens. To overwrite this, set the STUDIO_INTERPRETER_PYTHON_INTERPRETER_REST_SERVER_PORT environment variable.

PGX-Python Interpreter

Beginning with Data Studio 23.1.0, 6022 is the default port on which the REST server for the PGX-Python interpreter listens. To overwrite this, set the STUDIO_INTERPRETER_PGX_PYTHON_INTERPRETER_REST_SERVER_PORT environment variable.

Modify the `startup.sh` to:

export STUDIO_INTERPRETER_PYTHON_INTERPRETER_REST_SERVER_PORT=3038

export STUDIO_INTERPRETER_PGX_PYTHON_INTERPRETER_REST_SERVER_PORT=3039

This configuration changes the default interpreter ports to new ports.

c. Ports mentioned in the interpreter json files should be reconfigured. The interpreter file location is: “OFS_MMG/mmg-studio/server/builtin/interpreters/<interpreter>.json” file.

d. Execute startup.sh and check the studio/interpreter ports.

e. Similarly, execute ./datastudio.sh –help from OFS_MMG/mmg-studio/bin/ for all available options.

DS Studio Server port and its interpreters default port can be reconfigured to any available ports by following these steps:

a. Change the Datastudio URL with the desired `DS port. install.sh -u` must be triggered to change the current studio port to 8008 in the configuration files/tables.

b. After successful execution of install.sh. The ports can be updated by the user in the startup.sh of studio in the path `OFS_MMG/mmg-studio/bin/`.

i. Edit line no 24/25 of `OFS-MMG/mmg-studio/bin/startup.sh` and change as below by specifying the interpreter name and port to be modified.

- **In DS version 22.4.3**
  nohup "$DIR"/datastudio --jdbc -1 --eventjdbc -1 --shell -1 --eventshell -1 --graalvm -1 --eventgraalvm -1 --pgx -1 --eventpgx -1 --external --port 8008 --jdbc 3011 --eventjdbc 3031 --python 3012 --eventpython 3032 --markdown 3009 --eventmarkdown 3029 --spark 3014 --eventspark 3034 &> "$DIR"/nohup.out

  For PGX Interpreter

  Change it directly in the OFS_MMG/mmg-studio/interpreter-server/pgx-interpreter-22.4.3/bin/pgx-interpreter file "${1:-7022}" "${2:-7042}" values to "${1:-3022}" "${2:-3042}"

- **In DS version 23.3.5**
  nohup "$DIR"/datastudio --jdbc -1 --shell -1 --external --port 8008 --jdbc 3011 --python 3012 --markdown 3009 --spark 3014 --pgx 3022 &> "$DIR"/nohup.out & For event ports in DS 23.3.5 You need to set the environment variables DS_EVENT_HANDLER_HOST and DS_EVENT_HANDLER_PORT before interpreters are launched. Otherwise, the default values are used. This can be mentioned in the `startup.sh` of studio. example:

  – export DS_EVENT_HANDLER_HOST=localhost

  – export DS_EVENT_HANDLER_PORT=3432

  In order to change the port listening for events in the Data Studio server, adapt following server configuration:

  – studio-server:

    * thrift-server:

      * enabled: true

      * port: <desired port -defaulted to 8432>

      * mode: TCP

- **Generic Notes**
  Python Interpreter

  Starting from Data Studio 21.4.0, the REST server for the Python interpreter listens on port 6012 by default. One can overwrite this by setting the STUDIO_INTERPRETER_PYTHON_INTERPRETER_REST_SERVER_PORT environment variable.

  PGX-Python Interpreter

  Starting from Data Studio 23.1.0, the REST server for the PGX-Python interpreter listens on port 6022 by default. One can overwrite this by setting the STUDIO_INTERPRETER_PGX_PYTHON_INTERPRETER_REST_SERVER_PORT environment variable.

  The above can be mentioned in the `startup.sh` of studio as export STUDIO_INTERPRETER_PYTHON_INTERPRETER_REST_SERVER_PORT=3038 export STUDIO_INTERPRETER_PGX_PYTHON_INTERPRETER_REST_SERVER_PORT=3039

The above configuration will change the default interpreter ports and reconfigure to listen to the new ports. (For example: As mentioned in the below table).

ii. Ports mentioned in the interpreter json files also needs to be reconfigured. The interpreter file locations can be found at “OFS_MMG/mmg-studio/server/builtin/interpreters/<interpreter>.json” file.

iii. Execute startup.sh and check the studio/interpreter ports.

iv. Similarly you can execute as ./datastudio.sh –help from OFS_MMG/mmg-studio/bin/ location for all the available options.

The above steps will reconfigure Server/Interpreter to these ports:

- Server/Interpreters Modified Port
- DS Studio port 8008
- Jdbc 3011
- eventjdbc 3031
- python 3012
- eventpython 3032
- markdown 3009
- eventmarkdown 3029
- spark 3014
- eventspark 3034
- pgx 3022
- eventpgx 3042

24. Dataset issue with the latest version of pydantic package (2.18.7)
Pydantic package (2.18.7) is incompatibile with MMG functionality. So, when you reinstall the package the uninstall and reinstall pydantic package version 1.10.13.
    - python3 -m pip uninstall pydantic
    - python3 -m pip install pydantic==1.10.13 --user
25. Installation of Python Packages from Local Repository
In order to install the python dependencies in mmg-8.1.2.6.0.tar.gz from a local repository, use the following command.python3 -m pip install mmg-8.1.2.6.0.tar.gz.tar.gz --index-url http://artifactory.XYZ.com/artifactory/api/pypi/XYZ-py-local/simple --extra-index-url http://artifactory.XYZ.com/artifactory/api/pypi/XYZ-py-local/simple --trusted-host artifactory.XYZ.com
26. MMG Configuration steps for Interpreters
    a. **For JDBC**

    Update the below proeprty in jdbc.json underOFS_MMG/mmg-studio/server/builtin/interpreters

    "propertyName": "default.url"

    "defaultValue": "<JDBC_URL>"

    For example: jdbc:oracle:thin:@ofss-mum-1033.snbomprshared1.gbucdsint02bom.oraclevcn.com:15 21/MMG19PDB

    "propertyName": "default.user",

"defaultValue": "<schameusername>"

For example: The schema user to which you want to connect, for example: datastudio schema name,

"propertyName": "default.password",

"defaultValue": "<schemapassword>"

For example: Password of the provided schema user.

Start the jdbc interpreter by executing below command under //OFS_MMG/mmg-studio/interpreter-server/jdbc-interpreter-22.4.3/bin ./

jdbc-interpreter

If the jdbc interpreter needs to be included in the datastudio startup script remove the below entry from /OFS_MMG/mmg-studio/bin/startup.sh --jdbc -1.

For Spark

Configuration with Kerberos enabled remote spark cluster:

a. Copy the configured Spark directory from hadoop cluster to <MMG Studio>/interpreter-server/spark-interpreter/extralibs. For example: spark-2.4.8-bin-hadoop2.7

b. Copy the below files to the <MMG Studio>/interpreter-server/spark-interpreter/extralibs krb5.conf <keytabfile>.keytab

c. To run Spark in yarn-client mode, configure the following parameters in this file OFS_MMG/mmg-studio/server/builtin/interpreters/spark.json
spark.master = yarn-client

spark.driver.bindAddress = 0.0.0.0

spark.driver.host = <host> -> Apache Spark host

> **Note**
>
> When using the Kubernetes interpreter lifecycle, <host> can be the IP address or hostname of any node in your Kubernetes cluster. When using the Host interpreter lifecycle, <host> should be the IP address or hostname of the node that runs the Spark interpreter.

> **Note**
>
> When connecting to a YARN cluster, the Spark driver authenticates as the UNIX user that runs the Spark interpreter. You can set the HADOOP_USER_NAME environment variable to make the Spark driver authenticate as a different user. If you use the Host interpreter lifecycle, then you can do this by exporting the HADOOP_USER_NAME environment variable before starting the Spark interpreter process. If you us the Kubernetes interpreter lifecycle, then you can do this by setting the HADOOP_USER_NAME environment variable in the resource manifest (spark.yml).

d. Update file spark-defaults.conf keytab location to the location where <keytabfile>.keytab file is copied

e. Update file spark-env.sh with the krb5.conf location to the location where krb5.conf file is copied.

For example: Djava.security.krb5.conf=/OFS_MMG/mmg-studio/interpreter-server/spark-inte rpreter-22.4.2/extralibs/krb5.conf".

27. MMG SAAS - OFS_SRV_ACCT USER is seeded in the User Profile tables:
OFS_SRV_ACCT user is available in the following tables as a seeded user:
    - aaicl_sms_user_profile
    - aaicl_sms_user_profile_u

Application Pack 8.1.3.3.0 FAQs

1. **The the cx_Oracle connection is failing in DS with the following error in OEL 8.**
Fail to execute line 4: cx_Oracle.connect(dsn=dsn_alias)\nTraceback (most recent call last):\n File "/tmp/1638454321889-0/zeppelin_python.py", line 163, in <module>\n exec(code, _zcUserQueryNameSpace)\n File "<stdin>", line 4, in <module>\ncx_Oracle.DatabaseError: DPI-1047: Cannot locate a 64-bit Oracle Client library: "libnsl.so.1: cannot open shared object file: No such file or directory". Install the libnsl package as below: yum install libnsl or sudo yum install libnsl
2. **Python Interpreter fails with the `py4j` Error**
When running interpreters locally, they assume all the dependencies to be already installed and available. Python Interpreter needs `py4j` Package, exact steps to install it depend on the Operating System. If you use `pip`, it can be done with ```bash pip install --user py4j ``` Install the package for all users, root user can run this command without `--user`.
3. **What is the reason for the http error code 401 when I successfully log in to the MMG Application while MMG Studio is down?**
If MMG Studio is not up during the MMG application login, the mmg-ui logs capture the http error code : 401 . Since the cookie creation is done during MMG application login, the user must re login to the MMG application once the Studio is up and running.
4. **What should I do when the following error message is displayed, and the SSL module is unavailable for Linux 8?**
urllib3.exceptions.SSLError: Can't connect to HTTPS URL because the SSL module is not available. During handling of the above expectation, another exception occurred:
    a. Install the compat-openssl10 module on Linux 8.
    b. Log in to the server as a root user where MMG Application is installed.
    c. Run the following Shell command: yum -y install compat-openssl10.
5. **Why do multiple commits/versions appear in my remote Git repository after my first push, even though I only pushed once?**
When you push a local git repository to a remote one for the first time, all the commits that exist in your local repository's history for the branch that is being pushed are transferred to the remote repository. This happens even if the remote is new or empty. This is standard git behavior. If your local repository was initialized earlier, or was linked to a different remote connection, then all its' commits will still remain in your local git history. When you add a new remote connection and perform a push action, git pushes the entire commit history, not just the most recent change. So the remote connection will reflect all your local commits.

    Key Points:

    - Git tracks the full commit history locally, regardless of remote connections.
    - Pushing to a new remote connection brings the entire history of your pushed branch.
    - If you want only specific commits in the new remote connection, you must create a new repository or filter your history before pushing.