Oracle Financial Services Compliance Studio

Administration and Configuration Guide

Release 8.1.2.6.0

July 2024

F48792-01

ORACLE
Financial Services

ORACLE

OFS Compliance Studio Administration and Configuration Guide

and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

For information on third party licenses, click here.

Document Control

Table 1 lists the document control of this guide.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.6.12	July 2024	Added the Configure Custom Notebook in ECM section.
8.1.2.6.11	May 2024	Added a grant in the Fine Grain Data Access Control for Workspace section.
8.1.2.6.10	May 2024	Added the following sections: • Create a Model Group with a Single Data Segment • Create a Model Group with Multiple Data Segments • Task: Aggregate_Base_Features for Additional Segments
8.1.2.6.9	April 2024	Added the following sections: • Specify Unique Model Information • Adding User Defined Transformation (UDT) as Python Module • Task: Output Overlays
8.1.2.6.3	March 2024	Added the Scenario Model Batch Framework section. Added the Model Groups for Behavioral Scenario Model section. Added "Scenario Model" in the For ML and Typology Use Case section. Added step 4 in the Data Sourcing section. Added "importNotebookSM.sh" utility in the Importing Workspace Metadata for ML4AML section.
8.1.2.6.2	January 2024	Added step 3 in the Resetting Graph Pipeline Back to Day 0 section.
8.1.2.6.0	December 2023	Added the Custom Rulesets for Matching section.
8.1.2.6.0	November 2023	Updated CSA_8126 pipeline details in the following sections: • Pre-configured Rulesets for Matching, Merging, and Data Survival • Pre-configured Entity Resolution Pipelines • Creating Pre-Staging Tables in FSDF Added the following sections: • Removal of Entities from the Global Party (Deleted Party) • Ability to Remove Split and Merge Manually • Conda Environment in Notebook • Parameters for Entity Resolution Job execution • Python Libraries for Predefined Conda Environment Added F_OVERRIDE_FLAG in the Table 16.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.6.0	November 2023	Added F_ER_DS_SUBSEQUENT_BATCH, MAX_HISTORY_PARTITIONS and ER_DS_SYSTEM_PENDING_MAX_NO_REC parameters in the Additional Configurations section. Updated fcc_er_guid_persist_config table in the Properties for Global Party ID Persistence section. The fcc-python interpreter has been modified to python interpreter and corresponding changes are done in Table 8 and python Interpreter section.
8.1.2.6.0	November 2023	Added a new note for JDBC interpreter in the Table 8 and jdbc Interpreter section. The Configure ETL and Execute ETL sections are deprecated as these apply to legacy ETL which has been replaced with graph pipeline functionality. The "ML Name and Address Model Training" section has been deprecated for legacy ETL, and this similar functionality will be added in graph pipelines in future releases. The "Create and Execute a Run Executable for Scenario Notebooks" section has been deprecated as this functionality is replaced by model execution. The "PGX Advanced Configurations" section has been deprecated as this functionality is replaced by subgraph loading. Updated steps in the Creating Data Store section. Updated note and table in the Importing Workspace Metadata for ML4AML section.
8.1.2.5.1	September 2023	Added `enableVPD.sh` in the Table 22. Added a new Fine Grain Data Access Control for Workspace section.
8.1.2.5.0	July 2023	Added the following sections: • Setting Memory of Entity Resolution and Matching Services • Persisting the Data When F_PERSIST_GUID Flag is Set to True and F_MANUAL_APPROVAL Flag is Set to True/False Condition • Properties for Global Party ID Persistence Updated CSA_8125 pipeline details in the following sections: • Pre-configured Rulesets for Matching, Merging, and Data Survival • Pre-configured Entity Resolution Pipelines • Creating Pre-Staging Tables in FSDF

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.5.0	July 2023	Added "STG_DELETED_PARTIES_PRE" and "D_ADDRESS_END_DATE" functionalities in the Load Data into Pre-Staging Tables section. The Step 2 is updated related to command and new parameters such as Execution Mode, Current Runskey, and Run_Type in the following sections: • Steps (Create Index and Load the Data) • Steps (Perform Matching) • Steps (Data Survival) • Steps (Load Data in FCC_ER_OUTPUT Table) • Steps (Using Wrapper Shell Script)
8.1.2.5.0	July 2023	Added a note in the Profiler Table section. Updated cleanup steps in the following sections: • Cleanup Steps for Job Termination (Create Index and Load the Data) • Cleanup Steps for Job Termination (Perform Matching) • Cleanup Steps for Job termination (Data Survival) • Cleanup Steps for Job termination (Load Data in FCC_ER_OUTPUT Table) Added a note in the Persisting the Data When F_PERSIST_GUID and F_MANUAL_APPROVAL Flags are Set to False Condition section. The support for Legacy ETL is deprecated in the current release, and the related note is added in the required sections. Added a new note and step 1 in the Cleanup Steps When the Create Index and Load Data Job Terminated Manually, Cleanup Steps When the Data Survival Job Terminated Manually, Cleanup Steps When the Bulk Similarity Job Terminated Manually, Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually and Resetting Entity Resolution Back to Day 0 sections. Added the following sections in **M4AML** section: • Optimizing SQL performance for ASC • Feature Contributions JSON Format
8.1.2.4.4	June 2023	Updated steps in the Data Sourcing section. Updated steps and figure in the Configure a jdbc Interpreter Variant section.
8.1.2.4.3	May 2023	Updated latest ICIJ table list and steps in the Load Data into ICIJ Tables section.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.4.0	April 2023	Removed "DSADMIN" and updated DSUSRGRP group description in the Table 5. Updated steps in the Resetting Graph Pipeline Back to Day 0 section.
8.1.2.4.0	March 2023	Updated document to reflect that OpenSearch has replaced Elastic Search in 8.1.2.4.0. Updated steps in the following sections: • Configure a jdbc Interpreter Variant • Create a Credential • Link Credentials Added the following sections: • Migrating the Data from ElasticSearch to Open-Search • FCCM out-of-the-box Entity Resolution Pipeline on FSDF • Post Workspace Activity for ASC • Periodic Workspace Schema Cleanup for ASC • Schema Grants for AML Event Scoring Updated version for python-ml4aml interpreter in the python Interpreter section. Added utilities in the Importing Workspace Metadata for ML4AML section. Updated SAML Attribute Configuration in the Access Compliance Studio Using SAML Realm section.
8.1.2.3.0	January 2023	Added a note in the Customize the Data in the Tables for ER types section. Added FCC_M_ER_PROCESSING_COLUMNS and FCC_DS_REF_COLUMN_MAPPING tables in the Default Data in the tables section. Updated a note in the Configure the SSH Connection section. Added the following sections: • Add/ Stop MonitorIncremental Workspace Refresh • Workspace Schema • For ML and Typology Use Case • Disable the User in Compliance Studio after SSO Login • Added step 1 in the Unlock the Notebook section. • Added step 2 and modified step 3 in the Load Data into ICIJ Tables section. • Added description for column details of the FCC_ER_MAPPING in the Output Tables section.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.3.0	January 2023	Updated ASC use case information in the following sections: • Data Sourcing • Metadata Sourcing • Importing Workspace Metadata for ML4AML • Updated step 3 in the Access Compliance Studio Using SAML Realm section.
8.1.2.1.0	November 2022	• Added a note related to FIC_MIS_DATE in the Steps section. • Added a note related to unique constraint error in the STG_CUSTOMER_IDENTIFCTN_DOC table in the Data Survival section.
8.1.2.1.0	September 2022	OJET Upgrade (all UI elements are updated according to UI in the entire document. Added the following sections: • Using Wrapper Shell Script • Load Data into ICIJ Tables • Prescript Condition • Resetting Graph Pipeline Back to Day 0 • Initialize the Session • Data Model Support for AAI Applications • Typology Scenario Batch Framework • Execution Frequency Removed the "Setup PGX Server without kerberos" section. Updated steps in the OpenSearch Changes section. Updated steps in the Annual Model Validation and Define Task for Annual Model Validation sections. Updated step 1 in the Cleanup Steps When the Create Index and Load Data Job Terminated Manually and Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually sections. Updated note in the Groups in Identity Management section. Updated new tables in the Data Sourcing section. Updated commands in the Importing Workspace Metadata for ML4AML section. Removed content in the Unsupervised ML Batch Framework section. Updated the table name and added new tables for ML_Scoring in the Unsupervised Scoring section. Updated the optional parameters in the AMLES Historic Event Load section. Updated the optional parameters in the Batch and Task Parameters section. Updated the commands in the Restart Services section. Updated steps 1, 4 and 6 in the Utility Scripts section.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.0.1	July 2022	Updated minor changes in the Output Tables section.
8.1.2.0.1	June 2022	Added new **Cleanup Steps for Job Termination** sections in all ER jobs. Add new Cleanup Steps When the Create Index and Load Data Job Terminated Manually section. Add new Cleanup Steps When the Bulk Similarity Job Terminated Manually section. Add new Cleanup Steps When the Data Survival Job Terminated Manually section. Add new Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually section. Removed the NOTE related to IDNTYADMN in Groups in Identity Management section.
8.1.2.0.1	May 2022	As part of the v8.1.2.0.1 version, the following sections are added/updated: • New job is added in the Status Codes section. • New table is added in the Output Tables section. • Added Additional Configurations subsection in Create Index and Load the Data section • Added validation step in Data Survival section • Added new Job Load Data in FCC_ER_OUTPUT Table • Added Initial Run for High Volume Data section. • Added Cleanup Steps When the Create Index and Load Data Job Terminated Manually subsection in **Appendix**. • Added Utility Scripts section in **Appendix.** • Renamed the subsections and section, Resetting Entity Resolution Back to Day 0 in **Appendix**.
8.1.2.0.0	April 2022	Updated the following sections: • Removed Kubernetes-related information from Add or Modify Python Packages to the python Interpreter section. • Added a Note in Configure the SSH Connection section. • Removed the “Modify the Python Docker Images for the Python Interpreter” section. Updated the content for the deprecated ore Interpreter section.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.0.0	March 2022	Updated the following sections: • User Access and Permissioning Management • PGX Interpreter • Entity Resolution • ML Name and Address Incremental Training API • Data Memory Limit Added the following sections: • Changing Default Features and Custom Model Training • PGX Permissions • Roles, Functions and Permissions • Cleanup Steps When the Bulk Similarity Job Terminated Manually
8.1.1.1.0	November 2021	Updated the document for the v8.1.1.1.0 release.
8.1.1.0.0	October 2021	This is the first version created for the v8.1.1.0.0 release.

Table of Contents

1 Preface

This guide provides information related to the Oracle Financial Services (OFS) Compliance Studio application administrator.

1.1 Audience

This guide is intended for Administrators, and the basic knowledge of the following is recommended:

- UNIX commands
- Database concepts
- Big Data
- Python
- Scala
- Spark
- Oracle R
- SQL
- PGX
- PGQL
- Markdown

1.2 Related Documents

You can strive to keep this and all other related documents updated regularly; visit the OHC Documentation Library to download the latest version available there. The list of related documents is provided here.

- Oracle Financial Services Compliance Studio Installation Guide
- Oracle Financial Services Compliance Studio User Guide
- Oracle Financial Services Compliance Studio Matching Guide
- Oracle Financial Services Compliance Studio Data Model Guide
- Oracle Financial Services Compliance Studio Release Notes and Readme

1.3 Conventions

Table 2 explains the text conventions used in this guide.

Table 2: Convention

Convention	Description
Italics	• Names of books, chapters, and sections as references • Emphasis

Table 2: Convention

Bold	• The object of an action (menu names, field names, options, button names) in step-by-step procedures • Commands typed at a prompt • User input
`Monospace`	• Directories and subdirectories • File names and extensions • Process names • Code sample, including keywords and variables within a text and as separate paragraphs, and user-defined program elements within a text
Hyperlink	Hyperlink type indicates the links to external websites and internal document links to sections.
<Variable>	Substitute input value

1.4 Abbreviations

Table 3 lists the abbreviations used in this document.

Table 3: Abbreviations

Abbreviation	Meaning
OFS	Oracle Financial Services
OFSAA	Oracle Financial Services Analytical Application
BD	Behavior Detection
FCDM	Financial Crime Data Model
ICIJ	International Consortium of Investigative Journalists
MMG	Model Management and Governance
SSO	Single Sign-On
SSH	Secure Shell

2 About Compliance Studio Administration

OFS Compliance Studio is an advanced analytics application that supercharges anti-financial crime programs for better customer due diligence, transaction monitoring, and investigations by leveraging the latest innovations in artificial intelligence, open-source technologies, and data management. It combines Oracle's Parallel Graph Analytics (PGX), Machine Learning for AML, Entity Resolution, and notebook-based code development and enables Contextual Investigations in one platform with complete and robust model management and governance functionality.

Topics:

2.1 Capabilities offered by Compliance Studio

- Purpose Built for Fighting Crime
  - Fully defined and sourced Financial Crime Graph Model supporting detection and investigation
  - Provided Accelerators for finding the needles in the haystack.
  - What if Analysis for existing Scenarios
  - Integration with ECM and Investigation Hub to provide meaningful guidance to investigators for rules-based and ML-generated alerts
  - Enterprise-ready and compatible with the underlying OFSAA framework
  - Works with earlier 8.0.x releases of Oracle Financial Crime and Compliance Management Anti Money Laundering (AML), Enterprise Case Management, and Fraud applications.
- Entity Resolution for AML
  - Entity Resolution to enhance monitoring effectiveness and provide a single customer view
  - Linking and Resolution across internal & external data to improve single entity detection
  - Allows for Scenario/Model detection across internal data
  - Multi-attribute enabled with ML boosts for Name/Address models
  - Prebuilt Integrations and easily configurable for Data Sources like ICIJ, Safari, etc.
- Analytics of Choice
  - Choose from our proprietary models or bring your own
  - Fully embedded Graph Analytics Engine and Financial Crime Model
  - Embedded with a highly scalable in-memory Graph Analytics Engine (PGX)
  - Industry's most intuitive Graph Query Language to gain rapid insights
- Model Management & Governance
  - End-to-end management from model creation to model deployment.
    - Data Ingestion (Oracle DB, Graph, Hive)
    - Model Development

    - Supports virtually all open source packages, interpreters, etc.
    - Process in Database or Big Data
    - Model Training
    - Model Performance Evaluation
    - Model Explainability
    - Model Tracking and Audit
    - Approval Mechanisms
    - Model Deployment
    - Scheduling
    - Ongoing Monitoring
- ML Foundation for Financial Crimes
    - Integrated with Oracle Financial Crime Application Data and readily usable across the enterprise financial crime data lake.
    - Pre-engineered features and transformations to address each use case
    - Simplified APIs for each stage of the modeling lifecycle
    - Leverage the power of Graph, Supervised ML, and Unsupervised ML to build typology detection models, detect anomalies, and risk score customers or events
    - Event Scoring for false positive prediction and disposition
    - Ongoing Monitoring of Model Performance and Concept Drift
    - Automated Scenario Calibration and Scenario Conversion Utility for Oracle AML Scenarios

2.2 Configurable Features

The following are the key configurable features in Compliance Studio:

- Create users and roles to access Compliance Studio to access through AAI/SSO
- Assign roles and groups with required permissions
- The ability to customize and create interpreter variants to provide or restrict access to users
- Modify ready-to-use Python packages and versions
- Customize rulesets to generate similarity edges and resolved entities
- Apply Graph Fine-Grained Access Control to redact the sensitive data in the Graphs
- Monitor tasks that the logged-in users perform
- Offers ready-to-use extract, transform, load (ETL) operations for the creation of a global graph using Graph Pipelines.
- Entity resolution based on configurable rules.

2.3 Administration Overview

This section provides an overview of administration activities performed by an Administrator after installing the Compliance Studio application.

The following are the key configuration activities performed by an Administrator in Compliance Studio:

- Mapping User Groups*:* To access the application, users must be authenticated. In Compliance Studio, users and roles are authenticated based on Realms, such as FCCRealm, SAMLRealm, etc. These Realms use Identity Management systems to authenticate users. FCCRealm - uses Oracle Financial Services Analytical Applications Infrastructure (OFSAAI), and SAMLRealm uses an identity provider (IDP).
- User Group - Role Mapping: After authentication of users and roles, they must be authorized to use the application. The Compliance Studio offers a rich permission system, and users are mapped to the permissions to use the application.
- Configure Interpreters: Interpreters are used to execute code in different languages. Plug-ins enable users to use a specific language to process data on the selected execution platform. The Compliance Studio provides ready-to-use interpreters, such as jdbc-interpreter, python interpreter, etc. In Compliance Studio, you can either use a default interpreter variant or create a new variant for an interpreter to provide access to the database for different users. Interpreters are linked using credentials (a wallet and a password) to enable secure data access. Interpreters are configured based on usage.
- Entity Resolution: OFS Compliance Studio provides Entity Resolution (ER) capability. It allows firms to break through barriers in their data by gaining single views of their customers and their external entities and have the choice of monitoring them both under one consolidated Global Party.

  OFS Compliance Studio Entity Resolution is a configurable process that allows data to be matched and merged to create contextual links in the global graph or resolve relational party records to a global party record as part of ingestion. OFS Compliance Studio has pre-built configurations supporting matching (or linking) in the FCGM and resolving entities in CSA for data being loaded into Financial Services Data Foundation (FSDF).
- Schedule Scenario Notebook Execution: You can schedule a notebook execution using the scheduler.

> **NOTE** In the current release, Notebook execution using Batch is deprecated and will be removed in the future release. It is recommended to use the scheduler to execute the notebook in Batch.
>
> You can see the Migrating the Data from ElasticSearch to OpenSearch.

2.3.1 Key Concepts

This section provides insight into the following key concepts:

- **Interpreter**: An interpreter is a program that directly executes instructions written in a programming or scripting language without requiring them previously to be compiled into a machine language program. They are plug-ins that enable users to use a specific language to process data in the backend. Examples of Interpreters are jdbc-interpreter, spark-interpreters, python-interpreters, etc. Interpreters allow you to define customized drivers, URLs, passwords, connections, SQL results to display, etc.
- **Zeppelin Interpreter**: A plug-in enables Zeppelin users to use a specific language or data-processing-backend. For example, to use the Scala code in Zeppelin, you need a %spark interpreter.
- **Zeppelin**: Interactive browser-based notebooks enable data engineers, data analysts, and data scientists to be more productive by developing, organizing, executing, and sharing data code and visualizing results without referring to the command line or requiring the cluster details.

Notebooks allow these users not only allow to execute but to interactively work with long workflows.

- **Markdown (md)**: A plain text formatting syntax designed so that it can be converted to HTML. Use this section to configure the markdown parse type.
- **Parallel Graph Analytics (PGX)**: Graph analysis lets you reveal latent information that is not directly apparent from fields in your data but is encoded as direct and indirect relationships - metadata - between elements of your data. This connectivity-related information is not obvious to the naked eye but can have tremendous value when uncovered. PGX is a toolkit for graph analysis, supporting both efficient graph algorithms and fast SQL-like graph pattern matching queries.
- **PySpark**: A Python API is written in Python to support Spark. Spark is a distributed framework that can handle Big Data analysis. Spark is a computational engine that works with huge sets of data by processing them in parallel and batch systems.
- **Spark**: A fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python, and R. Spark is an optimized engine that supports general execution graphs.
- **PGQL**: A graph query language built on top of SQL, bringing graph pattern matching capabilities to existing SQL users and new users interested in graph technology but who do not have an SQL background.
- **Data discovery**, **exploration**, **reporting**, and **visualization** are key components of the data science workflow. Zeppelin provides a "Modern Data Science Studio" that supports ready-to-use Spark and Hive. Zeppelin supports multiple language backends, which has support for a growing ecosystem of data sources. Zeppelin's notebooks provide interactive snippet-at-time experience to data scientists. You can see a collection of Zeppelin notebooks in the Hortonworks Gallery.
- **Keytab File**: A Keytab is a file containing pairs of Kerberos principles and encrypted keys (which are derived from the Kerberos password). You can use a keytab file to authenticate to various remote systems using Kerberos without entering a password. However, when changing your Kerberos password, you must recreate all your keytabs files. They are commonly used to allow scripts to automatically authenticate using Kerberos, without requiring human interaction or access to the password stored in a plain-text file. The script can use the acquired credentials to access files stored on a remote system.
- **Oracle Wallet**: Oracle Wallet is a file that sources database authentication and signing credentials. It allows users to securely access databases without providing credentials to third-party software, and easily connect to Oracle products.
- **OpenSearch:** OpenSearch is a distributed search and analytics engine for all data types, including textual, numerical, geospatial, structured, and unstructured.

3 User Access and Permissioning Management

Compliance Studio uses a realm based on unique authentication and authorization for its users. Realm is a security policy domain defined for the application server. It is used to authenticate and authorize users of Compliance Studio.

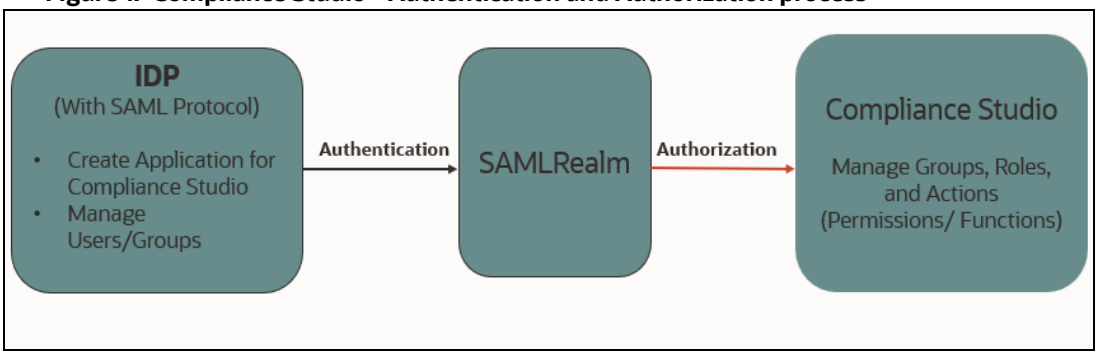
SAML Realm is selected based on the Identity Provider (IDP) during the installation. For more information, see the OFS Compliance Studio Installation Guide.

The Compliance Studio application is accessed using the following realm that you have selected during the installation of the Compliance Studio application:

- **SAMLRealm**: The SAMLRealm uses an identity provider (IDP) Identity Management system to support the SAML2.0 protocol 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 Groups are created in the IDP. For more information, see Access Compliance Studio Using SAML Realm section.

The following image illustrates the authentication and authorization process in the Compliance Studio.

Figure 1: Compliance Studio - Authentication and Authorization process



Topics:

- Mapping User Groups
- Access Compliance Studio Using SAML Realm

3.1 Mapping User Groups

Users must be mapped to User Groups that are mapped to access Oracle Financial Services Compliance Studio (OFS CS). The following subsections provide information about the user groups and roles required in addition to the information about configuring the user groups.

3.1.1 User Groups

Table 4 gives details about the User Groups in the OFS CS application.

Table 4: User Groups

User Group	Description
IDNTYADMN	Identity Administrator group

Table 4: User Groups

User Group	Description
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.
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.
ERADMIN	Entity resolution admin group. **NOTE:** This group has to be created manually in AAI and map it to the users.
ERUSER	Entity resolution user group. **NOTE:** This group has to be created manually in AAI and map it to the users.

NOTE

- At the first-time login, User Group mappings are initialized from AAI/IDCS for the newly provisioned users. These will be reflected in OFS CS Admin Console in next OFSC CS login.
- If User Group mappings are deleted in AAI/IDCS, it would not delete in OFS CS Admin Console. Admin needs to delete this in OFS CS 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.

3.1.2 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 5 lists the roles which are assigned to a particular User Group.

Table 5: User Group to Role Mapping

Group Name	Role Name
DSREDACTGRP	DSREDACT
IDNTYADMN	Batch Advance Role
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

Table 5: User Group to Role Mapping

Group Name	Role Name
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 5: 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

3.1.3 Functions and Roles required to perform CRUD operations for Conda

Table 6 provides details about the Functions and Roles required to perform CRUD operations for Conda in the OFS CS application.

For more information, see the **Conda Environments** section in the OFS Compliance Studio User Guide.

Table 6: 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

Table 6: Functions and Roles

Function	Role	Groups Mapped	Access
CONDAENVADD	CONDAENVWRITE	• MDLREV • MDLAPPR	Add a conda environment

3.2 Access Compliance Studio Using SAML Realm

This section provides information on managing users who can access Compliance Studio with Identity Provider (IdP or IDP). The IdP acts as the Single Sign-On (SSO) service provider for implementations between Compliance Studio, Investigation Hub, 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 Preconfigured Groups to access Compliance Studio using SAMLRealm.

To integrate Compliance 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. See the User Groups section for more information.
2. Create a SAML application in IDP.
3. Configure the SAML application.

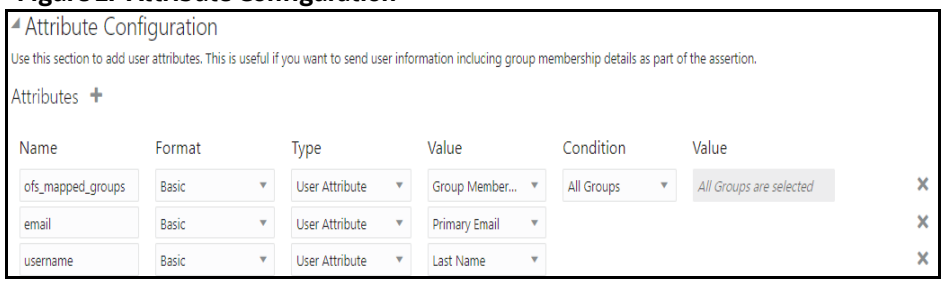
   Key configurations in the SAML application is as follows:

   - Entity ID: `https://<FQDN of Compliance studio Linux Server>:7001/cs`
   - Assertion Consumer URL: `http:// <FQDN of Compliance studio Linux Server>:7001/cs/home`

   NOTE **Response** in SAML response must be signed.

   - Include Signing Certificate in Signature: **Enabled**
   - Include Signing Certificate in Signature: **SHA-256**
   - Enable Single Logout: **Enabled**
   - Logout Binding: **POST**
   - Single Logout URL (SAML_LOGOUT_URL): `http://<FQDN of compliance studio>:7001/cs/signoff`
   - Logout Response URL: `http://<FQDN of compliance studio>:7001/cs/signoff`
   - Encrypt Assertion: **Disabled**
   - SAML Attribute Configuration

Figure 2: Attribute Configuration



Update the SAML attribute configuration as tabulated in the Table 7.

Table 7: 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.

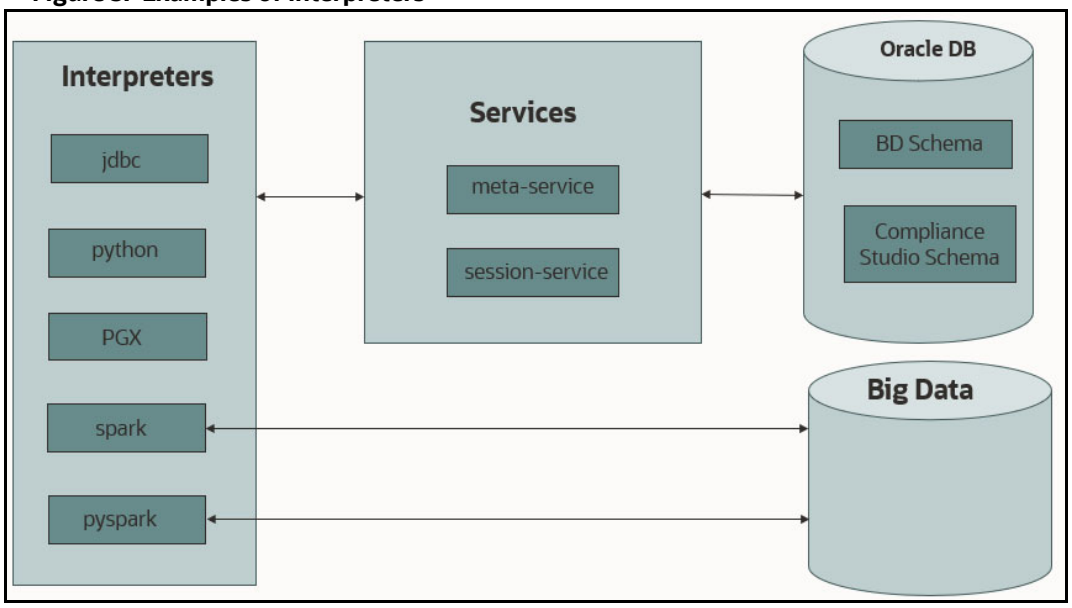
4 Interpreter Configuration and Connectivity

An interpreter is a program that directly executes instructions written in a programming or scripting language without requiring them previously to be compiled into a machine language program. Interpreters are plug-ins that enable users to use a specific language to process data in the backend. Examples of Interpreters are jdbc-interpreter, spark-interpreters, python-interpreters, etc. Interpreters allow you to define customized drivers, URLs, passwords, connections, SQL results to display, etc.

In Compliance Studio, Interpreters are used in Notebooks to execute code in different languages. Each Interpreter has a set of adjusted and applied properties across all notebooks. For example, using the python-interpreter makes it possible to change between versions, whereas the jdbc-interpreter offers to customize the URL, schema, or credentials. In Compliance Studio, you can either use a default interpreter variant or create a new variant for an interpreter. You can create more than one variant for an interpreter. The benefit of creating multiple variants for an Interpreter is to connect different versions of interpreters (Python version: 3, Python version: 2, etc.). This helps to connect a different set of users and database schema. For example, Compliance Studio schema, BD schema, etc. Compliance Studio provides secure and safe credential management such as Oracle Wallet (jdbc wallet), Password (jdbc password), or KeyStores to link to interpreter variants to access secured data.

The following image illustrates the examples of interpreters used in Compliance Studio and database connections.

Figure 3: Examples of Interpreters



Topics:

- Configure Interpreters
- Create a Credential
- Link Credentials
- Create an Interpreter Group
- Create an Interpreter Variant
- Enable Additional Spark or PySpark interpreter

4.1 Configure Interpreters

Compliance Studio has ready-to-use interpreters such as python, jdbc Interpreter, etc. You can configure them based on the use case. Additional variants of interpreters are created as multiple users might require different settings to access the database securely. The jdbc Interpreters use the credentials to enable secure data access.

> **NOTE** python, pyspark, spark, and python are some of the other available interpreters.

Interpreters are configured when you want to modify URL, data location, drivers, enable or disable connections, etc.

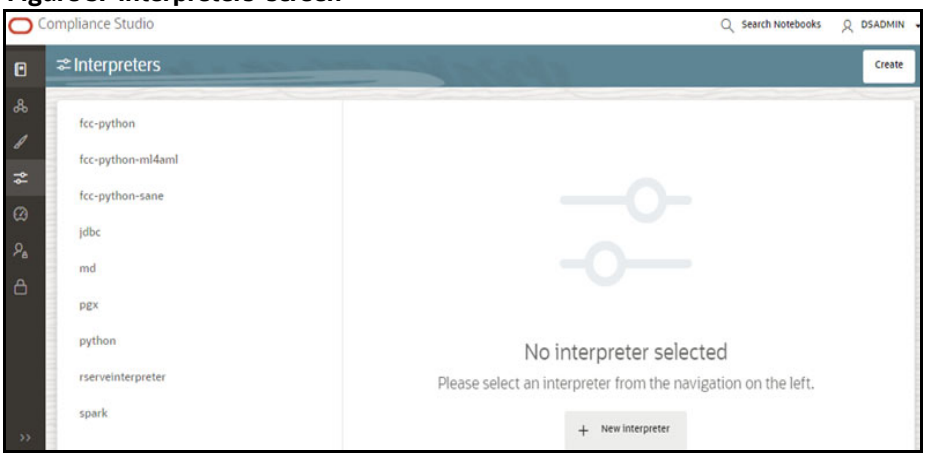
To configure ready-to-use interpreters, follow these steps:

1. On the **Workspace Summary** page, select Launch workspace to display the **CS Production workspace** window**.**

Figure 4: Workspace Summary

2. Click on the **User Profile** drop-down list and select **Data Studio Options** widget. The following options are available:
   - Interpreters
   - Tasks
   - Permissions
   - Credentials
   - Templates
3. Click **Interpreters** that you want to view from the list displayed on the LHS. The default configured interpreter variant is displayed on the RHS.

Figure 5: Interpreters' screen

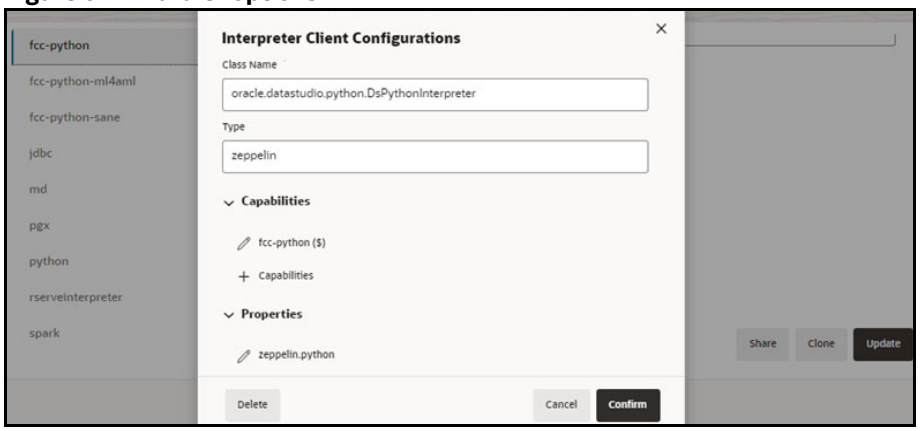


4. Modify the values in the fields as per requirement. For example, to modify a parameter's limit, connect to a different schema, PGX server, etc.

   You can modify the values in the following UI options:

   - Wizard

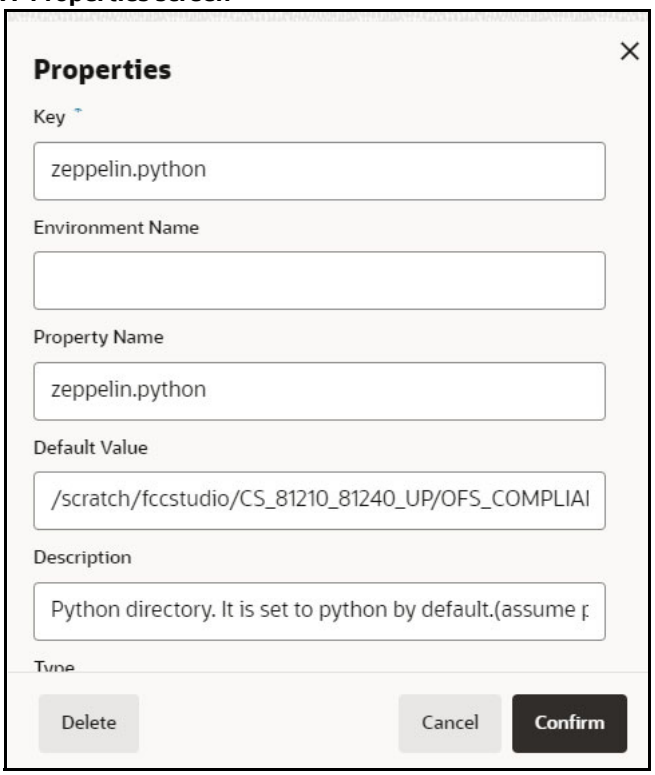
Figure 6: Wizard UI options



An interpreter can group multiple interpreter clients that all run in one JVM process and can be stopped together.

For example, the spark interpreter group contains the spark and pyspark interpreter client.

Figure 7: Properties screen



Group Configuration

Initial Code

For example, when using a Spark interpreter group with spark and pyspark interpreter clients. If you define the initialization code for the spark interpreter group, the initialization code will run when the runtime environment is created, i.e., the first time a user runs a paragraph of either spark or pyspark in a notebook with Compliance Studio running in NOTEBOOK session mode.

Initial Code Capability

The initial code capability defines what interpreter client to use to run the group initial code. For example, in the spark interpreter group, you would select the spark capability as the initial code capability to create a spark context for the group JVM process.

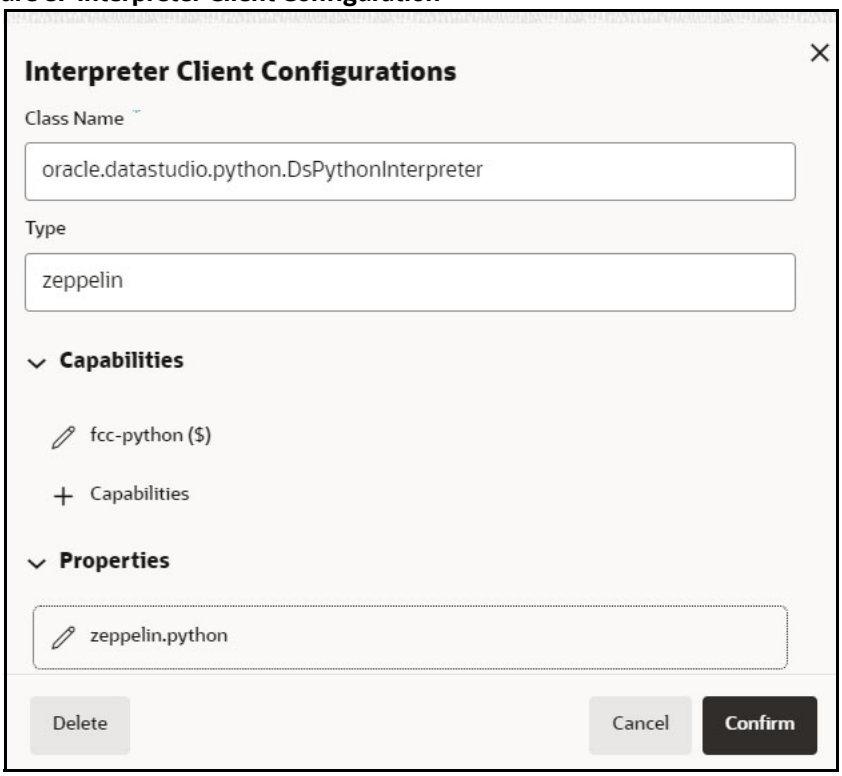
Credential Configurations

For linking any credentials to the interpreter, you have to define what credential types should be used and what credential mode to use. For example, the jdbc interpreter supports a credential type of type Password for the credential qualifier **jdbc_password** and a credential type of type Oracle Wallet for the credential qualifier **jdbc_wallet**. After defining the credential configuration, a new section for selecting the respective credential values will appear.

Interpreter Client Configuration

Interpreter properties can be configured for each interpreter client.

Figure 8: Interpreter Client Configuration



Lifecycle Configuration

Host Mode

In the Host lifecycle mode, the following properties can be configured:

- **Host**: The hostname on which the interpreter is listening. For example, localhost if the **interpreter** runs on the same machine as the server.
- Port: The port on which the interpreter is listening.

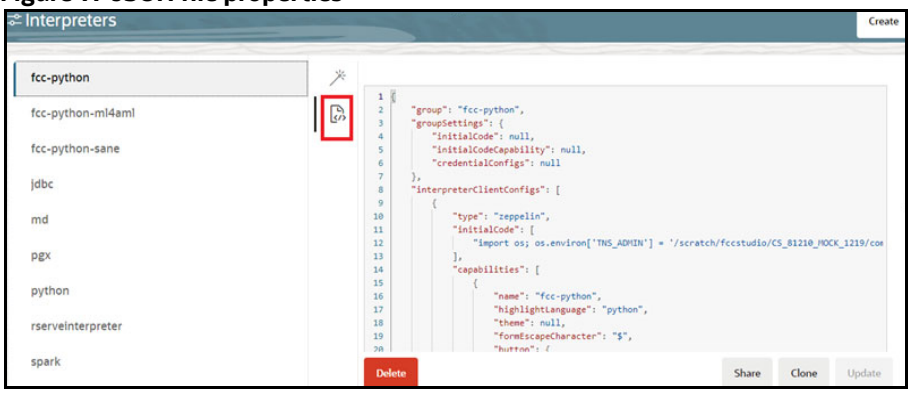
Credentials

A credential section appears if you have defined a credential configuration as part of the group settings. For each credential qualifier, an already defined credential can be selected. If the credential mode Per User is used, each individual user has to select their own credential.

- **JSON**:

You can modify the values in the properties of the interpreter in the JSON file, as shown in the following figure.

Figure 9: JSON file properties



5. Click **Update**. The modified values are updated in the Interpreter.
6. The user can also perform **Share**, **Clone**, and **Delete** operations on this screen.

Table 8 lists the Ready-to-use interpreter in Compliance Studio:

Table 8: Ready-to-use interpreter

Interpreters	Description
python Interpreter	The python interpreter is used to write Python code in a notebook to analyze data from different sources, machine learning, artificial intelligence, etc. The python interpreter uses a python conda environment. Compliance Studio comes with predefined conda environments as follows: • default_8.1.2.6.0 • ml4aml_8.1.2.6.0 • sane_8.1.2.6.0 Before executing any python notebooks, you need to attach the conda environment using drop-down option.
jdbc Interpreter	The jdbc interpreter is a ready-to-use interpreter used to connect to Studio schema. This Interpreter is used to connect and write SQL queries on any schema without any restriction. In the jdbc Interpreter, you can configure schema details, link Wallet Credentials to the jdbc Interpreter, etc. **NOTE:** This feature is not recommended approach because it can only be used to connect to a single schema, and all users will have access to that, rather than access being managed per user. In future releases this interpreter will not be enabled by default but instructions will be given to enable if required. **Limitation** • Data source configuration is not dynamic; instead, it is static from the Interpreter Configuration screen. • There is no restriction or secure access of data provided with this interpreter. **Recommendation** Users are recommended to use a python interpreter to get dynamic data source configuration; even data access permission features can also be used with this interpreter.

Table 8: Ready-to-use interpreter

md Interpreter	The md interpreter is used to configure the markdown parser type. This Interpreter displays text based on Markdown, which is a lightweight markup language. The connection does not apply to this Interpreter.
pgql Interpreter *(part of PGX interpreter)*	The pgql interpreter is a ready-to-use interpreter used to connect the configured PGX server. This Interpreter is used to perform queries on the graph in Compliance Studio. PGQL is a graph query language built on top of SQL, bringing graph pattern matching capabilities to existing SQL users and new users interested in graph technology but who do not have an SQL background.
pgx-python *(part of PGX interpreter)*	The pgx-python interpreter is a ready-to-use interpreter used to connect to the configured PGX server. It is a **python** based interpreter with a PGX python client embedded in it to query on graph present in the PGX server. By default, this Interpreter points to ml4aml Python Virtual environment.
pgx-algorithm Interpreter *(part of PGX interpreter)*	The pgx-algorithm interpreter is a ready-to-use interpreter that connects to the configured PGX server. This Interpreter is used to write an algorithm on the graph and is also used in the PGX interpreter.
pgx-java Interpreter *(part of PGX interpreter)*	The pgx-java interpreter is a ready-to-use interpreter that connects to the configured PGX server. It is **Java11** based interpreter with a PGX client embedded in it to query on graph present in the PGX server.
pyspark Interpreter	The pyspark interpreter connects to the big data environment by default. Users must write code for connection either in the Initialization section or in the notebook's paragraph. This Interpreter is used to write the pyspark language to query and perform analytics on data present in big data. This requires additional configuration, which must be performed as a prerequisite or as post-installation with the manual change of interpreter settings. In the pyspark Interpreter, you can configure the Python binary executable to use for PySpark in both driver and workers, set true to use IPython, else set to false, etc.
spark Interpreter	The spark interpreter connects to the big data environment by default. Users must write for connection either in the Initialization section or in the notebook's paragraph. This Interpreter is used to perform analytics on data present in the big data clusters in the Scala language. This requires additional configuration, which must be performed as a prerequisite or as post-installation with the manual change of interpreter settings. In the spark interpreter, you can configure the cluster manager to connect, print the Read Eval Print Loop (REPL) output, the total number of cores to use, etc.

4.1.1 python Interpreter

NOTE

- This preseeded conda environment replaces the three versions of the python interpreters (fcc-python, fcc-python-ml4aml, and fcc-python-sane) in the previous versions, and they will still exist in upgraded environments.
- If the User wants to configure the python interpreter (fcc-python, fcc-python-ml4aml, and fcc-python-sane) in the v8.1.2.6.0 environment, they can configure it manually. For more information, see **Manual Steps for Configuring Python Interpreter** section in the OFS Compliance Studio Installation Guide.

In Compliance Studio, the python interpreter uses a python conda environment. Compliance Studio comes with predefined conda environment as follows:

- **default_8.1.2.6.0**
- **ml4aml_8.1.2.6.0**
- **sane_8.1.2.6.0**

%python interpreter points to a different conda environment. Table 9 lists the predefined conda environment.

Table 9: Predefined Conda Environment

Conda Environment	Description
default_8.1.2.6.0	Default python interpreter.
ml4aml_8.1.2.6.0	Python interpreter for ML4AML use cases.
sane_8.1.2.6.0	Python interpreter for scoring Name and Address Matching.

NOTE

- You can create a new conda environment in the Compliance Studio UI. For more information, see the **Conda Environments** section in the OFS Compliance Studio User Guide.
- Configure the python libraries. For more information about python libraries, see the section.

4.1.1.1 Configure a python Interpreter

To configure an python interpreter variant, follow these steps:

1. On the Interpreter page LHS menu, select python. The python interpreter pane is displayed.
2. On Interpreter Settings page, expand **Interpreter Client Configurations** and click Edit icon for **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed.
3. Enter the following information in the python interpreter variant pane as described in the Table 10.

Table 10: Python Interpreter Settings

Field	Description

Table 10: Python Interpreter Settings

zeppelin.python	Enter the Python installed path. The value points to the default Python version set for the Interpreter. **NOTE**: To use a different Python version, see Change Python Version in the python Interpreter section.
zeppelin.python.useIPython	Set to **True** to use IPython, else set to **False**.
zeppelin.python.maxResult	Enter the maximum number of results that must be displayed.
zeppelin.interpreter.output.limit	Output message from interpreter exceeding the limit will be truncated.

4.1.1.2 Change Python Version in the python Interpreter

In the python Interpreter, the Linux console uses the default python version in. `/user/fccstudio/python_user/bin/python` as value. If you want to modify the python version, either you can create an interpreter variant or modify the existing python version in the same interpreter variant.

> **NOTE** The **python2** is the default version used in the Linux console and is no longer supported. Hence, you can use any version of **python3** or any conda environment with a specific python version or a specific version of python packages.

To use a different version of Python, follow these steps:

1. Navigate to the **python** Interpreter Settings page.
2. Expand **Interpreter Client Configurations** and click Edit icon for <Class Name> (zeppelin). The Interpreter Client Configurations Window is displayed.
3. Click `zeppelin.properties`. The Properties window is displayed.
4. Change the default Python version in the `Default Value` parameter to the new version. `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/python-packages/defaultVirtualEnv/bin/<Python Version>`.

   By default, it is python3.

   For example, `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/python-packages/defaultVirtualEnv/bin/python3`.
5. Create a new interpreter variant and configure the version in the `Default Value` parameter. For information on creating a new interpreter variant, see Create an Interpreter Variant section. For example, to use Python 3.6.13, create a new python interpreter variant and enter the value as python 3.6.13.

4.1.1.3 Add or Modify Python Packages to the python Interpreter

When a user wants to write something in Python, but the packages are not present. Use case: ML or AI code. By default, the Linux server has a limited number of packages present inside it.

To add desired Python packages to the python Interpreter, follow these steps:

- For Compliance Studio installed on-premise:

  To add or modify Python libraries to the python Interpreter, contact System Administrator to install the required additional Python libraries on the Processing Server (Studio Notebook Server). The newly added Python libraries must be accessible to the Linux user for Compliance Studio.

To add the python packages for python3, follow these steps:

1. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/python-packages/bin` directory.
2. Run the following command:

```
python3 -m pip install <package name> --user
```

4.1.2 jdbc Interpreter

> **NOTE** This feature is not recommended approach because it can only be used to connect to a single schema, and all users will have access to that, rather than access being managed per user. In future releases this interpreter will not be enabled by default but instructions will be given to enable if required.
>
> **Limitation**
>
> - Data source configuration is not dynamic; instead, it is static from the Interpreter Configuration screen.
> - There is no restriction or secure access of data provided with this interpreter.
>
> **Recommendation**
>
> Users are recommended to use a python interpreter to get dynamic data source configuration; even data access permission features can also be used with this interpreter.

The jdbc Interpreter is a ready-to-use interpreter used to connect Studio schema without OFSAA. This Interpreter is used to connect and write SQL queries on any schema without any restriction. The jdbc interpreter has no security attributes. It can be used to access any schema. In the jdbc interpreter, you can configure schema details, link Wallet Credentials to the jdbc Interpreter, etc.

Prerequisites

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/conf` directory.
2. Open the `application.yml` file and update **overwrite-builtin** property as **false**.
3. Save the changes and close the `application.yml` file.
4. Restart Compliance Studio.

4.1.2.1 Configure a jdbc Interpreter Variant

NOTE

This feature is not recommended approach because it can only be used to connect to a single schema, and all users will have access to that, rather than access being managed per user. In future releases this interpreter will not be enabled by default but instructions will be given to enable if required.

Limitation

- Data source configuration is not dynamic; instead, it is static from the Interpreter Configuration screen.
- There is no restriction or secure access of data provided with this interpreter.

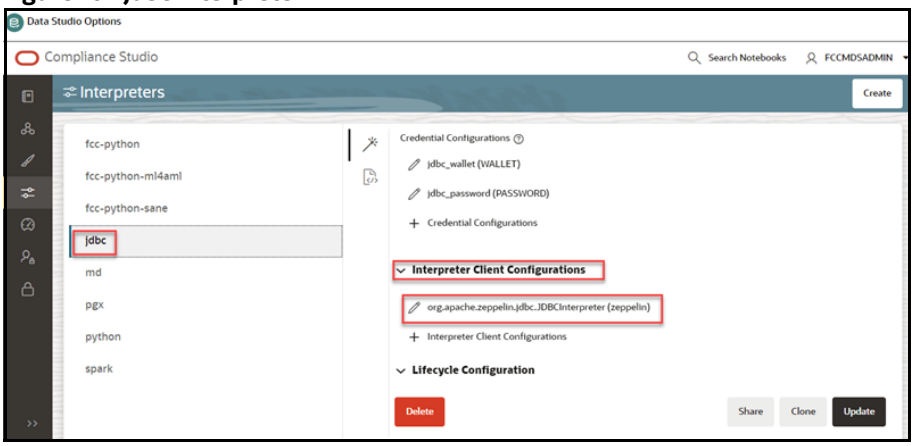
Recommendation

Users are recommended to use a python interpreter to get dynamic data source configuration; even data access permission features can also be used with this interpreter.

To configure a jdbc interpreter variant, follow these steps:

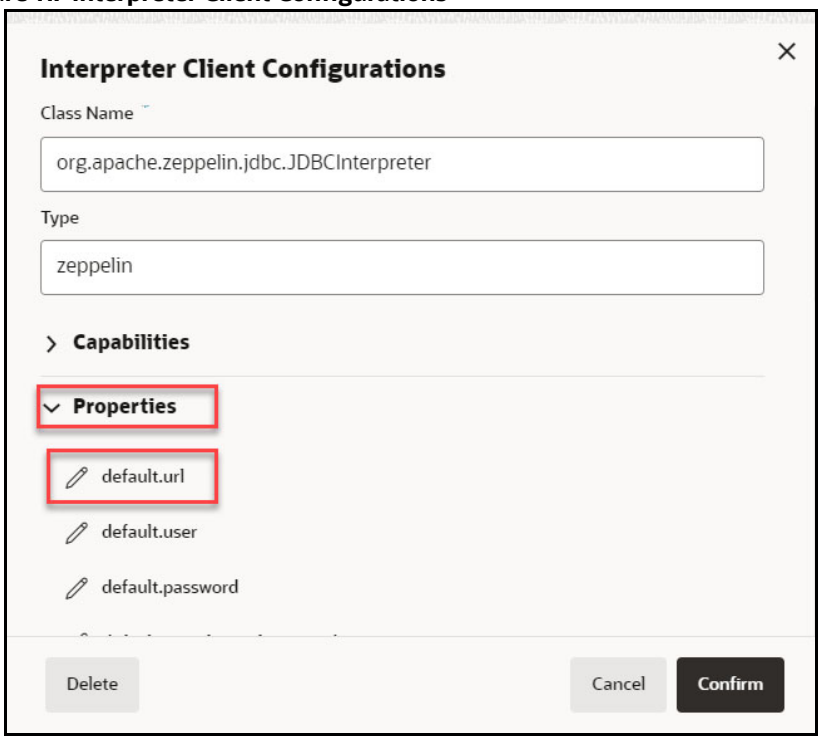
1. On the Interpreter page LHS menu, select **jdbc**. The jdbc interpreter pane is displayed.

Figure 10: jdbc Interpreter



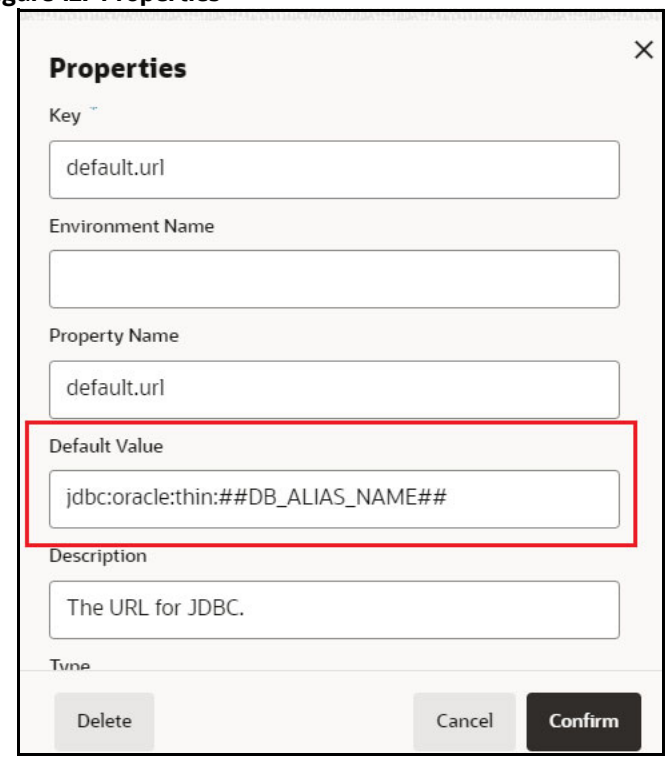
2. On Interpreter Settings page, expand **Interpreter Client Configurations** and click Edit icon on the **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed.

Figure 11: Interpreter Client Configurations



3. Click **default.url** under the Properties. The Properties page is displayed.

Figure 12: Properties



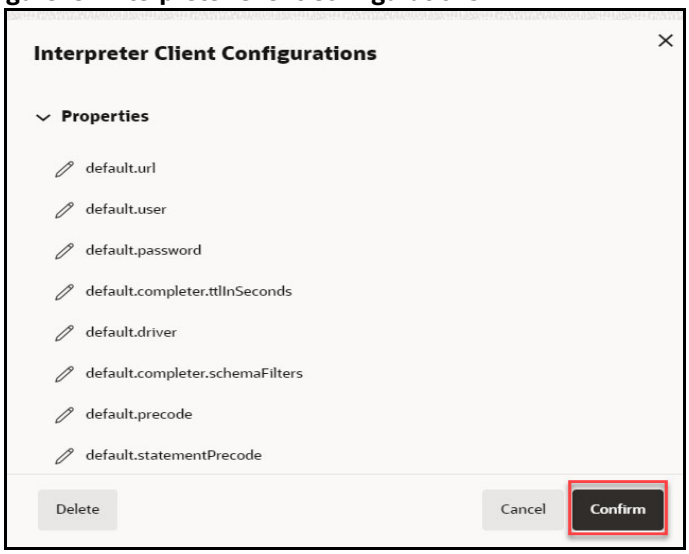
4. Enter the alias name in the **Default Value** field.

   The alias name is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/wallet/tnsnames.ora` directory.

For example, jdbc:oracle:thin:##DB_ALIAS_NAME##

5. Click **Confirm**. The Interpreter Client Configurations page is displayed.

Figure 13: Interpreter Client Configurations



6. Click **default.user** property and it should be null in the **Default Value** field.
7. Click **default.password** property and it should be null in the **Default Value** field.

> **NOTE** Retain the default settings for the remaining properties in the Interpreter Client Configurations.

8. Click **Update**. The modified values are updated in the Interpreter.

4.1.2.2 Link Wallet Credentials to jdbc Interpreter

> **NOTE** This feature is not recommended approach because it can only be used to connect to a single schema, and all users will have access to that, rather than access being managed per user. In future releases this interpreter will not be enabled by default but instructions will be given to enable if required.
>
> **Limitation**
>
> - Data source configuration is not dynamic; instead, it is static from the Interpreter Configuration screen.
> - There is no restriction or secure access of data provided with this interpreter.
>
> **Recommendation**
>
> Users are recommended to use a python interpreter to get dynamic data source configuration; even data access permission features can also be used with this interpreter.

Compliance Studio provides secure and safe credential management. Examples of credentials are passwords, Oracle Wallets, or KeyStores. Use this section to link credentials (a wallet and a password) to the jdbc interpreter variant to enable secure data access. This linking enables the jdbc interpreter to

securely connect to the specified Oracle database. For more information on linking Wallet Credentials to jdbc Interpreter, see the Link Credentials section.

> **NOTE** The Credentials section is enabled if an interpreter variant can accept credentials.

You can also create new credentials and link to jdbc Interpreter. For more information, see Create a Credential section.

4.1.3 md Interpreter

This Interpreter displays text based on Markdown, which is a lightweight markup language. In the md interpreter, you can configure the markdown parser type. Markdown (md) is a plain text formatting syntax designed so that it can be converted to HTML. Use this section to configure the markdown parser type.

To configure the md interpreter variant, follow these steps:

1. On the md Interpreter page LHS menu, select md. The md interpreter pane is displayed.
2. On Interpreter Settings page, expand **Interpreter Client Configurations** and click Edit icon for **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed.
3. Enter the markdown parser type and click **Update**. To confirm the modified configuration.

4.1.4 PGX Interpreter

The PGX has the following interpreters:

- **pgql**: The pgql interpreter is a ready-to-use interpreter used to connect the configured PGX server. This Interpreter is used to perform queries on the graph in Compliance Studio. PGQL is a graph query language built on top of SQL, bringing graph pattern matching capabilities to existing SQL users and new users interested in graph technology but who do not have an SQL background.
- **pgx-algorithm:** The pgx-algorithm is a ready-to-use interpreter used to connect to the configured PGX server. This Interpreter is used to write an algorithm on the graph and is also used in the PGX interpreter.
- **pgx-java**: The pgx-java interpreter is a ready-to-use interpreter used to connect to the configured PGX server. It is **Java11** based interpreter with a PGX client embedded in it to query on graph present in the PGX server.
- **pgx-python**: The pgx-python interpreter is a ready-to-use interpreter used to connect to the configured PGX server. It is a **python** based interpreter with a PGX python client embedded in it to query on graph present in the PGX server. By default, this Interpreter points to ml4aml Python Virtual environment.

To configure the pgql interpreter variant, follow these steps:

1. On the Interpreter page LHS menu, select pgql. The pgql interpreter pane is displayed.
2. On Interpreter Settings page, expand **Interpreter Client Configurations** and click Edit icon for **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed.

3. Enter the following information in the pgql interpreter variant pane as tabulated in the Table 11.

Table 11: PGX interpreter

Field	Description
graphviz.formatter.class	Enter the class which implements the formatting of the visualization output. For example, `oracle.datastudio.graphviz.formatter.DataStudioFormatter`
graphviz.driver.class	Enter the class which implements the PGQL driver. For example: `oracle.pgx.graphviz.driver.PgxDriver`
base_url	Enter the base URL of the PGX. For example, `http://<HOSTNAME>:7007`
zeppelin.interpreter.output.limit	Enter the output message limit. Any message that exceeds the limit is truncated. For example, 102 or 400.
num_cached_resultsets	Maximum number of results sets kept open on the PGX server per interpreter session. Only checked when the interpreter is used, and therefore it should only be used with expiring interpreter sessions. For example: 5
resultset_expiration_time_secs	Number of seconds after which unused results sets are closed on the PGX server. Only checked when interpreter session is used and should only be used with expiring interpreter sessions. For example: 3600
zeppelin.python.useIPython	Set to 'True' to use IPython, else set to 'False'.
zeppelin.python	Enter the Python installed path. The value points to the default Python version set for the Interpreter. **NOTE**: To use a different Python version, see Change Python Version in the python Interpreter section.

4.1.5 pyspark Interpreter

Users must write for connection either in the Initialization section or in the notebook's paragraph. This interpreter is used to write the pyspark language to query and perform analytics on data present in big data. This requires additional configuration, which must be performed as a prerequisite or as post-installation with the manual change of interpreter settings.

In the pyspark interpreter, you can configure the Python binary executable for PySpark in both driver and workers, set 'True' to use IPython, else set it to 'False'.

To configure the pyspark interpreter variant, follow these steps:

1. On the Interpreter page LHS menu, select pyspark. The pyspark interpreter pane is displayed.

2. On Interpreter Settings page, expand **Interpreter Client Configurations** and click Edit icon for **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed.

3. Enter the following information in the pyspark interpreter variant pane as tabulated in the Table 12.

Table 12: pyspark interpreter

Field	*Description*
zeppelin.pyspark.python	Enter the Python binary executable for PySpark in both drivers and workers. The default value is python. For example, `python`
zeppelin.pyspark.useIPython	Set to 'True' to use IPython, else set to 'False'.
zeppelin.interpreter.output.limit	Output message from interpreter exceeding the limit will be truncated

4.1.6 spark Interpreter

The spark Interpreter does not connect to any schema by default. Users must write for connection either in the Initialization section or in a notebook's paragraph. This interpreter performs analytics on data present in Big data clusters in the Scala language. This requires additional configuration, which must be performed as a pre-requisite or as post-installation with the manual change of interpreter settings.

In spark interpreter, you can configure the cluster manager to connect, print the Read–eval–print loop (REPL) output, the total number of cores to use, etc.

To configure the spark interpreter variant, follow these steps:

1. On the Interpreter page LHS menu, select spark. The spark interpreter pane is displayed.

2. On Interpreter Settings page, expand **Interpreter Client Configurations** and click Edit icon for **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed.

> **NOTE** The user must select the pyspark Class Name.
> For example, org.apache.zeppelin.spark.SparkInterpreter.

3. Enter the following information in the spark interpreter variant pane as tabulated in the Table 13.

Table 13: spark interpreter

Field	Description
pgx.baseUrl	Enter the PGX Base URL. This is the location where the data is pushed. For example, `http://<HOSTNAME>:7007`

Table 13: spark interpreter

spark.executor.memory	Enter the amount of memory to use for the executor process. Executor memory per worker instance. For example, 512m and 32g. In Spark, the executor-memory flag controls the executor heap size (similarly for YARN and Slurm). The default value is 512MB per executor. In addition, the driver-memory flag controls the amount of memory to allocate for a driver, which is 1GB by default and should be increased in case you call a collect or take(N) action on a large RDD inside your application.
spark.master	Enter the cluster manager to connect. For example, `local[*]`
spark.yarn.archive	Enter the archive containing the required. Spark jars for distribution to the YARN cache make Spark runtime jars accessible from the YARN side.
spark.app.name	Enter the name of the application. For example, `Zeppelin`
zeppelin.spark.ui.hidden	Set to True or False.
zeppelin.spark.maxResult	Enter the maximum number of results that must be fetched.
spark.pyspark.python	Enter the Python binary executable for PySpark in both driver and executors. For example, `python`
zeppelin.spark.enableSupportedVersionCheck	Set to 'True' or 'False'.
args	Enter the Spark command-line args.
zeppelin.spark.useNew	Set to 'True' to use the new version of the SparkInterpreter.
zeppelin.spark.useHiveContext	Set to 'True' to use HiveContext instead of SQLContext.
zeppelin.spark.uiWebUrl	Overrides Spark UI default URL. Value should be a full URL `(http://{hostName}/{uniquePath})`
zeppelin.spark.printREPLOutput	Enter to print the REPL output.
spark.cores.max	Enter the total number of cores to use. NOTE: Empty value uses all available cores.
spark.driver.bindAddress	Hostname or IP address where to bind listening sockets.
zeppelin.interpreter.output.limit	Output message from interpreter exceeding the limit will be truncated.

4.2 Create a Credential

New credentials are created when database details are changed or updated. For example, change in Transparent Network Substrate (TNS) due to hostname change or compulsory periodic update of schema passwords.

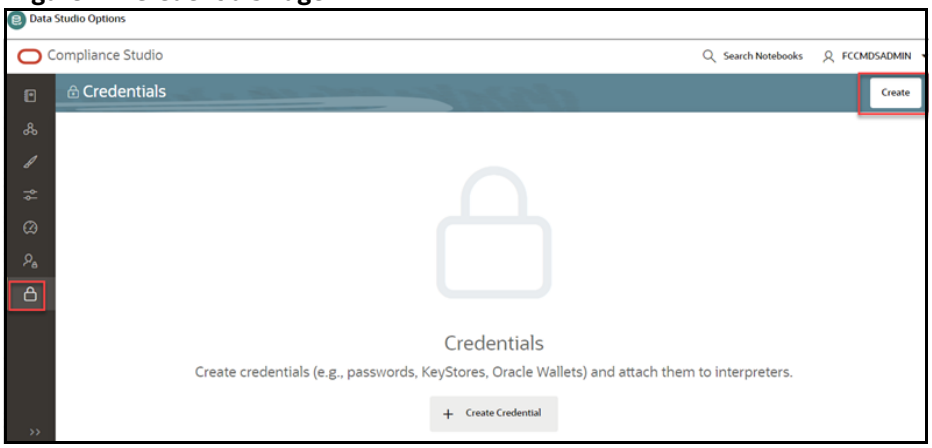
Oracle Wallet provides a simple and easy method to manage database credentials across multiple domains. It allows you to update database credentials by updating the Wallet instead of having to change individual data store definitions.

Use this section to add a new credential to the interpreters.

To create a new password credential for the wallet, follow these steps:

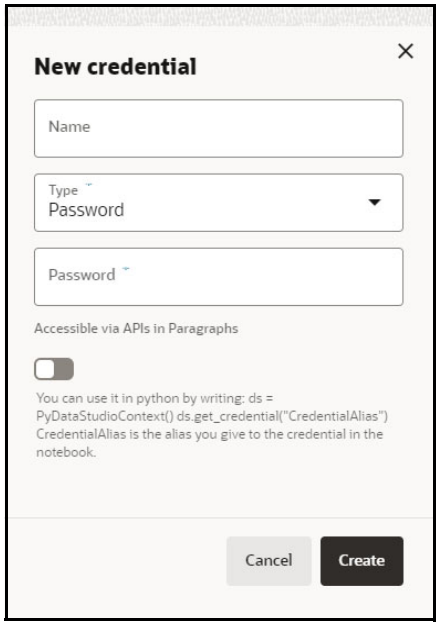
1. On the Compliance Studio workspace LHS Menu, click **Credentials**. The Credentials page is displayed.

Figure 14: Credentials Page



2. Click **Create**. The New Credential dialog box is displayed.

Figure 15: New Credential for Password



3. Enter the following information in the New credential dialog as tabulated in the Table 14.

Table 14: Create Credential dialog

Field	Description
Name	Enter the name for the password credential.

Table 14: Create Credential dialog

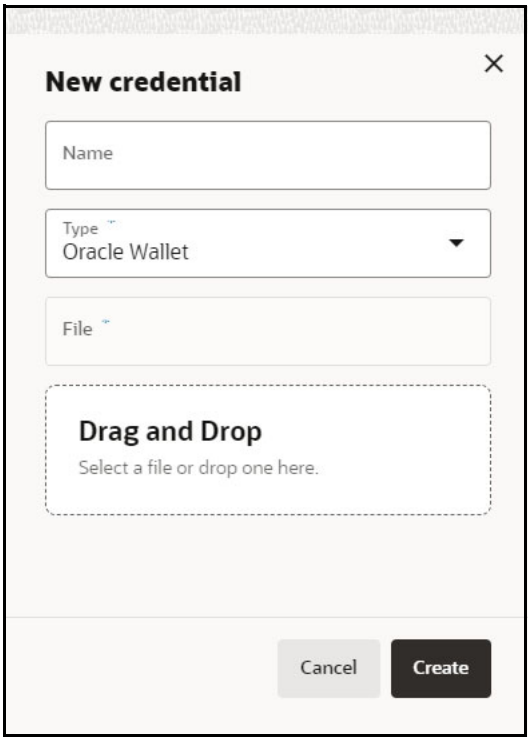
Type	From the drop-down list, select the Password type.
Password	Enter the wallet password for the password credential.
Accessible via APIs in Paragraphs	Move this toggle switch to right to enable this option.

4. Click **Create**. The password is created for the wallet and displayed on the Credentials page.

To create a wallet credential, follow these steps:

1. Click **Create**. The New Credential dialog box is displayed.

Figure 16: New Credential for Wallet



2. Enter the following information in the New credential dialog as tabulated in the Table 15:

Table 15: Create Credential dialog box

Field	Description
Name	Enter the name for the wallet credential.
Type	From the drop-down list, select the Oracle Wallet type.

Table 15: Create Credential dialog box

File	Upload the wallet zip file that includes the following files: • `tnsnames.ora` • `ewallet.p12` • `cwallet.sso` These files are available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/wallet` directory. **NOTE:** • The wallet file must be in .zip format. • The maximum file size allowed for the credential file is 128Kb.

3. Click **Create**. The wallet credential is created and displayed on the Credentials page.

4.3 Link Credentials

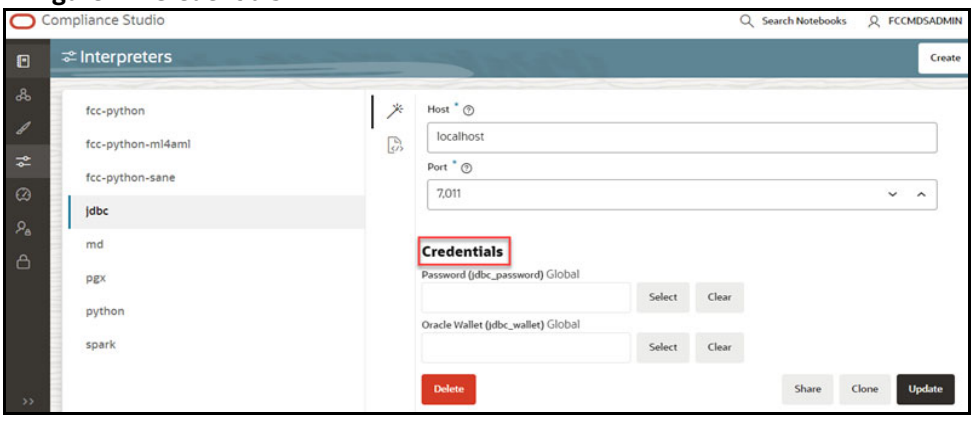
Compliance Studio provides secure and safe credential management. Examples for credentials are passwords, Oracle Wallets, or KeyStores. Use this section to link credentials (a wallet and a password) to the jdbc interpreter variant to enable secure data access. This linking enables the jdbc interpreter to securely connect to the specified Oracle Database. You can also create new credentials to connect to the new interpreter variants based on your requirement. For more information, see Create a Credential section.

> **NOTE** You can link credentials only for jdbc interpreters. The Credential section is enabled if an Interpreter variant can accept credentials.

To link ready-to-use credentials to the required interpreters, follow these steps:

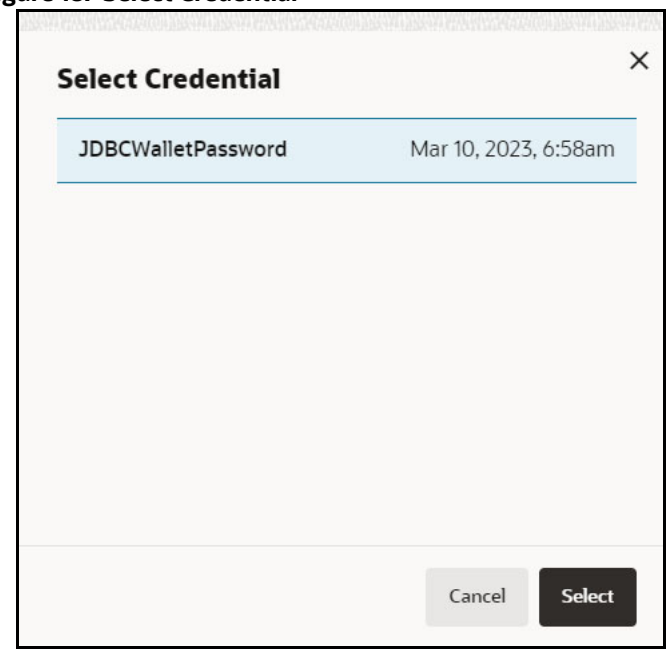
1. On the Interpreters page, select the required interpreters. For example, jdbc.
2. Navigate to the **Credentials** section.

Figure 17: Credentials



3. Click **Select** to select the Password (jdbc password) that you want to link to the Interpreter variant. The Select Credential dialog is displayed.

Figure 18: Select Credential



4. Select the required Password (jdbc_password) and click **Select**.
5. Click **Select** on the Credentials section to select the Oracle Wallet (jdbc_wallet) that you want to link to the Interpreter variant. The Select Credential dialog is displayed.
6. Select the required Oracle Wallet (jdbc_wallet) and click **Select**.
7. Click **Update** on the Credentials section to save the changes.

   The required password and Oracle Wallet are linked to the jdbc Interpreter.
8. Restart Compliance Studio.

4.4 Create an Interpreter Group

In Compliance Studio, you can either use a default interpreter group or create a new group for an interpreter. You can create more than one group for an interpreter. Multiple groups for an interpreter are created to connect different versions of interpreters (Python version: 3, Python version: 2) and connect a different set of users and database schema. For example, Compliance Studio schema, BD schema, etc.

To create a new interpreter group, follow these steps:

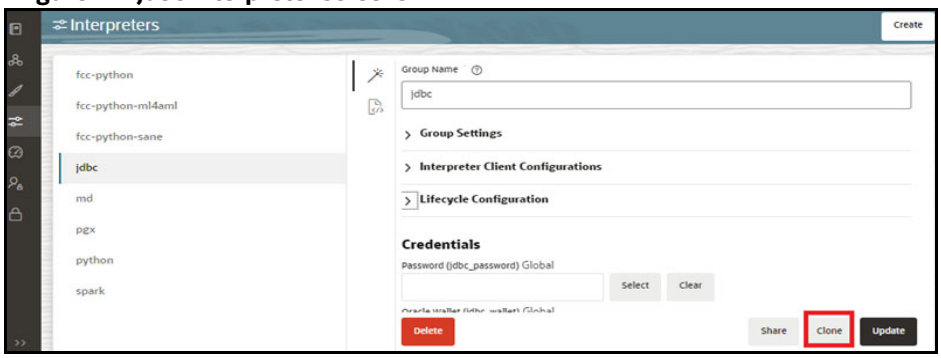
1. On the Interpreters page, click the required interpreters from the LHS list. For example, jdbc interpreter.
2. The default interpreter group is displayed on the RHS.
3. On the default interpreter, click **Clone** button to create a new group. The Create Interpreter Group dialog box is displayed.
4. Enter the Name for the new interpreter group. Click **Create**. A new group is created with a name, **<Interpreter Type>.<Group Name>.**
5. Provide the new schema details, such as the default.url, default.user, and default.password.

4.5 Create an Interpreter Variant

1. Log in to the Compliance Studio application.
2. Launch the **CS Production** Workspace.
3. Click on the **User Profile** drop-down list and select **Data Studio Options**.
4. Click **Interpreters**.

   By default, the Interpreters page lists all the available interpreters.
5. Click **jdbc** interpreter on the LHS. The default configured interpreter variant is displayed on the RHS:

Figure 19: jdbc interpreter screens



6. Click **Clone** on the RHS. The pop-up window displayed for the group name.
7. Enter the group name in the **Group Name** text box and click **Create**. The new group is created and displayed on LHS.
8. Click **<New group name>** on the LHS. The default configured interpreter variant is displayed on the RHS.

   You can modify the values in the interpreter properties in the JSON file or Wizard view.

4.6 Enable Additional Spark or PySpark interpreter

Interpreter variants do not apply to Spark or PySpark interpreters. Hence, you must enable an additional set of interpreters.

To enable an additional Spark or PySpark interpreter, see Enabling an Additional Spark or PySpark Interpreter chapter in the OFS Compliance Studio Installation Guide.

5 Schedule Scenario Notebook Execution

It is recommended to use the scheduler to execute the notebook in Batch.

Topics:

5.1 Prerequisites

> **NOTE** This is deprecated in the current release and will be removed in the future release.

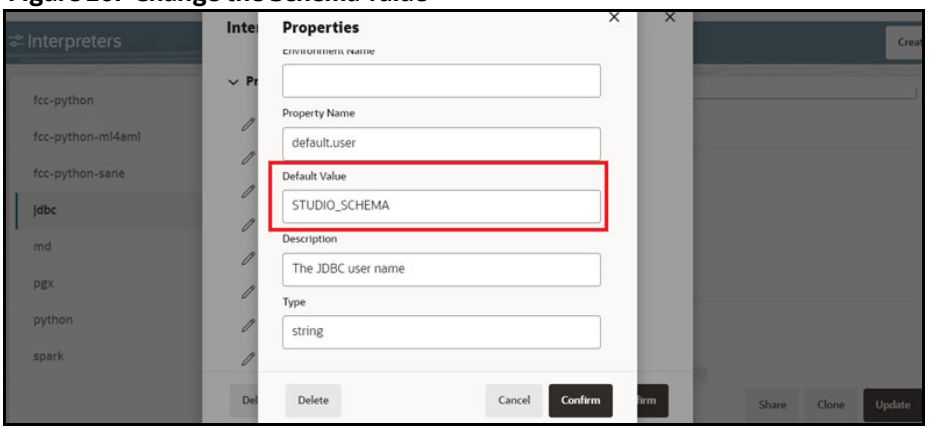
After installation, you need to create a new variant of the interpreter and change the schema from **STUDIO_SCHEMA** to **BD_SCHEMA** to execute Scenario notebooks.

To create a new variant and change the schema:

1. To create a new variant, see Create an Interpreter Variant section.
2. Click **<New group name>** on the LHS. The default configured interpreter variant is displayed on the RHS.

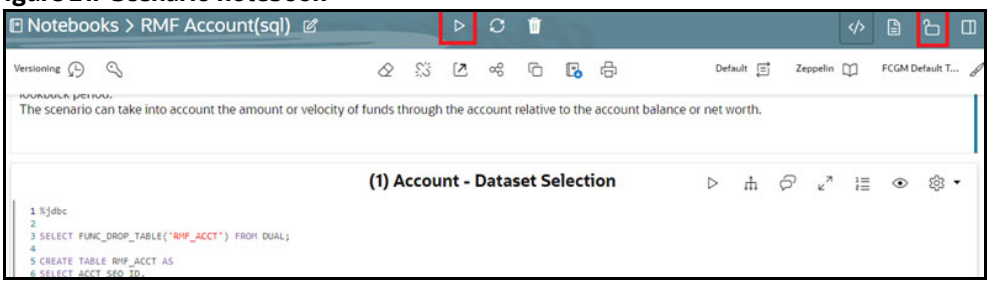
   You can modify the values in the interpreter properties in the JSON file or Wizard view.
3. On the Interpreter Settings page, expand **Interpreter Client Configurations** and click the Edit icon for **<Class Name> (zeppelin).** The Interpreter Client Configurations Window is displayed. Click **defaultuser** property. The property window is displayed.

Figure 20: Change the Schema value



4. On the property window, change the value from STUDIO_SCHEMA to BD_SCHEMA in the **Default Value** text box. Click **Confirm**.
5. Click **Confirm** and click **Update**.
6. On RHS, click on JSON view and copy the interpreter's name that is required to update the interpreter name under each paragraph in the scenario notebook.
7. Navigate to the Compliance Studio server with the same URL by changing the port to 7008. `(http://hostname:7008 from http://hostname:7001/cs/home )`
8. Open the scenario notebook (**RMF Account(sql)**), unlock the notebook, and replace it with the new interpreter name in each paragraph.

Figure 21: Scenario notebook



9. Click Run Paragraph's execute icon to execute the notebook.

5.2 Using Scheduler

To schedule a model and scenarios for execution using the scheduler, see the **Using Scheduler Service** section in the OFS Compliance Studio User Guide.

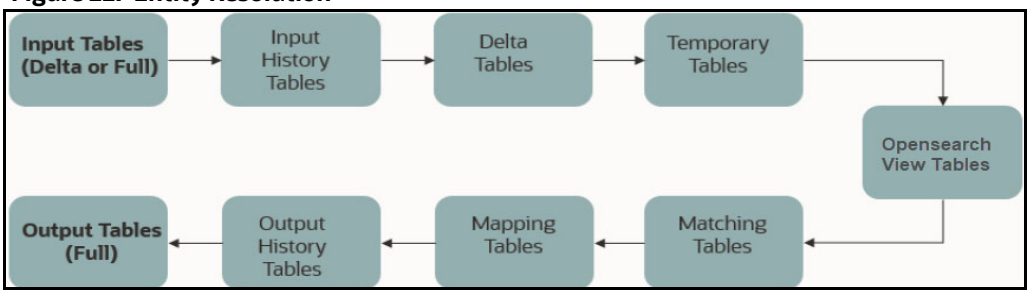
For more details, see Migrating the Data from ElasticSearch to OpenSearch section.

6 Entity Resolution

OFS Compliance Studio provides Entity Resolution (ER) capability. It allows firms to break through barriers in their data by gaining single views of their customers and their external entities and have the choice of monitoring them both under one consolidated Global Party.

OFS Compliance Studio Entity Resolution is a configurable process that allows data to be matched and merged to create contextual links in the global graph or resolve relational party records to a global party record as part of ingestion. OFS Compliance Studio has pre-built configurations supporting matching (or linking) in the FCGM and resolving entities in CSA for data being loaded into Financial Services Data Foundation (FSDF).

Figure 22: Entity Resolution



- **Comparison for Delta Processing**

  The first time Entity Resolution runs, it operates on the full data set. This means the initial run will take longer than subsequent runs after the initial processing where deltas (changed records) are calculated (regardless of whether full or delta data is populated in the input tables) so that matching happens only on new and changed records for improved performance.

- **Candidate Selection**

  Selection of candidates for matching. OpenSearch is a distributed search and analytical engine for all structured and unstructured data types in OFS Compliance Studio.

- **Matching**

  Matching rules are used to compare entities to identify pairs that refer to the same entity. It creates a probable link between entities by comparing the attributes of the entities.

  For example, deduplicating customers, resolving derived entities, or linking customers or derived entities to external data such as Panama papers or sanctions lists with different rules and thresholds.

  For more information on scoring methods, see the OFS Compliance Studio Matching Guide.

  For more information on creating, see the **Creating a Ruleset** section in the OFS Compliance Studio User Guide.

- **Grouping**

  It is used to Group (entity Ids or Customer Ids) based on similarity links between entities using matching rules and applying the merge rules on similarities. Once it is grouped, the system assigns the global party id to each Group.

> **NOTE** Grouping is an automatic process. Grouping will be based on the match edges without any configuration.

- **Merge Rules**

  Merging rules are used to group multiple entities or customers into a single global party based on the merge ruleset.

  For more information on creating the Merging Rules, see the **Creating a Ruleset** section in the OFS Compliance Studio User Guide.

- **Persisting**

  Records identified for merging will be collapsed into a single global party record, and a mapping from this global party record to the original party records will be created. Ongoing changes to the original party records may impact the global parties. For more details, see Using Wrapper Shell Script section.

- **Data Survival**

  When party records are identified for merging, a single output party record is created for the main or parent Dataset. Entity Resolution provides a mechanism to select the best data view from across the multiple-party records using attribute-by-attribute selection functions like Most Common or Maximum. It also provides a mechanism for transforming the child records stored in related tables, such as an address, email, or document ids.

  - **Merge and Split Global Parties**: Entity Resolution provides a mechanism to merge, split, create manually, and rearrange the entities for Global parties. Whenever there is a manual action (merge, split, create, rearrange) to the entities of a global party, the same data survival logic will be applied. See **Using the Merge and Split Global Parties** section in the OFS Compliance Studio User Guide on how to perform the actions.

  For more information on configuring the rules for attribute survival, see the Data Survival section.

> **NOTE**
>
> - When the records are not matched and not merged, they pass straight through and have a one-to-one mapping with the global party.
> - Where Entity has been resolved/unresolved, data origin is set to **EntRes** for all the records.
> - The Data Survival job cannot override the manual actions to a global party in batch mode.

Topics:

- Using Pre-configured Datasets and Rulesets
- FCCM out-of-the-box Entity Resolution Pipeline on FSDF
- Executing the ER Jobs
- Persisting the Data
- Entity Resolution Metadata

6.1 Using Pre-configured Datasets and Rulesets

6.1.1 Pre-configured Rulesets for Matching, Merging, and Data Survival

The application provides preconfigured rulesets/rules for Matching, Merging, and Data Survival for the following Entity Resolution pipeline:

- **CSA_8126**

> **NOTE**
> - The lower version pipelines are supported only if you are upgrading.
> - A set of seeded match rules are available which are used in the out-of-the-box ER pipeline.

See the **Creating Rulesets** section in the OFS Compliance Studio User Guide for creating and configuring rulesets.

6.1.2 Custom Rulesets for Matching

Compliance Studio provides custom rulesets for matching in the Entity Resolution. While creating any custom matching rulesets, the admin user has to make sure that the minimum value of weightage across matching attributes for across **RULES** should be updated in "result.bulkResultMinScore" parameter in the application.properties file in the below path.

- If Elastic Search is configured for Entity Resolution:
  - `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/matching-service-es/conf`
  - `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/matching-service-es/conf`
- If Open Search is configured for Entity Resolution:
  - `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/matching-service/conf`
  - `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/matching-service/conf`

> **NOTE** After the changes, restart Compliance Studio.

For example;

Attribute 1 – Weightage – 0.4

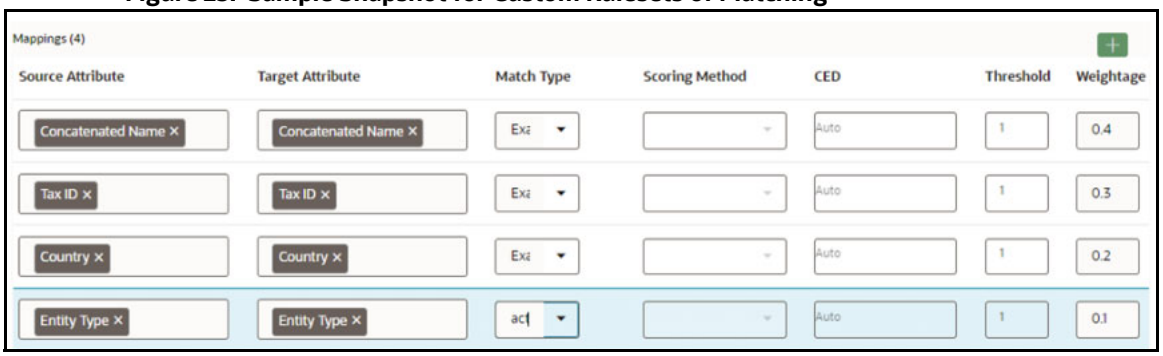
Attribute 2 – Weightage – 0.3

Attribute 3 – Weightage – 0.1

Attribute 4 – Weightage – 0.2

Then, the value parameter "result.bulkResultMinScore" should be set to 0.1.

Figure 23: Sample Snapshot for Custom Rulesets of Matching



6.2 FCCM out-of-the-box Entity Resolution Pipeline on FSDF

6.2.1 Pre-configured Entity Resolution Pipelines

The application is preconfigured to support the following Entity Resolution pipeline:

- **CSA_8126**

> **NOTE**
> - The lower version pipelines are supported only if you are upgrading.
> - Additional types of entity Resolution can be configured. For more information, see the Entity Resolution Metadata section.

For more information on how to run ER in different workspaces, see the **Run ER in Different Workspaces** section in OFS Compliance Studio Installation Guide.

6.2.2 Prerequisites for out-of-the-box ER Pipelines

1. The out-of-the-box ER pipeline requires a set of pre-staging tables to be available in the OFSAA staging area.
2. A pre-configured FSDF staging model.

   The pre-staging table definitions along with few ER specific tables are available in terms of a data model file which can be uploaded to OFSAA with the help of AAI's Data model management.

6.2.2.1 Creating Pre-Staging Tables in FSDF

Entity Resolution requires a set of pre-staging tables to be available in the OFSAA staging area and the pre-configured FSDF staging model.

The table definitions are available in terms of a data model file which can be uploaded to OFSAA with the help of AAI's Data model management.

> **NOTE** The `ER_81260.ODM` file is applicable only for Behavior Detection 8.1.2.6.0 version and CSA_8126 pipeline.

To upload the data model, follow these steps:

1. Copy `ER_81260.ODM` from `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/entity-resolution/datamodels` to `<AAI Application Server>/<FSDF_STG_INFODOM>/erwin/erwinXML`.
2. To upload the Data Model, perform the following:
   a. Model Upload Using **JSON/Erwin XML**.
   b. Select Upload Mode as **Sliced**.
   c. Select **Object Registration Mode** as **Incremental Object Registration**.
   d. Select **Upload File Type** as **JSON**.
   e. Select the **erwin XML** or **Database XML** or **ODM** file for upload from the drop-down list.

   Other options can be set to default and proceed to Upload.

For more information on uploading the Data Model, see the **Upload Business Model** section in the Oracle Financial Services Analytical Applications Infrastructure User Guide.

6.2.3 Load Data into Pre-Staging Tables

Data should be loaded into the pre-staging tables using an ETL process before Entity Resolution is run.

> **NOTE** Ensure the pre-staging tables are available in FSDF. See Creating Pre-Staging Tables in FSDF section.

You can load the records into Pre-staging tables every day using any one of the following methods:

- **Full Dataset/Full Load**: Load all the records with the same **fic_mis_date** and process all the records on the same **fic_mis_date**.
- **Delta Dataset/Delta Load**: Load only the modified, new records and records to be deleted based on **fic_mis_date** and process the identified new, modified and deleted records based on new **fic_mis_date**.

The **fic_mis_date** is the date on which the data is entered/loaded in the system.

For example,

- **Day 0**: 1000 records on 1st February (fic_mis_date)
- **Day 1**: 10 records added on 2nd February(fic_mis_date)

If a Full Dataset/Full load, **1000** records on **1st February** and all **1010** records are loaded and processed on **2nd February**.

If Delta load/Delta Dataset, **1000** records on **1st February** and additional **10** records are loaded and processed on **2nd February**.

> **NOTE** A full load needs to be run on the first day, and then on subsequent days, either full or delta data sets can be loaded into the **PRE** tables.
>
> Whether full or delta is run, the output tables will always contain full data for downstream applications to consume. This allows for the handling of deactivated parties due to matching and merging changes.
>
> If loading the **PRE** tables with delta only, records that should no longer be included will not be removed from the system. For this reason, a periodic full run may be required.

The following tables are pre-staging tables of out-of-the-box ER pipeline:

- **STG_PARTY_MASTER_PRE**: This table contains Customer details, name, DOB, etc. This table contains a person or organization that is a party of financial institutions. Here party refers to the customer, issuer and guarantor, etc. This table will hold the master list of parties and details like party name, age, education, profession, gender etc.
- **STG_DELETED_PARTIES_PRE**: This table contains parties id to be deleted from the Entity Resolution. If any available parties are to be removed explicitly from the system, then the STG_DELETED_PARTIES_PRE table should be populated with party ids (V_PARTY_ID) of the deleted parties against the corresponding FIC_MIS_DATE. The deleted parties will not be the part of matching process and final STG output tables of ER.
- **STG_PARTY_DETAILS_PRE**: This table contains additional Party details and is an extension of the STG_PARTY_MASTER_PRE table.
- **STG_ADDRESS_MASTER_PRE**: This table contains the master list of all addresses that are linked to the parties. The addresses in this table are mapped to one or more parties in the STG_PARTY_ADDRESS_MAP_PRE table using the V_ADDRESS_ID column.
- **STG_PARTY_EMAIL_MAP_PRE**: A party can have multiple email addresses. This table identifies all the email addresses that are associated with a party. Email Address is linked to a party via the purpose type for which this email address is used in relation to a party. For example, the purpose could be a Personal Email Address, Business Email Address, etc.
- **STG_PARTY_ADDRESS_MAP_PRE**: A party can have multiple addresses. This table identifies all the addresses that are associated with a party. The address is linked to a party via the purpose type for which this address is used about a party. For example, the purpose could be Mailing Address, Business Address, Home Address, etc.

> **NOTE**
> - There should not be double quotes ("") special characters in any attributes. Load to OpenSearch will not consider records containing the double quotes in any of the columns.
>
>   For example,
>
>   #15, Ground Floor, **"**VK Circle**,"** 1st Main Road**,** Bangalore.
>
>   VK Circle will not be considered as part of the address in the above address.

NOTE

- In the STG_PARTY_ADDRESS_MAP_PRE table, set the D_ADDRESS_END_DATE attribute to a date less than fic_mis_date if an address is to be deleted from the system. This will remove the address as part of the Entity Resolution batch run.

- **STG_PARTY_PHONE_PRE**: A party can have multiple phone numbers. This table identifies all the phone numbers that are associated with a party. The phone number is linked to a party via the purpose type for which this phone number is used in relation to a party. For example, Purpose could be Home Phone, Business Phone, Mobile Phone, etc.
- **STG_CUSTOMER_IDENTIFCTN_DOC_PRE**: This table stores the information regarding identification documents provided by customers. There should be a document associated with each Customer Identification Document record. Various documents submitted by the customer are identified by document type as BC- Certificate of Birth, BL- Business License, VR- Vehicle Registration Card or Title, VRC- Voter's Registration Card, etc.

6.2.4 Output Tables

The equivalent output tables exist in CSA according to the input tables for the respective pipelines.

For example, if the input table is **STG_PARTY_MASTER_PRE,** the output table will be **STG_PARTY_MASTER.** It is the same for FSDF 8124, 8125 and 8126.

After executing the Data survival Job, the output tables store the corresponding global party data.

NOTE

- By default, the output tables are available in FSDF. The purpose of the tables is the same as the input tables.
- Regardless of Full load or Delta load, the output tables contain the complete set of records with the current **fic_mis_date**. Such global parties can be removed from output tables where mappings have changed, and parties are deactivated.

The following are the output tables:

- STG_PARTY_MASTER
- STG_PARTY_DETAILS
- STG_PARTY_EMAIL_MAP
- STG_PARTY_ADDRESS_MAP
- STG_ADDRESS_MASTER
- STG_PARTY_PHONE_MAP
- STG_CUSTOMER_IDENTIFCTN_DOC

6.2.5 Entity Resolution Mapping Information

FCC_ER_MAPPING: It stores the mapping between Customer IDs in the input table **STG_PARTY_MASTER_PRE** and Global Party IDs in the output table **STG_PARTY_MASTER**.

The Table 16 describes column details in the FCC_ER_MAPPING.

Table 16: FCC_ER_MAPPING Details

Column Name	Description
V_GLOBAL_ID	It represents the global party id generated after Entity Resolution.
V_ENTITY_ID	It represents the original entity ids. For example, STG_PARTY_MASTER_PRE.V_PARTY_ID
F_LRI_FLAG	It indicates the state of a global id. The expected values are 'Y' or 'N'. 'Y'indicates active and 'N' indicates inactive.
D_CREATED_DATE	It stores the date and timestamp of a newly created Global Id from both ER batches and manual actions. **NOTE:** In case of **add** scenario, the **D_CREATED_DATE** column will be updated for the added entity in a global party. Existing entities will remain unchanged.
D_UPDATED_DATE	It stores the date and timestamp of an updated/deactivated Global Id from ER batches and manual actions. **NOTE:** In case of **split and merge,** the **D_UPDATED_DATE** column will be updated only for the deactivated global ids, and **D_CREATED_DATE** will be updated for the newly generated global ids.
V_ACTION	For information about V_ACTION column, see the Table 17.
V_PIPELINE_ID	It represents the implementation of Entity Resolution flow. For example, you have two pipeline ids for two versions of FSDF (i.e., 811 and 812).
V_COMMENT_ID	It stores the ID reference of the comments that are entered by a user while performing manual actions on a global party from **Manual Decision UI** and **Merge and Split UI**. This column will only store the **Id** and the respective comment will be stored in the **fcc_er_gp_comments** table.
F_OVERRIDE_FLAG	This flag controls whether to override the manual decision or not irrespective of the V_MD_FLAG value. By default it should be null.
V_MD_FLAG	It stores the state of the records upon which manual actions are taken **from Manual Decision UI** and **Merge and Split UI**. The expected values are: • MA - Manual Approved / Manual Action • PMA - Pending Manual Approval • MR - Manual Rejection **NOTE:** The value in this column will be NULL for the records generated from Entity Resolution batches. The values will be populated for the entities upon which any manual action has been taken from **Merge and Split UI**.

Table 16: FCC_ER_MAPPING Details

Column Name	Description
N_RUN_SKEY	It signifies the execution identifier of an Entity Resolution batch. This identifier will be updated for all the impacted entities in an ER batch. For example: When a new global party is created, a new entity is added to an existing global party, an existing global party is split, existing global parties are merged or an existing global party is deactivated.

The Table 17 describes **V_ACTION** column in the **FCC_ER_MAPPING.**

Table 17: V_ACTION Details

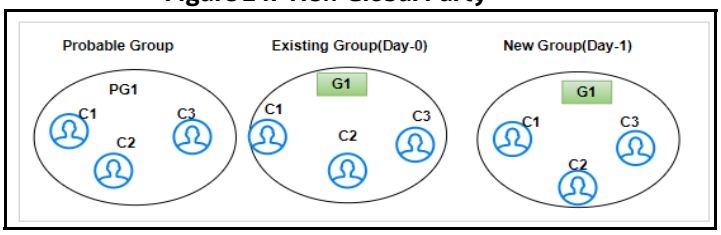
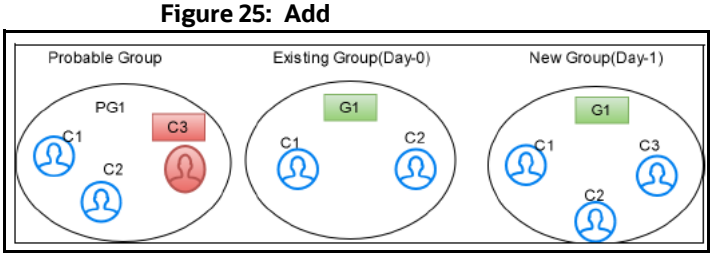
Value	Description
	Batch Execution
new global party	On the first run of ER batches, the value of the V_action column will be a **new global party** for all the records. In subsequent batches, if there is no change in the existing entities, it will remain the same as new global party. **Figure 24: New Global Party**  For example, G1 has C1, C2 and C3 entities. After the Day 1 batch execution, if there is no change in the existing group. Still, G1 has C1, C2 and C3 entities with the same global id.
add	If a new entity is available and matches the existing group, then it is defined as **add** in the V_ACTION column for a newly added entity. If a new entity matches the existing group, it will be added to the existing group and assigned the same global id. **Figure 25: Add**  For example, G1 has C1 and C2 entities. After the Day 1 batch execution, if C3 entity matches with C1 or C2 then C3 will be added to the existing group G1 with the same global id.

Table 17: V_ACTION Details

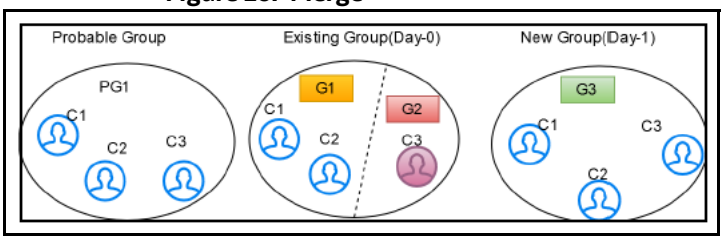
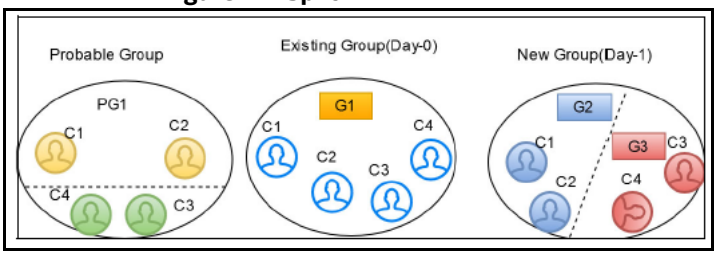
Value	Description
merge	If there is a data change in the entity of a different group and it merges with another group, then it is defined as **merge** in the V_ACTION column for the merged entities. The changed entities can be merged with an existing group with new global id is assigned and the previous global id will be de-activated. **Figure 26: Merge**  For example, G1 has C1 and C2 entities and G2 has a C3 entity. After the Day 1 batch execution, if C3 entity matches with an existing group then C3 will be merged into the existing group with a new global id. The V_ACTION column for G3 will merge and G1 and G2 will be deactivated.
split	If there is a data change in the existing group entity which does not matches with other entities of an existing group; then it is defined as **split** in the V_ACTION column for the split entities. The changed entities can be split into a new group and a new global id is assigned to each. **Figure 27: Split**  For example, G1 has C1, C2, C3 and C4 entities. After the Day 1 batch execution, if C3 and C4 entities are not matched with the existing entities of the group then C3 and C4 will be split into a new group. G2 has C1 and C2 entities and G3 has C3 and C4 entities with a new global id assigned to each group. The V_ACTION column for G2 and G3 will split and G1 will be deactivated.

Table 17: V_ACTION Details

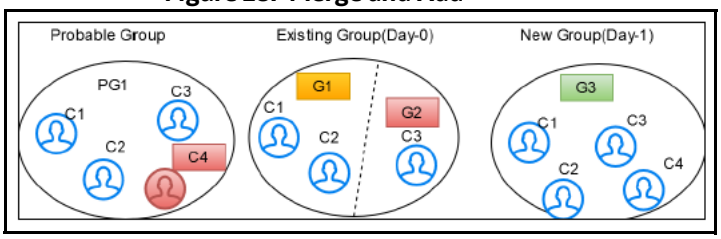
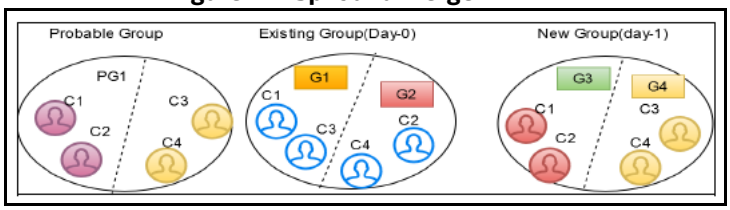
Value	Description
merge and add	If there is a data change in the existing group and a new entity is available, which also matches with the existing group; then it is defined as **merge and add** in the V_ACTION column for the updated and new entities. All the entities are grouped into a single group with a new global id. **Figure 28: Merge and Add**  For example, G1 has C1 and C2 entities, G2 has C3 entity. After the Day 1 batch execution, if C4 entity is added newly and C3 entity got changed then common entities are merged into a single group and a new entity is added to the group with a new global id (G3 has C1, C2, C3, and C4 entities). The V_ACTION column for G3 will merge and add, G1 and G2 will be deactivated.
split and merge	If there is a data change in the entity of the first group that matches with another entity of the second group and also an entity from the second group matches with any entity of first group; then it is defined as **split and merge** in the V_ACTION column for affected entities. The changed entities can be split and merged into a new group with a new global id is assigned to each group. **Figure 29: Split and Merge**  For example, G1 has C1 and C3 entities and G2 has C2 and C4 entities. After the Day 1 batch execution, if C1 matches with C2 and C3 matches with C4 then C2 and C4 will be split separately and merged with C1 and C3 respectively. G3 has C1 and C2 entities and G4 has C3 and C4 entities with a new global id assigned to each group. The V_ACTION column for G3 and G4 will split and merge and G1 and G2 will be deactivated.

Table 17: V_ACTION Details

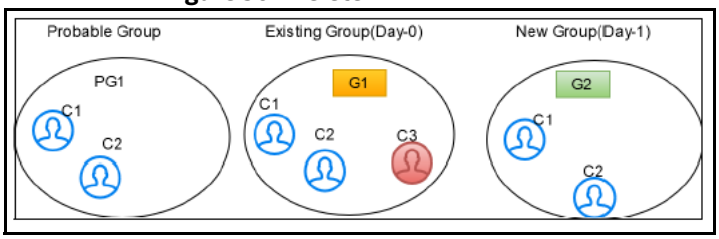
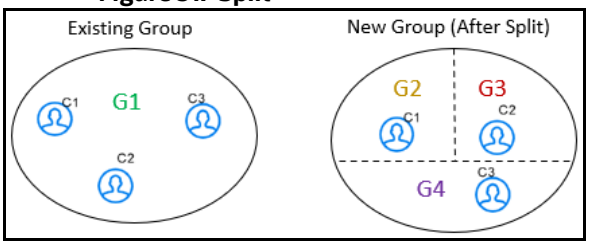
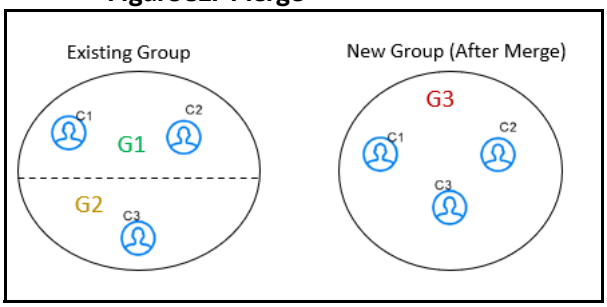
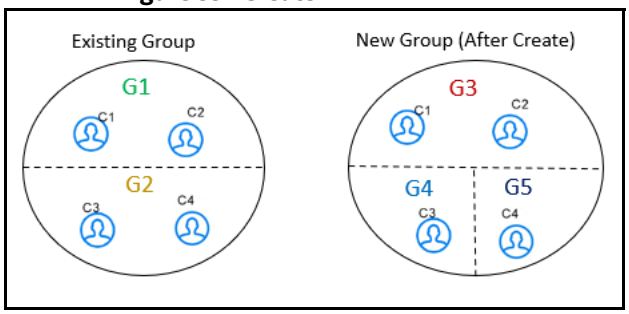
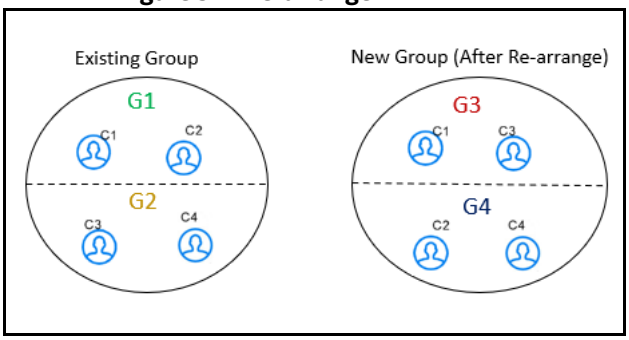
Value	Description
delete	During batch execution, if an entity is unavailable in the existing group, it is defined as **delete** in the V_ACTION column and a new global id is assigned to the remaining entities. **Figure 30: Delete**  For example, G1 has C1, C2, and C3 entities. After the Day 1 batch execution, if C3 is deleted from the existing group then G2 has C1 and C2 entities with a new global id. The V_ACTION column for G2 will delete and G1 will be deactivated.
Manual Action	
split	You can split the entities into different groups with new global ids assigned to each. **Figure 31: Split**  For example, G1 has C1, C2, and C3 entities. After split, G2 has C1, G3 has C2 and G4 has C3 with new global ids assigned to each group. The V_ACTION column for G2, G3 and G4 will split and G1 will be deactivated.
merge	You can merge the different entities into a single group with a new global id is assigned. **Figure 32: Merge**  For example, G1 has C1 and C2 entities, G2 has C3 entities. After merge, G3 has C1, C2, and C3 entities with a new global id. The V_ACTION column for G3 will merge and G1 will be deactivated.

Table 17: V_ACTION Details

Value	Description
create	You can create a new entity from the existing group with a new global id is assigned. **Figure 33: Create**  For example, G1 has C1 and C2 entities, G2 has C3 and C4 entities. After create, G3 has C1 and C2 entities, G4 has C3 entity and G5 has C4 entity with new global ids assigned to each group. The V_ACTION column for G3, G4 and G5 will create and G1 will be deactivated.
re-arrange	You can re-arrange the entities from another group with a new global id is assigned. **Figure 34: Re-arrange**  For example, G1 has C1 and C2 entities, G2 has C3 and C4 entities. After re-arrange, G3 has C1 and C3 entities and G4 has C2 and C4 entities with new global ids assigned to each group. The V_ACTION column for G3 and G4 will re-arrange and G1 and G2 will be deactivated.

6.2.6 Consolidated Information of the Resolved Entities

FCC_ER_OUTPUT: It is a subset of all staging tables and stores specific column details from each staging output table.

6.3 Executing the ER Jobs

You can execute the following available jobs either manually or automatically a using wrapper shell script for Entity Resolution in a specified sequence:

1. Create Index and Load the Data (`ER_Create_And_Load_Data_Into_Index.sh`)

2. Perform Matching (`ER_Run_Bulk_Similarity_Job.sh`)
3. Data Survival (`ER_Run_Data_Survival_Engine.sh`)
4. Load Data in FCC_ER_OUTPUT Table (`ER_Run_Full_Data_Output.sh`)

> **NOTE** You can proceed with data movement from staging to FCDM during **Load Data in FCC_ER_OUTPUT Table** execution.

Before running the ER jobs, the user should ensure the following:

- Create an ER Schema
- Grant Permission to ER Schema
- Add ER Schema Wallet details
- Update `resources.xml` with ER Schema details

See the **Entity Resolution** section in the OFS Compliance Studio Installation Guide.

After installation, the user can follow the same steps in Configure the `resources.xml` for Multiple ER Schemas in OFS Compliance Studio Installation Guide to create additional ER schemas.

> **NOTE** You can use only one ER schema per **pipelineid** for each FSDF version, and the same ER schema cannot be used with other **pipelineid** for any ER job execution.

6.3.1 Create Index and Load the Data

> **NOTE** Ensure you have configured the **Logstash** parameter as **true** (`index.logstash-conf.apply`) in load-to-open-search `application.properties` to load the data from Database.

6.3.1.1 Job

`ER_Create_And_Load_Data_Into_Index.sh` performs the following:

- It creates all the output tables required at the different stages of Entity resolution tasks.
  - Input to this job will be pipeline id as an argument so that all the tables related to that pipeline ID will be created.
  - Index view table, Matching output table, Manual matches output table, Merge Map output table, Manual map merge output table, final dataset output tables. This task will create all these tables.

- It creates the index for the given Dataset and loads the data into the index table based on values provided in the **index.pipeline-id** argument.

> **NOTE** In systems where the delta is already derived by means of other techniques/ processes and the system is sure about the nature of data as a "true delta"; it is possible to skip the delta-computation within ER for faster turnaround in Create Index and Load the Data Job. In such cases, the input from PRE tables is considered to be the actual delta. This could be achieved by setting a batch parameter value accordingly.
>
> To skip delta computation, the "deltaComputed" parameter in <job1_script script name> should be set to 'true' (including single quotes). Any input from _PRE tables is assumed to be delta (modified/new records). Note that deltaComputed is considered only when Create Index and Load the Data job is executed with the load type as DeltaLoad.
>
> Previous execution _CHUNKED (example: H$STG_PARTY_MASTER_PRE_101_CHUNKED_1) tables are not required while executing Create Index and Load the Data job with deltaComputed as 'true'. If you are planning to execute Create Index and Load the Data job with deltaComputed as true for every time/always, the chunk creation during Create Index and Load the Data job can be skipped by setting the F_CREATE_CHUNKS value as false in the FCC_ER_CONFIG table in FSDF schema.

6.3.1.2 Steps

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin`.
2. Run the following command:

```
nohup ./ER_Create_And_Load_Data_Into_Index.sh "<pipelineid>"
"<ERSchemaId>" "<Load Type>" "<FIC_MIS_DATE>" "<FSDFVersion>"
"<Batch_group>" "Source_Batch" "<Data_Origin>" "<Run_Type>" &
```

> **NOTE**
> - "Batch_group" refers to the table FCC_PROCESSING_GROUP in the Compliance Studio schema.
> - "<Source_Batch>" and "<Data_Origin>" are not relevant now as execution parameters and they are added for future use.

For example, you can use the following commands for CSA_8126 pipeline.

FSDF 8126 version: `nohup ./ER_Create_And_Load_Data_Into_Index.sh "CSA_8126" "ER_SCHEMA_PP_ALIAS" "FullLoad" "20151210" "8126" "CSA_812" "CSA_812" "US" "RUN" &`

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

6.3.1.3 Additional Configurations

To enhance the efficiency of history maintenance and delta processing, perform the following:

> **ATTENTION** The default values are based on hardware configurations (**Eight-core CPU** and **64 GB RAM**) and delta size (**ten million** records).

1. Log in to ER Schema.

2. Navigate to the **FCC_ER_CONFIG** table and configure the **V_PARAM_VALUE** value based on the DB performance.

   You can modify the following parameters in the table with **Pipeline_ID** as **CSA_812** before executing the job based on your volume of data:

   - **PREV_CHUNKS**: The number of chunks of history tables during the last execution of the job. By default, it is set to **10**. You should not modify the value. This parameter value will be modified automatically when you modify the **TODAY_CHUNKS** value and execute the job successfully.
   - **TODAY_CHUNKS**: The number of chunks of history tables for the current day/date. By default, it is set to **10**. You can modify this value to change the number of chunks to be processed in the respective history tables when the job execution time is longer.

   > **NOTE** Here the chunk value is based on the volume of data being processed. It is recommended to increase the value to **15** when the volume of data being processed is more than **50** million records and monitor the performance.

   - **MAX_JOBS**: Maximum number of jobs to schedule in the Database at a time. By default, it is set to **35**. You can modify this value to reduce job execution time.

   > **NOTE** Increasing this value only when the Database is not shared for the other processes is recommended.

   - **CHUNK_SIZE**: The number of records to process in one chunk. It is set to **2000000** (2 million records in each chunk) by default.

   > **NOTE** It is recommended to retain the default value. You can decrease it to a lower value for better performance only when the server (where the Database is installed) has less than **eight** CPUs.

   - **MAX_HISTORY_PARTITIONS**: The maximum number of partitions to be retained in the H$ tables.

     The minimum allowed value is **1**. If the user provides a value less than this number, then it will retain 1 partition by default.

     The maximum allowed value is **3**. If the user provides a value greater than this number, then it will retain 3 partitions by default.

   > **NOTE** The value for **MAX_HISTORY_PARTITIONS** parameter should be a positive integer. The valid range is 1 to 3.

   - **F_ER_DS_SUBSEQUENT_BATCH**: This parameter is used when the user approves a record from the **Pending - System Requests** tab of the **Merge & Split Global Entities** UI.

     The valid values are True and False. By default, the value is set to False.

     If it is set to True, then data survival is applied to the approved system request on the subsequent day's batch run.

     If it is False, then the data survival is applied immediately upon approving the system request from the UI.

- **ER_DS_SYSTEM_PENDING_MAX_NO_REC**: This is the maximum number of records which can be approved from the **Pending - System Requests** tab of the **Merge & Split Global Entities** UI at once.

  By default, the value is set to 10. The valid values range is 1 to 100.

  If the user tries to approve more records than the number mentioned for this parameter, an alert is displayed to the user on the UI.

  This is applicable only when **F_ER_DS_SUBSEQUENT_BATCH** is set to False.

  If **F_ER_DS_SUBSEQUENT_BATCH** is set to True, this count is overridden and all the records from the UI can be approved using the **Approve All** button.

3. Save the changes.

6.3.1.3.1 Profiler Table

The table, **ER_PERFORMANCE_TIME_PROFILER** in ER schema, helps the user track the current status of the batch and debug performance issues.

The **ER_PERFORMANCE_TIME_PROFILER** table stores the queries that are executed during delta processing. Here are a few parameters that help to debug which query is failed:

- **V_TABLE_NAME**: It stores the table name for which the query was executed.
- **N_CHUNK**: It stores the chunk number that is executed.
- **D_STARTTIME**: It stores Database time when the query starts to execute.
- **D_ENDTIME**: It stores the Database time when the query got executed.
- **V_TOTAL_TIME**: It stores the duration of the query execution.
- **V_STATUS**: Current status of the query. The values are **START**, **RUNNING**, or **END**.
- **V_QUERY**: It stores the query that was executed.
- **N_RUN_SKEY**: It stores the **runSKey** value of the currently executing job.

To check the query status, perform the following:

1. Log in to ER Schema.
2. Run the following command:

   ```
   SELECT * FROM ER_PERFORMANCE_TIME_PROFILER WHERE N_RUN_SKEY =
   <CURRENT_RUNSKEY>
   ```

   For example,

   ```
   SELECT * FROM ER_PERFORMANCE_TIME_PROFILER WHERE N_RUN_SKEY = 200
   ```

3. Check **V_STATUS**. The status other than the **END** value indicates the failed query.

> **NOTE** If any unexpected failure occurs, there is no explicit cleanup activity to be performed in the **Create Index and Load Data** job as it is automatically taken care of re-run of the job.

6.3.1.3.2 Cleanup Steps for Job Termination

Execution of manual cleanup is required in case of any fatal user error, such as executing the job against incorrect FIC_MIS_DATE, except for any semantic and logic validation taken. After contacting

My Oracle Support, you can perform cleanup steps. For more information about cleanup steps, see the Cleanup Steps When the Create Index and Load Data Job Terminated Manually section.

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

6.3.2 Perform Matching

6.3.2.1 Job

The **ER_Run_Bulk_Similarity_Job.sh** triggers the matching engine to generate the matches in the match output table for rulesets saved against a pipeline-id argument for fetching rulesets.

6.3.2.2 Steps

> **NOTE** Make sure to check the fcc_er_matching and fcc_er_manual_match tables before proceeding. Check the logs accordingly if there are no records in fcc_er_matching and fcc_er_manual_match generated.

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin`.
2. Run the following command:

```
nohup ./ER_Run_Bulk_Similarity_Job.sh "<pipelineid>"
"<ERSchemaId>""<Match Type>" "<Batch_group>" "<Run_Type>" &
```

> **NOTE** "Batch_group" refers to the table FCC_PROCESSING_GROUP in the Compliance Studio schema.

For example, you can use the following commands for CSA_8126 pipeline.

FSDF 8126 version:

```
nohup ./ER_Run_Bulk_Similarity_Job.sh "CSA_8126"
"ER_SCHEMA_PP_ALIAS" "FullLoad" "CSA_812" "RUN" &
```

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

> **NOTE** If the Bulk Similarity Edge job fails internally due to Incorrect schema details and then returns a success message. You can check the log file in `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs` for more details on the failure.

6.3.2.3 Matching Output

The results of the ER matching are stored in the following tables:

- **FCC_ER_MATCHING**: The results that exceed the automatic threshold limit are stored.
- **FCC_ER_MANUAL_MATCH**: The results between the automatic and manual thresholds are stored.

You can see the following details for the above tables:

- **SCORE**: Score of the match between Source and Target Entity
- **MATCH_DESCRIPTION:** Describes the attributes responsible for matching
- **SRC_DESC**: Describes attributes of Source considered for matching

- **TRG_DESC**: Describes attributes of Target considered for matching
- **V_PIPELINE_ID**: Describes the Pipeline Id of ER Type
- **N_RULESET_ID**: Describes the Ruleset responsible for matching
- **SRC_ORIGINAL_ID**: Describes the unique identifier for the Source entity
- **TRG_ORIGINAL_ID**: Describes the unique identifier for the Target entity

6.3.2.3.1 Cleanup Steps for Job Termination

Execution of manual cleanup is required in case of any fatal user's error. After contacting My Oracle Support, you can perform cleanup steps. For more information about cleanup steps, see the Cleanup Steps When the Bulk Similarity Job Terminated Manually section.

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

6.3.3 Data Survival

NOTE

- Ensure only one preconfigured ruleset is enabled for Merging and Data Survival. See the Pre-configured Rulesets for Matching, Merging, and Data Survival section**.** The job will be failed with a unique constraint error if multiple rulesets are enabled.
- If there is a unique constraint error in the **STG_CUSTOMER_IDENTIFCTN_DOC** table during the Data survival job, you should ignore the below error.

```
2022-11-04 11:47:56,560 - globalparty.util.GlobalPartyUtils - 238 [ERROR]: Error ORA-00001: unique constraint (ER10_0805_PERF.XPKSTAGE_CUSTOMER_IDENTIFICATION_DOCUMENT_2) violated at row offset 10135

NoneType: None

2022-11-04 11:47:56,560 - globalparty.util.GlobalPartyUtils - 238 [ERROR]: Error ORA-00001: unique constraint (ER10_0805_PERF.XPKSTAGE_CUSTOMER_IDENTIFICATION_DOCUMENT_2) violated at row offset 10143

NoneType: None

2022-11-04 11:47:56,560 - globalparty.util.GlobalPartyUtils - 238 [ERROR]: Error ORA-00001: unique constraint (ER10_0805_PERF.XPKSTAGE_CUSTOMER_IDENTIFICATION_DOCUMENT_2) violated at row offset 10145

NoneType: None

2022-11-04 11:47:56,561 - globalparty.util.GlobalPartyUtils - 238 [ERROR]: Error ORA-00001: unique constraint (ER10_0805_PERF.XPKSTAGE_CUSTOMER_IDENTIFICATION_DOCUMENT_2) violated at row offset 10151
```

6.3.3.1 Job

The **ER_Run_Data_Survival_Engine.sh** job performs the following:

- **ER_Merge_Engine**: It triggers the merge engine, and records will be inserted in the mapping table based on the merge rules saved against the pipeline id argument.
- **ER_Data_Survival_Engine**: It triggers the data survival engine, and final outputs will be stored in tables based on the dataset survival rule stored against pipeline id.

6.3.3.2 Steps

> **NOTE** To re-run this job after a failure, the value of the **n_run_status** column in the **fcc_batch_run** table in Compliance Studio Schema should be changed to **4** for the respective **n_run_skey**.

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin.`
2. Run the following command:

```
nohup ./ER_Run_Data_Survival_Engine.sh "<pipelineid>" "<ERSchemaId>"
"<ERSchemaName>" "<Batch_group>" "<Match_Type>" "<FIC_MIS_DATE>"
"<Run_Type>" &
```

> **NOTE** "Batch_group" refers to the table FCC_PROCESSING_GROUP in the Compliance Studio schema.

For example, you can use the following commands for CSA_8126 pipeline.

FSDF 8126 version: `nohup ./ER_Run_Data_Survival_Engine.sh "CSA_8126" "ER_SCHEMA_PP_ALIAS" "ER_SCHEMA_PP" "CSA_812" "FullLoad" "20151210" "RUN" &`

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

> **NOTE**
> - The user should not have **Type** "Distinct" and "All**"** together with other columns that return unique values in child tables.
> - If the Batch, Backup, and Recovery processes fail when you execute the `ER_Run_Data_Survival_Engine.sh`, you need to re-run the same job again to ensure the Data is available in Archive only for the Mapping table (**FCC_ER_MAPPING**).
> - To increase/decrease the execution efficiency according to the server size using **ER_THREADS** and **ER_BATCH_SIZE** parameters, perform the following:
>   - Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin`
>   - Open the `ER_Run_Data_Survival_Engine.sh` and set the following parameters:
>   - `export ER_THREADS=<No of threads>`
>   - `export ER_BATCH_SIZE=<Batch Size>`

NOTE Example:

- `export ER_THREADS=4`
- `export ER_BATCH_SIZE=10000`

3. Validate to ensure Global party IDs are generated for the Entities in the following Staging Output tables after executing the job:
   - `STG_PARTY_MASTER`
   - `STG_PARTY_DETAILS`
   - `STG_PARTY_EMAIL_MAP`
   - `STG_PARTY_PHONE_MAP`
   - `STG_ADDRESS_MASTER`
   - `STG_PARTY_ADDRESS_MAP`
   - `STG_CUSTOMER_IDENTIFCTN_DOC`

NOTE Data Survival process expects the above STG tables to retain the snapshot of the previous **FIC_MIS_DATE** to complete the process successfully.

6.3.3.2.1 Cleanup Steps for Job termination

Execution of manual cleanup is required in case of any fatal user's error. After contacting My Oracle Support, you can perform cleanup steps. For more information about cleanup steps, see the Cleanup Steps When the Data Survival Job Terminated Manually section.

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

6.3.3.3 Properties for Global Party ID Persistence

When global parties change (parties are added or removed), the system can be configured to either create a new global party id or to keep one of the existing ids depending on need to preserve global party in downstream systems.

The fcc_er_guid_persist_config table contains the configuration for Global Party ID Persistence.

Table 18 describes column/flag deatils in the FCC_ER_GUID_PERSIST_CONFIG.

Table 18: FCC_ER_GUID_PERSIST_CONFIG Details

Column Name/Flag	Description
V_ACTION	It represents the different actions that can be performed on the Global Party ID. The possible actions are: • add • delete • merge • split and merge • merge and add • split
F_PERSIST_GUID	This flag represents whether the Global Party ID should be persisted or not whenever it undergoes change. The valid values are Y and N. The GUID is persisted if the flag is set to Y for the particular action.
F_MANUAL_APPROVAL	This flag represents manual approval is required when GUID undergoes change. The valid values are Y and N. If the flag is set to Y, then user gets the request to approve the changes in the UI. For more information, see the **Pending - System Requests Tab** section in the OFS Compliance Studio User Guide.
F_DEFAULT_VALUE	This flag represents the default value that will override the values present in the F_PERSIST_GUID and F_MANUAL_APPROVAL flags.

NOTE

- Only the flags in F_PERSIST_GUID and F_MANUAL_APPROVAL should be modified. F_DEFAULT_VALUE should not be modified for any action.
- For add and delete actions, the GUID always persists irrespective of the user input in the F_PERSIST_GUID flag.
- For delete action, manual approval is not required irrespective of the user input provided in the F_MANUAL_APPROVAL flag.
- If F_PERSIST_GUID and F_MANUAL_APPROVAL flags for the split action are set to Y and Y respectively, then flags for split and merge action will also be considered as Y and Y regardless of the user input. Similarly, If F_PERSIST_GUID and F_MANUAL_APPROVAL flags for the split and merge action are set to Y and Y respectively, then flags for the split action will also be considered as Y and Y regardless of the user input.

The following image shows default configuration of the fcc_er_guid_persist_config table.

Figure 35: fcc_er_guid_persist_config table

	V_ACTION	F_PERSIST_GUID	F_MANUAL_APPROVAL	F_DEFAULT_VALUE
1	add	Y	N	Y-
2	delete	Y	N	Y-N
3	merge	Y	Y	(null)
4	split and merge	Y	Y	(null)
5	merge and add	Y	Y	(null)
6	split	Y	Y	(null)

6.3.4 Load Data in FCC_ER_OUTPUT Table

6.3.4.1 Job

The **`ER_Run_Full_Data_Output.sh`** job executes the SQL procedure that joins the following staging output tables and populates data for the split and merge UI:

- `STG_PARTY_MASTER`
- `STG_PARTY_DETAILS`
- `STG_PARTY_EMAIL_MAP`
- `STG_PARTY_PHONE_MAP`
- `STG_ADDRESS_MASTER`
- `STG_PARTY_ADDRESS_MAP`
- `STG_CUSTOMER_IDENTIFCTN_DOC`

> **NOTE** If you want to perform slicing for the initial input data to run **Day 0** batch, it is recommended to run `ER_Create_And_Load_Data_Into_Index.sh`, `ER_Run_Bulk_Similarity_Job.sh`, and `ER_Run_Data_Survival_Engine.sh` jobs for all slices. The Output Tables are expected to have the resolved entities at the end of this process. At this point, `ER_Run_Full_Data_Output.sh` job can be executed for bringing the entire data across all slices into the output table.

6.3.4.2 Steps

> **NOTE** To re-run this job after a failure, the value of the **n_run_status** column in the **fcc_batch_run** table in Compliance Studio Schema should be changed to **6** for the respective **n_run_skey**.

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin`.
2. Run the following command:

```
nohup ./ER_Run_Full_Data_Output.sh "<pipelineid>" "<ERSchemaId>"
"<FIC_MIS_DATE>" "<Batch_group>" "<Load_Type>" "<Run_Type>" &
```

> **NOTE** “Batch_group” refers to the table FCC_PROCESSING_GROUP in the Compliance Studio schema.

For example, you can use the following commands for CSA_8126 version:

FSDF 8126 version : `nohup ./ER_Run_Full_Data_Output.sh "CSA_8126" "ER_SCHEMA_PP_ALIAS" "20151210" "CSA_812" "FullLoad" "RUN" &`

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

3. Validate specific column details are loaded in **FCC_ ER_OUTPUT** table from each staging output table for the Entities after executing the job.

6.3.4.2.1 Cleanup Steps for Job termination

Execution of manual cleanup is required in case of any fatal user's error. After contacting My Oracle Support, you can perform cleanup steps. For more information about cleanup steps, see the Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually section.

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

6.3.5 Initial Run for High Volume Data

The initial run (Day 0) of Entity Resolution on a high volume of data is expected to take a longer time and more reStores based on the performance. For an efficient initial run (Day 0), you can run the utility scrip to a faster turn-around time for individual batches as the load is moderately low. See Data Slicing Utility Script for more details.

6.3.6 Status Codes

The **fcc_batch_run** table in Compliance Studio Schema explains the status codes generated for ER jobs. See the status codes in **n_run_status** column for respective **n_run_skey** values.

Table 19 lists the ER job status codes:

Table 19: ER Job Status Codes

ER Job Name	During Execution	Success	Failure
ER_Create_And_Load_Data_Into_Index.sh	1	2	11
ER_Run_Bulk_Similarity_Job.sh	3	4	12
ER_Run_Data_Survival_Engine.sh	5	6	13
ER_Run_Full_Data_Output.sh	7	8	14

6.3.7 Using Wrapper Shell Script

You can execute the following jobs automatically using wrapper shell script (`Wrapper_Run_ER.sh`) for Entity Resolution in a specified sequence:

1. Create Index and Load the Data (`ER_Create_And_Load_Data_Into_Index.sh`)
2. Perform Matching (`ER_Run_Bulk_Similarity_Job.sh`)
3. Data Survival (`ER_Run_Data_Survival_Engine.sh`)
4. Load Data in FCC_ER_OUTPUT Table (`ER_Run_Full_Data_Output.sh`)

6.3.7.1 Steps

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin`.
2. Run the following command:

```
nohup ./Wrapper_Run_ER.sh "<pipelineid>" "<ERSchemaId>" "<Load_Type>"
"<FIC_MIS_DATE>" "<FSDF_Version>" "<Current_Batch>" "<Source Batch>"
"<Data Origin>" "<ERSchemaName>" "<Run_Type>" &
```

> **NOTE**
> - "Current_Batch refers to the table FCC_PROCESSING_GROUP in the Compliance Studio schema.
> - "<Source_Batch>" and "<Data_Origin>" are not relevant now as execution parameters and they are added for future use.

For example, you can use the following commands for CSA_8126 version:

```
nohup ./Wrapper_Run_ER.sh "CSA_8126" "ER_SCHEMA_PP_ALIAS" "FullLoad"
"20151210" "8126" "CSA_812" "CSA_812" "US" "ER_SCHEMA_PP" "RUN" &
```

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

3. Validate to ensure Global party IDs are generated for the Entities in the following Staging Output tables after executing the job:
   - STG_PARTY_MASTER
   - STG_PARTY_DETAILS
   - STG_PARTY_EMAIL_MAP
   - STG_PARTY_PHONE_MAP
   - STG_ADDRESS_MASTER
   - STG_PARTY_ADDRESS_MAP
   - STG_CUSTOMER_IDENTIFCTN_DOC

6.3.7.1.1 Cleanup Steps for Job termination

If job is terminated manually, see the following sections:

- For **Create Index and Load Data** job, see Cleanup Steps When the Create Index and Load Data Job Terminated Manually section.
- For **Bulk Similarity** job, see Cleanup Steps When the Bulk Similarity Job Terminated Manually section.
- For **Data Survival** job, see Cleanup Steps When the Data Survival Job Terminated Manually section.
- For **Load Data in the FCC_ER_OUTPUT** job, See Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually section.

For more information about parameters, see the Parameters for Entity Resolution Job execution section.

For example:

If the wrapper shell script is terminated manually during Bulk Similarity job execution, then you have to perform cleanup for the Bulk Similarity job. After completing the cleanup, execute the Bulk Similarity job and subsequent jobs manually.

6.4 Persisting the Data

Probable groups are created for entities that match. Merge rules are applied to all entities within a probable group to define which entities should be grouped into a global party. Day-on-day changes to the underlying party records may impact the global party group of which they are apart. The following sections show where the match or merge changes may impact a global party and when the global party would be deactivated and new global parties would be created. This can occur when matching criteria change or when groups and manually linked or de-linked.

> **NOTE** The change in a non-matching attribute will not change the global party group but may change attributes on the global party record if it impacts the data survival mechanism.

6.4.1 Persisting the Data When F_PERSIST_GUID and F_MANUAL_APPROVAL Flags are Set to False Condition

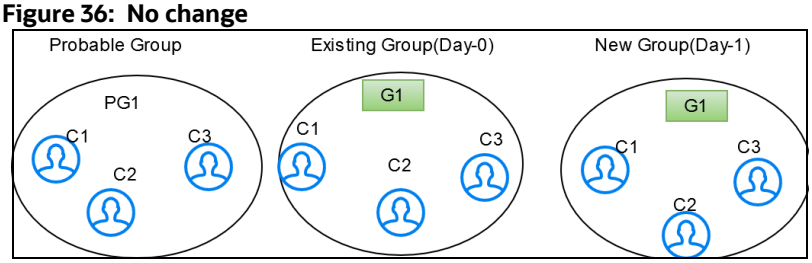
> **NOTE** This section is applicable only if F_PERSIST_GUID and F_MANUAL_APPROVAL flags are set to False in the FCC_ER_GUID_PERSIST_CONFIG table in the ER schema.

6.4.1.1 No change

Existing group elements are a subset of probable group elements, and the number of elements is the same in both groups. All elements in the existing Group have the same global id. The existing global id is assigned to probable group elements.

For example, G1 has C1, C2 and C3 entities. After the Day 1 batch execution, if there is no change in the existing group. Still, G1 has C1, C2 and C3 entities with the same global id.

Figure 36: No change

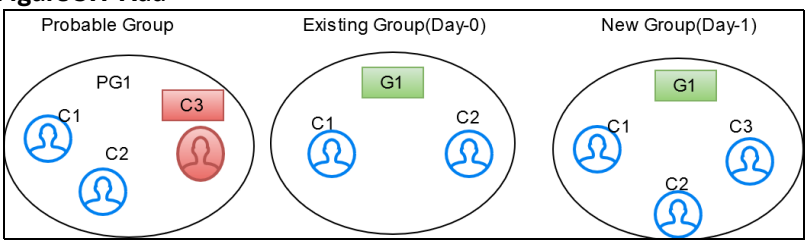


6.4.1.2 Add

Existing group elements are a subset of probable group elements, and the number of elements in the probable Group is more than the existing Group. Extra elements in the probable Group don't have any global id assigned yet. New elements are added to the existing Group, and the same global id is assigned.

For example, G1 has C1 and C2 entities. After the Day 1 batch execution, if C3 entity matches with existing group then C3 will be added to the existing group G1 with the same global id.

Figure 37: Add



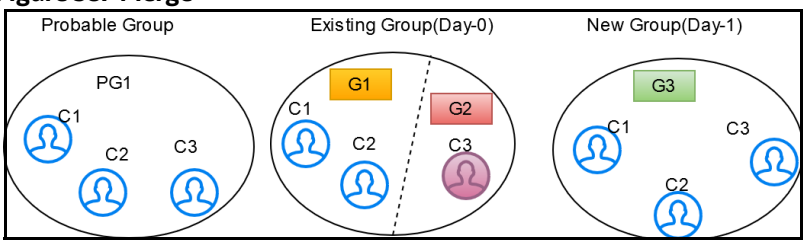
6.4.1.3 Merge

Existing group elements are a subset of probable group elements, and the number of elements is the same in both groups. Elements in the existing Group have different global ids assigned.

Elements are merged into a single group, and a new global id is assigned.

For example, G1 has C1 and C2 entities and G2 has a C3 entity. After the Day 1 batch execution, if C3 entity matches with an existing group then C3 will be merged into the existing group with a new global id assigned.

Figure 38: Merge

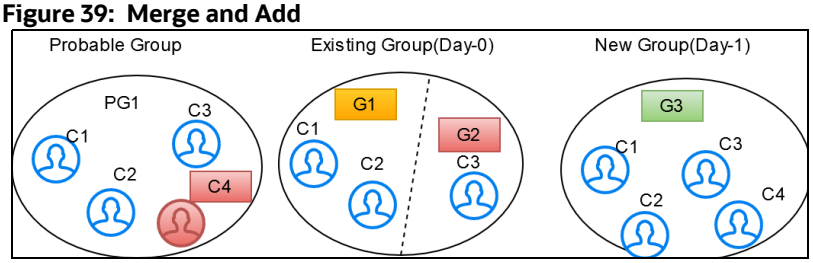


6.4.1.4 Merge and Add

Existing group elements are a subset of probable group elements, and the number of elements in the probable Group is more than the existing Group. Extra elements in the probable Group don't have any global id assigned yet, and standard elements have different global IDs assigned already. Common elements are merged into a single group, and new elements are added to the Group with a new global id.

For example, G1 has C1 and C2 entities, G2 has C3 entity. After the Day 1 batch execution, if C4 entity is added newly and C3 entity got changed then common entities are merged into a single group and a new entity is added to the group with a new global id (G3 has C1, C2, C3, and C4 entities) assigned.

Figure 39: Merge and Add



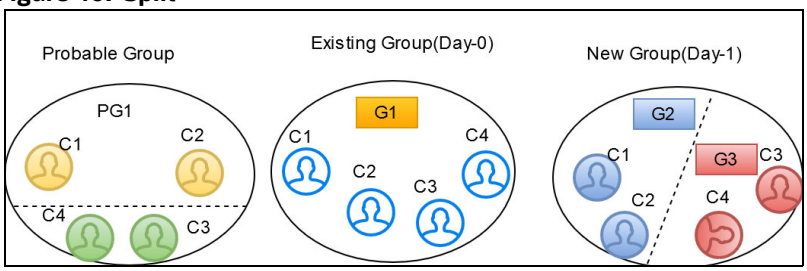
6.4.1.5 Split

After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements in both probable

and existing groups is the same. A single global id is assigned to all elements in the existing Group, and then probable group elements are split into different groups with new global ids assigned to each.

For example, G1 has C1, C2, C3 and C4 entities. After the Day 1 batch execution, if C3 and C4 entities are not matched with the existing entities of the group then C3 and C4 will be split into a new group. G2 has C1 and C2 entities and G3 has C3 and C4 entities with new global id is assigned to each group.

Figure 40: Split

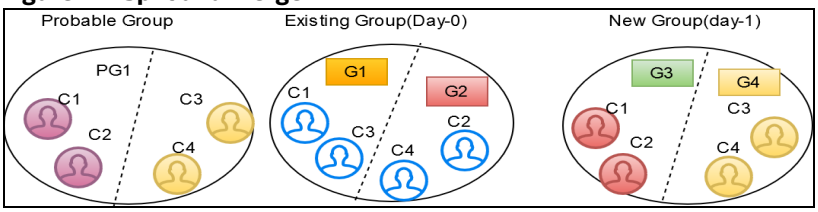


6.4.1.6 Split and Merge

After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements in both probable and existing groups is the same, and different global ids are assigned to elements in the existing Group, then probable group elements are split into different groups and merged, satisfying the same ruleset criteria with new global ids assigned to each.

For example, G1 has C1 and C3 entities and G2 has C2 and C4 entities. After the Day 1 batch execution, if C1 matches with C2 and C3 matches with C4 then C2 and C4 will be split separately and merged with C1 and C2 respectively. G3 has C1 and C2 entities and G4 has C2 and C4 entities with a new global id assigned to each group.

Figure 41: Split and Merge

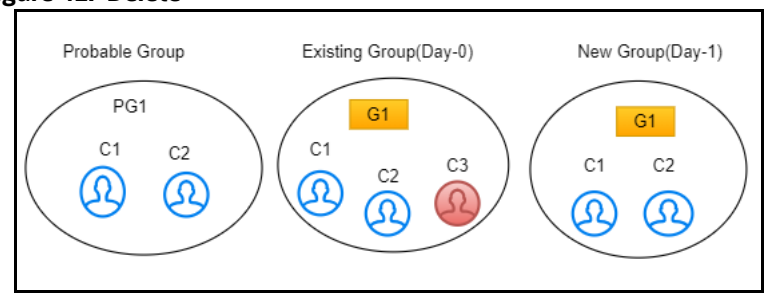


6.4.1.7 Delete

If an element exists in the existing Group, but the same element doesn't belong to any probable group and doesn't exist in the customer/entity dataset, it is deleted from the existing group with same global id assigned.If the deleted record is part of STG_DELETED_PARTIES_PRE table then underlying customers will also be deleted.

For example, G1 has C1, C2, and C3 entities. After the Day 1 batch execution, if C3 is deleted from the existing group then G1 has C1 and C2 entities with same global id.

Figure 42: Delete



6.4.2 Persisting the Data When F_PERSIST_GUID Flag is Set to True and F_MANUAL_APPROVAL Flag is Set to True/False Condition

NOTE

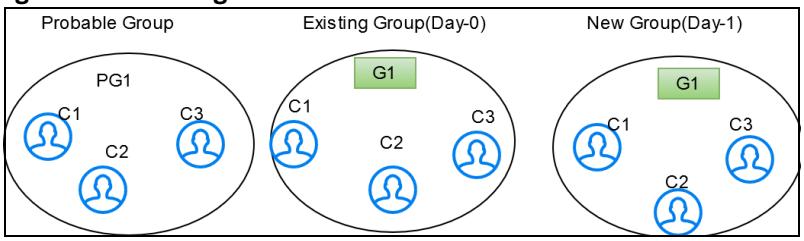
- This section is applicable only if F_PERSIST_GUID flag is set to True and F_MANUAL_APPROVAL flag is set to True/False in the FCC_ER_GUID_PERSIST_CONFIG table in the ER schema.
- Generally, Global Party ID will be persisted to the party that has **most number of entities** and if the number of entities are same between the parties, then the **least Global Party ID** will be persisted (it differs case to case).

6.4.2.1 No change

Existing group elements are a subset of probable group elements, and the number of elements is the same in both groups. All elements in the existing Group have the same global id. The existing global id is assigned to probable group elements.

For example, G1 has C1, C2 and C3 entities. After the Day 1 batch execution, if there is no change in the existing group. Still, G1 has C1, C2 and C3 entities with the same global id.

Figure 43: No change

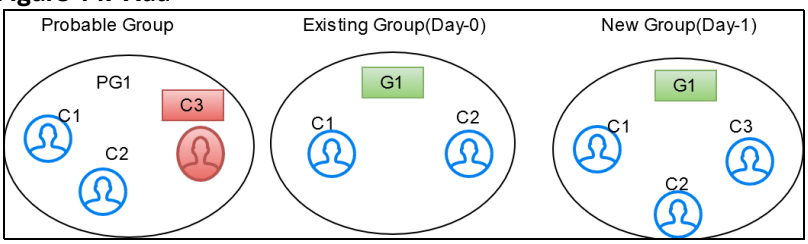


6.4.2.2 Add

Existing group elements are a subset of probable group elements, and the number of elements in the probable Group is more than the existing Group. Extra elements in the probable Group do not have any global id assigned yet. New elements are added to the existing Group, and the same global id is assigned.

For example, G1 has C1 and C2 entities. After the Day 1 batch execution, if C3 entity matches with existing group then C3 will be added to the existing group G1 with the same global id.

Figure 44: Add



6.4.2.3 Merge

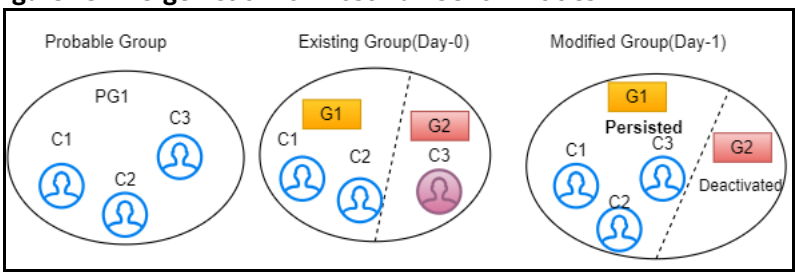
Existing group elements are a subset of probable group elements, and the number of elements is the same in both groups. Elements in the existing Group have different global ids assigned. Elements are merged into a single group, and the existing global id is persisted.

> **NOTE** You can manually persist the existing global id based on your requirement, only if F_PERSIST_GUID flag is set to True and F_MANUAL_APPROVAL flag is set to True/False in the FCC_ER_GUID_PERSIST_CONFIG table in the ER schema.
>
> For more information about manually persisting the existing global id, see **Persisting the Global Party ID through the Manual Action** section in the OFS Compliance Studio User Guide.

Case 1: If number of entities are different between the groups.

For example, G1 has C1 and C2 entities and G2 has a C3 entity. After the Day 1 batch execution, if C3 entity matches with an existing group then C3 will be merged into the existing group with same global id is persisted and G2 will be deactivated.

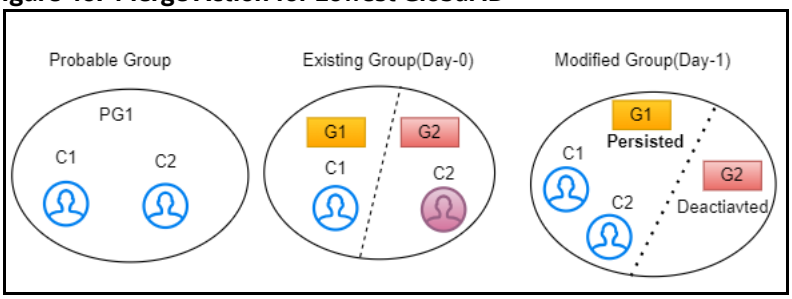
Figure 45: Merge Action for most Number of Entities



Case 2: If number of entities are same between the groups.

For example, G1 has C1 entity and G2 has a C2 entity. After the Day 1 batch execution, if C2 entity matches with an existing group then C2 will be merged into the existing group with same global id is persisted and G2 will be deactivated.

Figure 46: Merge Action for Lowest Global ID



6.4.2.4 Merge and Add

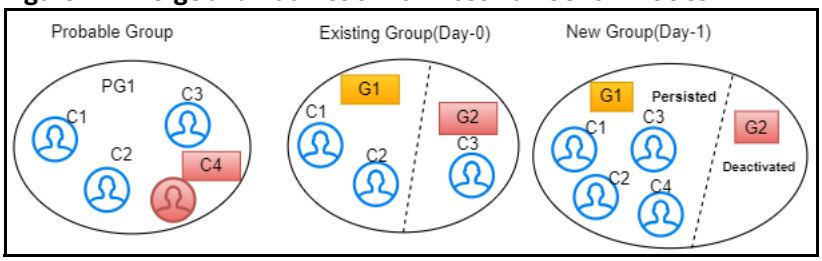
Existing group elements are a subset of probable group elements, and the number of elements in the probable Group is more than the existing Group. Extra elements in the probable Group do not have any global id assigned yet, and standard elements have different global IDs assigned already. Common elements are merged into a single group, and new elements are added to the Group with existing global id is persisted.

> **NOTE** You can manually persist the existing global id based on your requirement, only if F_PERSIST_GUID flag is set to True and F_MANUAL_APPROVAL flag is set to True/False in the FCC_ER_GUID_PERSIST_CONFIG table in the ER schema.
>
> For more information about manually persisting the existing global id, see **Persisting the Global Party ID through the Manual Action** section in the OFS Compliance Studio User Guide.

Case 1: If number of entities are different between the groups.

For example, G1 has C1 and C2 entities, G2 has C3 entity. After the Day 1 batch execution, if C4 entity is added newly and C3 entity got changed then common entities are merged into a single group and a new entity is added to the group with existing global id (G1 has C1, C2, C3, and C4 entities) is persisted and G2 will be deactivated.

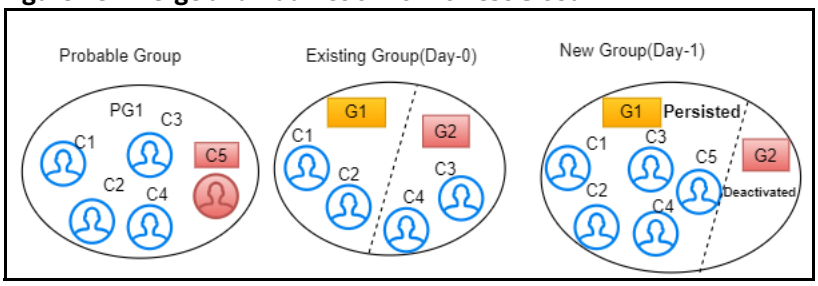
Figure 47: Merge and Add Action for most Number of Entities



Case 2: If number of entities are same between the groups.

For example, G1 has C1 and C2 entities, G2 has C3 and C4 entities. After the Day 1 batch execution, if C5 entity is added newly and C4 entity got changed then common entities are merged into a single group and a new entity is added to the group with existing global id (G1 has C1, C2, C3, C4 and C5 entities) is persisted and G2 will be deactivated.

Figure 48: Merge and Add Action for Lowest Global ID



6.4.2.5 Split

After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements in both probable and existing groups is the same. A single global id is assigned to all elements in the existing Group, and

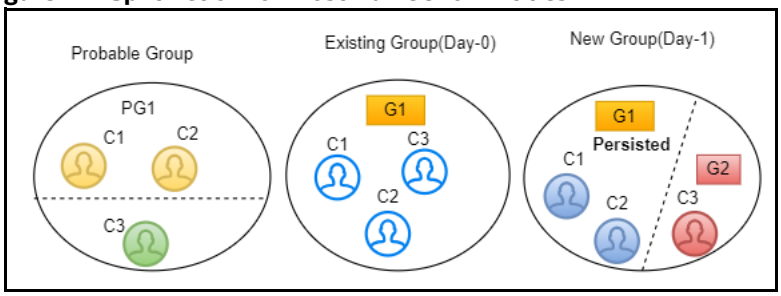
then probable group elements are split into different groups with existing global id is persisted for one group and new global id assigned to another group.

> **NOTE** You can manually persist the existing global id based on your requirement, only if F_PERSIST_GUID flag is set to True and F_MANUAL_APPROVAL flag is set to True/False in the FCC_ER_GUID_PERSIST_CONFIG table in the ER schema.
>
> For more information about manually persisting the existing global id, see **Persisting the Global Party ID through the Manual Action** section in the OFS Compliance Studio User Guide.

Case 1: If number of entities are different between the groups.

For example, G1 has C1, C2, and C3 entities. After the Day 1 batch execution, if C3 entity is not matched with the existing entities of the group then C3 will be split into a new group. G1 has C1 and C2 entities with existing global id is persisted and G2 has C3 entity with new global id assigned.

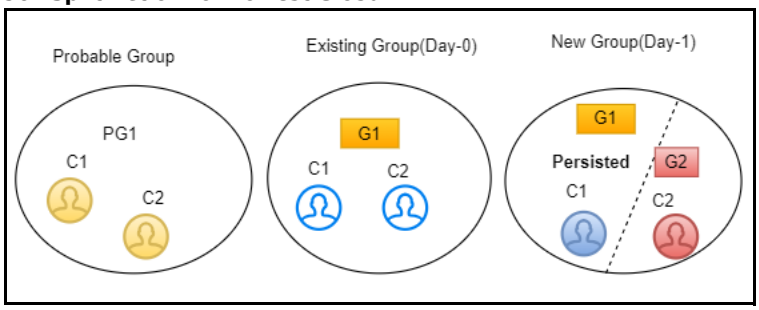
Figure 49: Split Action for most Number of Entities



Case 2: If number of entities are same between the groups.

For example, G1 has C1 and C2 entities. After the Day 1 batch execution, if C2 entity is not matched with the existing entities of the group then C2 will be split into a new group. G1 has C1 entity with existing global id is persisted and G2 has C2 entity with new global id assigned.

Figure 50: Split Action for Lowest Global ID



6.4.2.6 Split and Merge

After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements in both probable and existing groups is the same, and different global ids are assigned to elements in the existing Group,then probable group elements are split into different groups and merged, satisfying the same

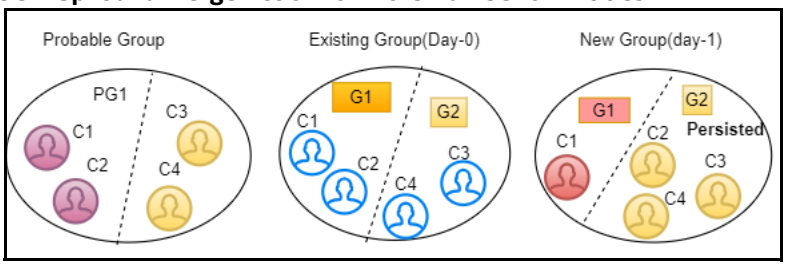
ruleset criteria with existing global id is persisted for one group and new global id assigned to another group.

> **NOTE** You can manually persist the existing global id based on your requirement, only if F_PERSIST_GUID flag is set to True and F_MANUAL_APPROVAL flag is set to True/False in the FCC_ER_GUID_PERSIST_CONFIG table in the ER schema.
>
> For more information about manually persisting the existing global id, see **Persisting the Global Party ID through the Manual Action** section in the OFS Compliance Studio User Guide.

Case 1: If number of entities are different between the groups.

For example, G1 has C1 and C2 entities and G2 has C3 and C4 entities. After the Day 1 batch execution, if C2 matches with C3 and C4 then C2 will be split separately and merged with C3 and C4 respectively. G1 has C1 with a new global id assigned and G2 has C2, C3 and C4 entities with existing global id is persisted.

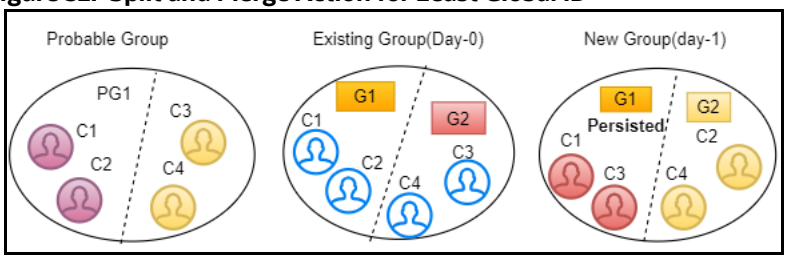
Figure 51: Split and Merge Action for more Number of Entities



Case 2: If number of entities are same between the groups.

For example, G1 has C1 and C2 entities and G2 has C3 and C4 entities. After the Day 1 batch execution, if C1 matches with C3 and C2 matches with C4 then C3 and C4 will be split separately and merged with C1 and C2 respectively. G1 has C1 and C3 entities with existing global id is persisted and G2 has C2 and C4 entities with a new global id assigned.

Figure 52: Split and Merge Action for Least Global ID

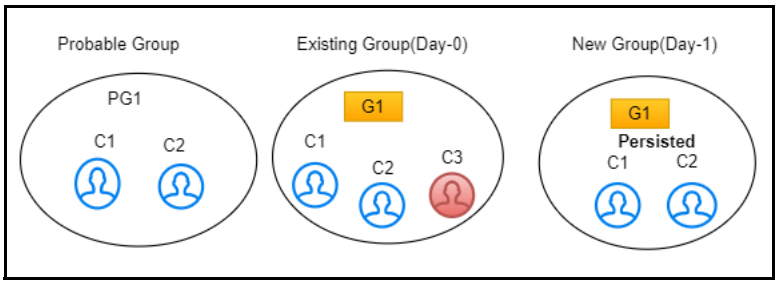


6.4.2.7 Delete

If an element exists in the existing group, but the same element does not belong to any probable group and does not exist in the customer/entity dataset, it is deleted from the existing group with same global id is assigned to the Group. If the deleted record is part of STG_DELETED_PARTIES_PRE table then underlying customers will also be deleted.

For example, G1 has C1, C2, and C3 entities. After the Day 1 batch execution, if C3 is deleted from the existing group then G1 has C1 and C2 entities with same global id is persisted.

Figure 53: Delete



6.5 Entity Resolution Metadata

Metadata tables manage the operation of the Entity Resolution jobs.

6.5.1 Default Data in the tables

The following are the complete set of tables that are used for the ER:

- **The following tables store the table structure definition for Party Master**:
  - **FCC_M_ER_TABLES**: This table contains information about different tables required by the product as part of an Entity Resolution process. The values in the column V_FSDF_VERSION differentiate FSDF versions to the tables belong to. This is used for creating Datasets and Data Surviving Rules.
  - **FCC_M_ER_TABLES_TL**: This table contains translative information for FCC_M_ER_TABLES, with multiple translations based on the Locale column.
  - **FCC_M_ER_COLUMNS**: This table contains information about columns a table has. It has mappings of columns and tables so that you can get the table's available columns information based on table Id. This is used for creating Datasets and Data Surviving Rules.
  - **FCC_M_ER_ATTRIBUTE**: This table contains information about columns. It has a column's information such as logical name and description. This is used for creating Datasets and Data Surviving Rules.
  - **FCC_M_ER_ATTRIBUTE_COLUMN_MAP**: This table contains mapping information of attributes and columns. It also stores information about the relationship between tables. This is used for creating Datasets and Data Surviving Rules.
  - **FCC_M_ER_ATTRIBUTE_TL**: This table contains translative information for table FCC_M_ER_ATTRIBUTE, which can have multiple translation information based on the Locale column.
- **The following tables store the Dataset definition**:
  - **FCC_M_ER_DATASET**: This table contains information about Datasets. It has a master (parent) table information like STG_PARTY_MASTER_PRE (when resolving FSDF data), output table, and pipeline Id, and tables where the data will flow when the data survival job is run.
  - **FCC_M_ER_DATASET_GROUP**: This table contains information about a Group of other tables that are part input dataset. It has an input group table like STG_PARTY_ADDRESS_PRE and also stores the join condition with the Master table, STG_PARTY_MASTER_PRE.
  - **FCC_M_ER_DATASET_MAP**: This table contains information about the mapping table, which provides the relationship between the Master and Group tables. For example,

STG_PARTY_ADDRESS_MAP_PRE stores the relationship between the STG_PARTY_MASTER_PRE and STG_PARTY_ADDRESS_PRE tables.

- **FCC_M_ER_DATASET_TL**: This table contains translative information for table **FCC_M_ER_DATASET**, which can have multiple translations based on the Locale column.

- **The following tables store the Preconfigured Match and Merge Ruleset**:
  - **FCC_MATCH_RULESET**: This table contains the information of the Rulesets created in Matching Rules UI. It gives information like the Pipeline ID, Ruleset Name, and Ruleset Description and contains ruleset details in JSON format.
  - **FCC_MERGE_RULESET**: This table contains the information of the Rulesets created in Merge Rules UI. It gives information like the Pipeline ID, Ruleset Name, and Ruleset Description and contains ruleset details in JSON format.
- **The following tables store the Dataset Survival Rule**:
  - **FCC_DATASURV_RULES**: This table contains the information on the Rules created in Data Survival Rules UI. It gives information like the Pipeline ID, Ruleset Name, and Ruleset Description and contains ruleset details in JSON format. This table contains information only for the Master table.
  - **FCC_DATASURV_GROUPS**: This table contains data survival rules, such as rule id, UI JSON, and query JSON. UI JSON is used on the UI side, and query JSON is used as input JSON for the Data survival engine. This table contains information only for child tables.
  - **FCC_DATASURV_TYPE**: This table contains information about different Data Survival Algorithms, such as Longest, Latest, Most Common, etc. There is a Type drop-down on Data Survival UI to choose values (fetched from this table) for a particular column.

  Data survival rules of out-of-the-box ER pipeline survive the “Latest” data based on FIC_MIS_DATE. Since data for ER is always considered as a complete snapshot for the extraction date (FIC_MIS_DATE), the FIC_MIS_DATE will be standard across the entire snapshot. Hence ER internally considers the additionally maintained D_LAST_UPDATED_DATE column in H$ tables to find out the latest data for survival. This is achieved by an additional set of metadata maintained in the following tables:
  - **FCC_M_ER_PROCESSING_COLUMNS**: This table stores the table name, column name, and ER pipeline id.
  - **FCC_DS_REF_COLUMN_MAPPING**: This table stores the table name, reference column name (the standard column of the table, i.e., FIC_MIS_DATE), target column name (the actual column on which “Latest” should be considered, i.e., D_LAST_UPDATED_DATE), and ER pipeline id.

    For Example, the sample records for both tables are as follows:

Figure 54: Sample Record for FCC_M_ER_PROCESSING_COLUMNS

	V_TABLE_NAME	V_COLUMN_NAME	V_PIPELINE_ID
1	STG PARTY ADDRESS MAP PRE	FIC MIS DATE	CSA 812
2	STG PARTY MASTER PRE	FIC MIS DATE	CSA 812
3	STG CUSTOMER IDENTIFCTN DOC PRE	FIC MIS DATE	CSA 812
4	STG PARTY EMAIL MAP PRE	FIC MIS DATE	CSA 812
5	STG PARTY PHONE MAP PRE	FIC MIS DATE	CSA 812
6	STG PARTY ADDRESS MAP PRE	FIC MIS DATE	CSA 812

Figure 55: Sample Record for FCC_DS_REF_COLUMN_MAPPING

	V_TABLE_NAME	V_REF_COLUMN_NAME	V_TARGET_COLUMN_NAME	V_PIPELINE_ID
1	STG PARTY MASTER PRE	FIC MIS DATE D	LAST UPDATED DATE	CSA 812
2	STG PARTY EMAIL MAP PRE	FIC MIS DATE D	LAST UPDATED DATE	CSA 812
3	STG CUSTOMER IDENTIFCTN DOC PRE	FIC MIS DATE D	LAST UPDATED DATE	CSA 812
4	STG PARTY PHONE MAP PRE	FIC MIS DATE D	LAST UPDATED DATE	CSA 812
5	STG PARTY ADDRESS MAP PRE	FIC MIS DATE D	LAST UPDATED DATE	CSA 812

NOTE These metadata tables should be seeded with appropriate values in any similar use cases.

- **The following table stores the flattening data query**:
  - **FCC_STUDIO_ER_QUERIES**: This table contains queries to fattening data from input tables for each pipeline id. The information in this table can be amended via an API if additional attributes need to be brought into matching.
- **The following tables to populate fields in Match and Merge Ruleset UI:**
  - **FCC_ER_INDEX**: This table contains the index name on the ruleset UI screen in Source Index Name and Target Index Name Field.
  - **FCC_IDX_M_JSON_MAP**: This table contains the mapping of each index populated on OpenSearch, making the initial candidate selection for records to be scored by the matching engine. This is required for Match and Merge Rulesets mapping screen. You need to add custom attributes for mapping manually. For more information on how to map, see the Steps section.
  - **FCC_ER_ATTRIBUTES**: This table contains attributes matched in ruleset UI in Source and target attribute for the respective index.

NOTE The Original ID is not masked but underlying all the attributes are hidden using the F_IS_MASKED column in the fcc_er_attributes table. This attribute is applicable only for Merge and Split Global Entities UI.

  - **FCC_IDX_M_LOOKUP**: This table contains the file name/index name of synonyms and Stopwords, which are used to improve the performance of Name/Address matching.
  - **FCC_IDX_M_LOOKUP_VALUES**: This table contains populated values for the above index names.
  - **FCC_ER_M_BKP_CONFIG**: This table contains the backup and failure recovery details.

6.5.2 Customize the Data in the Tables for ER types

Entity Resolution can be adapted for additional use cases by configuring the data in the metadata tables.

NOTE Out-of-the-box pipeline definitions should not be edited for customizations. If there are any customizations, create a copy of out-of-the-box pipeline definitions to apply any customizations otherwise the customizations will not persist when upgraded.

6.5.2.1 List of tables

- FCC_M_ER_DATASET
- FCC_M_ER_DATASET_GROUP
- FCC_M_ER_DATASET_MAP
- FCC_M_ER_DATASET_TL
- FCC_STUDIO_ER_QUERIES
- FCC_ER_INDEX
- FCC_IDX_M_JSON_MAP
- FCC_ER_ATTRIBUTES

6.5.2.2 Steps

Perform the following steps to customize the data using API:

1. Get the Datasets that exist in the system:
   a. Configure the hostname.
   b. Run the following command:

```
curl --location --request GET 'http://<HOSTNAME>:7051/datasurvival/
getDataSet' \
--header 'Content-Type: application/json'
For example,
curl --location --request GET 'http:// hostname.com:7051/datasurvival/
getDataSet' \
--header 'Content-Type: application/json'
```

> **NOTE** To modify the Dataset, you can provide the existing value for `datasetName` to edit the JSON file and modify the other parameters except for `datasetName` in the same file according to the requirement.

2. Enter the details of the Dataset in the Request JSON.
   a. Configure the hostname.
   b. Run the following command:

```
curl --location --request POST 'http://<HOSTNAME>:7051/datasurvival/
createdataset' \
--header 'Content-Type: application/json' \
--data-raw '{
    "fcc_m_er_dataset": {
        "tableId": "",
        "datasetName": "",
        "mapTable": "",
        "matchTable": "",
```

```
            "manualMatchTable": "",
            "manualMapTable": "",
            "viewDataset": "",
            "outputTable": "",
    "pipelineId":"",
            "statusFl": "",
            "productPartFl": "",
            "code": ""
        },
        "fcc_m_er_dataset_tl": {
            "tlTableId": "",
            "locale": "en-US",
            "tlDdatasetName": "Customer811"
        },
        "fcc_m_er_dataset_group": [
            {
                "groupTableId": "",
                "mapTableId": "",
                "groupMapTableJoin": "",
                "outputTable": "",
                "statusFl": "",
                "productPartFl": "",
                "code": "",
    "isParent":"Y"
            },
            {
                "groupTableId": "",
                "mapTableId": "",
                "groupMapTableJoin": "",
                "outputTable": "",
                "statusFl": "",
                "productPartFl": "",
                "code": "",
                "isParent":""
            },
            {
```

```
            "groupTableId": "",
            "mapTableId": "",
            "groupMapTableJoin": "",
            "outputTable": "",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":""
        },
        {
            "groupTableId": "",
            "mapTableId": "",
            "groupMapTableJoin": "",
            "outputTable": "",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":""
        },
        {
            "groupTableId": "",
            "mapTableId": "",
            "groupMapTableJoin": "",
            "outputTable": "",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":""
        }
    ],
    "fcc_m_er_dataset_map": [
        {
            "mapTableId": "",
            "datasetMapTableJoin": "",
            "outputTable": "",
            "statusFl": "Y",
```

```
            "productPartFl": "Y",
            "code": ""
        }
    ]
}'
```

For example,

```
curl --location --request POST 'http:// hostname.com:7051/
datasurvival/createdataset' \
--header 'Content-Type: application/json' \
--data-raw '{
    "fcc_m_er_dataset": {
        "tableId": "220",
        "datasetName": "Customer811",
        "mapTable": "FCC_ER_MAPPING_811",
        "matchTable": "FCC_ER_MATCHING_811",
        "manualMatchTable": "FCC_ER_MANUAL_MATCH_811",
        "manualMapTable": "FCC_ER_MANUAL_MAP_811",
        "viewDataset": "FCC_ER_VIEW_811",
        "outputTable": "STG_PARTY_MASTER",
"pipelineId":"CSA811",
        "statusFl": "",
        "productPartFl": "",
        "code": ""
    },
    "fcc_m_er_dataset_tl": {
        "tlTableId": "220",
        "locale": "en-US",
        "tlDdatasetName": "Customer811"
    },
    "fcc_m_er_dataset_group": [
        {
            "groupTableId": "221",
            "mapTableId": "",
            "groupMapTableJoin": "STG_PARTY_MASTER_PRE.V_PARTY_ID =
STG_PARTY_DETAILS_PRE.V_PARTY_ID",
            "outputTable": "STG_PARTY_DETAILS",
```

```
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":"Y"
        },
        {
            "groupTableId": "226",
            "mapTableId": "",
            "groupMapTableJoin": "STG_PARTY_MASTER_PRE.V_PARTY_ID =
STG_CUSTOMER_IDENTIFCTN_DOC_PRE.V_CUST_REF_CODE",
            "outputTable": "STG_CUSTOMER_IDENTIFCTN_DOC",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
            "isParent":""
        },
        {
            "groupTableId": "223",
            "mapTableId": "224",
           "groupMapTableJoin": "STG_ADDRESS_MASTER_PRE.V_ADDRESS_ID
= STG_PARTY_ADDRESS_MAP_PRE.V_ADDRESS_ID",
            "outputTable": "STG_ADDRESS_MASTER",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":""
        },
        {
            "groupTableId": "225",
            "mapTableId": "",
            "groupMapTableJoin": "STG_PARTY_DETAILS_PRE.V_PARTY_ID =
STG_PARTY_PHONE_MAP_PRE.V_PARTY_ID",
            "outputTable": "STG_PARTY_PHONE_MAP",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":""
```

```
        },
        {
            "groupTableId": "222",
            "mapTableId": "",
            "groupMapTableJoin": "STG_PARTY_DETAILS_PRE.V_PARTY_ID =
STG_PARTY_EMAIL_MAP_PRE.V_PARTY_ID",
            "outputTable": "STG_PARTY_EMAIL_MAP",
            "statusFl": "",
            "productPartFl": "",
            "code": "",
"isParent":""
        }
    ],
    "fcc_m_er_dataset_map": [
        {
            "mapTableId": "224",
          "datasetMapTableJoin": "STG_PARTY_DETAILS_PRE.V_PARTY_ID =
STG_PARTY_ADDRESS_MAP_PRE.V_PARTY_ID",
            "outputTable": "STG_PARTY_ADDRESS_MAP",
            "statusFl": "Y",
            "productPartFl": "Y",
            "code": ""
        }
    ]
}'
```

3. Delete the existing Dataset:
   a. Configure the hostname.
   b. Run the following command:

```
curl --location --request POST 'http://<HOSTNAME>:7051/datasurvival/
deleteDataSet' \
--header 'Content-Type: application/json' \
--data-raw '{
"dataSetId":""
"datasetName":""
}'
For example,
```

```
curl --location --request POST 'http:// hostname.com:7051/
datasurvival/deleteDataSet' \
--header 'Content-Type: application/json' \
--data-raw '{
"dataSetId":"273"
"datasetName":"Customer811"
}'
```

4. Get Dataset Hierarchy for table relation summary:
   a. Configure the hostname.
   b. Run the following command:

```
curl --location --request POST 'http://<HOSTNAME>:7051/datasurvival/
getDataSetHierarchySummary' \
--header 'Content-Type: application/json' \
--data-raw '{
    "dataSetId": "",
"datasetName": ""
}'
For example,
curl --location --request POST 'http:// hostname.com:7051/
datasurvival/getDataSetHierarchySummary' \
--header 'Content-Type: application/json' \
--data-raw '{
    "dataSetId": "273",
"datasetName": "Customer811"
}'
```

5. Get Dataset Hierarchy Tables' Data:
   a. Configure the hostname.
   b. Run the following command:

```
curl --location --request POST 'http://<HOSTNAME>:7051/datasurvival/
getDataSetHierarchy' \
--header 'Content-Type: application/json' \
--data-raw '{
    "dataSetId": "",
"datasetName": ""
}'
For example,
curl --location --request POST 'http:// hostname.com:7051/
datasurvival/getDataSetHierarchy' \
```

```
--header 'Content-Type: application/json' \
--data-raw '{
    "dataSetId": "273",
"datasetName": "Customer811"
}'
```

6. To change any field name in the OpenSearch Index for the ER type:
   a. Modify the value in the QUERY column in the **FCC_STUDIO_ER_QUERIES** to bring the field name in the ES Index.
   b. Add the QUERY column values to the **V_IDX_JSON** column in the **FCC_STUDIO_ER_QUERIES**

      NOTE: Ensure the value is the same in both columns, QUERY, and V_IDX_JSON.
7. To populate the Source and target index on Ruleset UI:
   a. Add a new record in the table, FCC_ER_INDEX.
   b. Add Source and target attributes on respective indexes in the table FCC_ER_ATTRIBUTES.
   c. Create a new Ruleset for the customized ER type(s) in tables in the previous step. See the Creating Rulesets section in the OFS Compliance Studio User Guide for creating and configuring rulesets.
   d. Execute the ER jobs with customized ER type(s). For more information on how to execute the jobs, see the Executing the ER Jobs section.

6.5.3 Populate the Metadata for Data Survival in Compliance Studio Schema

The **FCC_M_ER_ATTRIBUTE_PREC** table in Compliance Studio Schema stores information about the attribute column name, code of the attribute value, and the precedence value.

Table 20 structure with examples:

Table 20: Metadata

v_metadata_type	v_column_cd	n_precedence
Occupation	teacher	2
Geo-location	US	3

6.5.3.1 REST API to Load Metadata into Compliance Studio Schema

This is used to upload metadata and precedence and update the precedence for existing metadata types in the `FCC_M_ER_ATTRIBUTE_PREC` table.

URL: `http://<hostname>:7051/datasurvival/loadDataSurvMetadata`

Request Method: POST

Request Headers: Content-Type: application/json

Request body:

```
[{
        "vmetadataType": "Geo Risk",
```

```
        "vcolumnCd": "UK",
        "nprecedence": "6"
    },
    {
        "vmetadataType": "Geo Risk",
        "vcolumnCd": "US",
        "nprecedence": "5"
    },
  {
        "vmetadataType": "Geo Risk",
        "vcolumnCd": "FIN",
        "nprecedence": "3"
    }
]
```

6.5.3.2 REST API to Update Metadata Type

This is used to delete the existing set of metadata and update the metadata type and precedence with a new set of metadata.

URL: `http://<hostname>:7051/datasurvival/updateMetadataType`

Request Method: POST

Request Headers: Content-Type: application/json

Request body:

```
[{
        "vmetadataType": "Geo Risk",
        "vcolumnCd": "UK",
        "nprecedence": "6"
    },
    {
        "vmetadataType": "Geo Risk",
        "vcolumnCd": "US",
        "nprecedence": "5"
    }
]
```

6.5.3.3 REST API to Get Metadata Type and Precedence

This is used to get the records available in the precedence table.

URL: `http://<hostname>:7051/datasurvival/getAttributePrecMetadata`

Request Method: GET

Request Headers: Content-Type: application/json

6.5.3.4 REST API to Delete any Metadata Type

This is used to delete all records for a specific metadata type in the precedence table.

URL: `http://<hostname>:7051/datasurvival/deleteMetadataType?vMetadataType=<Metadata Type>`

For example,

`http://testserver.oracle.com:7051/datasurvival/deleteMetadataType?vMetadataType=Occupation`

Request Method: POST

Request Headers: Content-Type: application/json

6.6 Removal of Entities from the Global Party (Deleted Party)

For large volume processing in Entity Resolution, delta processing is recommended for performance reasons. When delta processing is used the system needs to be aware of when there are parties to be deleted as well as added or changed.

The delete actions refers to the parties being removed from the system and from global parties, and they are to be skipped from further processing selectively.

STG_DELETED_PARTIES_PRE: This table contains parties id to be deleted.

6.6.1 Impact on Manual Decisioning on Deleting Parties

Delta Load: If you delete any customers with manual matches (if manual matches are present in the pending approval/reject), then the particular manual match will be moved to the rejected tab in the Compliance Studio UI.

Full Load: If the customer is deleted, then the manual match containing customers will be moved to the FCC_ER_MATCHING_DELETED table.

Manual Decisioning: The matches in FCC_ER_MATCHING and FCC_ER_MANUAL_MATCH tables are invalid and moved to the FCC_ER_ MATCHING _DELETED table when the party id is deleted. As matches are moved to DELETED, the pending requests (for approval or rejections) will be removed from the UI list, and those matches will no longer be reflected in the Manual Decisioning UI. You can view different statuses in the STATUS_CD column in the FCC_ER_MANUAL_MATCH table.

STATUS_CD: It stores the state of the records upon which manual actions are taken from the Manual Decisioning UI. The possible statuses are:

- SR - System Rejected (The batch rejected manual matches should be marked with a separate reject code)
- PR - Pending Rejected
- A - Request Approved
- R - Request Rejected
- IRR - Pending Request for Rejection
- IRA - Pending Request for Approval

6.7 Ability to Remove Split and Merge Manually

In the creation of global parties any manual split or merges take precedence over system changes even when data changes. If data is changed in upstream systems, you may wish to remove any manual decisions from having precedence and revert to the automatic behavior.

The override flag can be enabled only when manual action is taken on the particular global party id.

The F_OVERRIDE_FLAG in the FCC_ER_MAPPING table controls whether to override the manual decision or not, irrespective of the V_MD_FLAG value. The value of F_OVERRIDE_FLAG can be selected using the **Action** drop-down from the UI. For more information, see the **Using Merge and Split Global Entities** section in the OFS Compliance Studio User Guide.

6.8 Expiry of Entity Address Mapping

If an address mapped to the parties is to be removed from the system, then set the D_ADDRESS_END_DATE attribute as a date lesser than fic_mis_date/previous date in the STG_PARTY_ADDRESS_MAP_PRE table. This will remove the address mapping as part of the Entity Resolution batch run from the STG_PARTY_ADDRESS_MAP table but the mapped address will be available in the STG_ADDRESS_MASTER table.

The expired address mapping records will still be loaded into history tables (H$STG_PARTY_ADDRESS_MAP_PRE and H$STG_ADDRESS_MASTER_PRE), and it will not be present in the flattened input table (FCC_ER_FULL).

7 ML for AML (ML4AML)

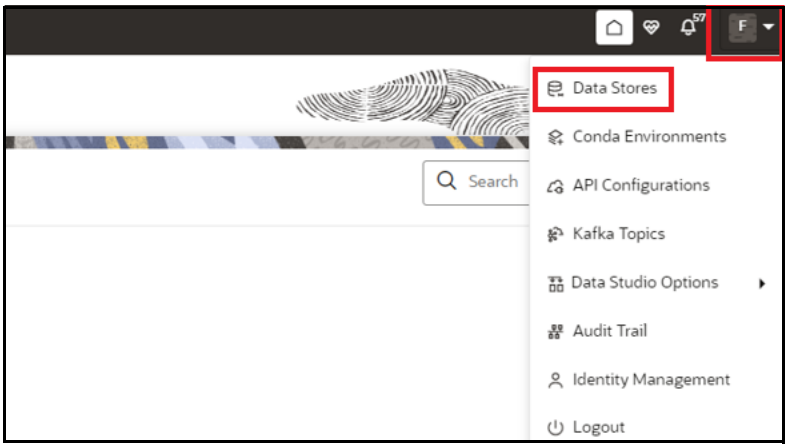
Topics:

7.1 Creating Data Store

To create a data store, follow these steps:

1. Navigate to **Workspace Summary** page.

Figure 56: Workspace Summary



2. Click on the **User Profile** drop-down list and select **Data Store**.

3. Click on **Add Data Store** button to create the data store for the sandbox workspace. The Add Data Store page is displayed.

Figure 57: Add Data Store with Oracle Database

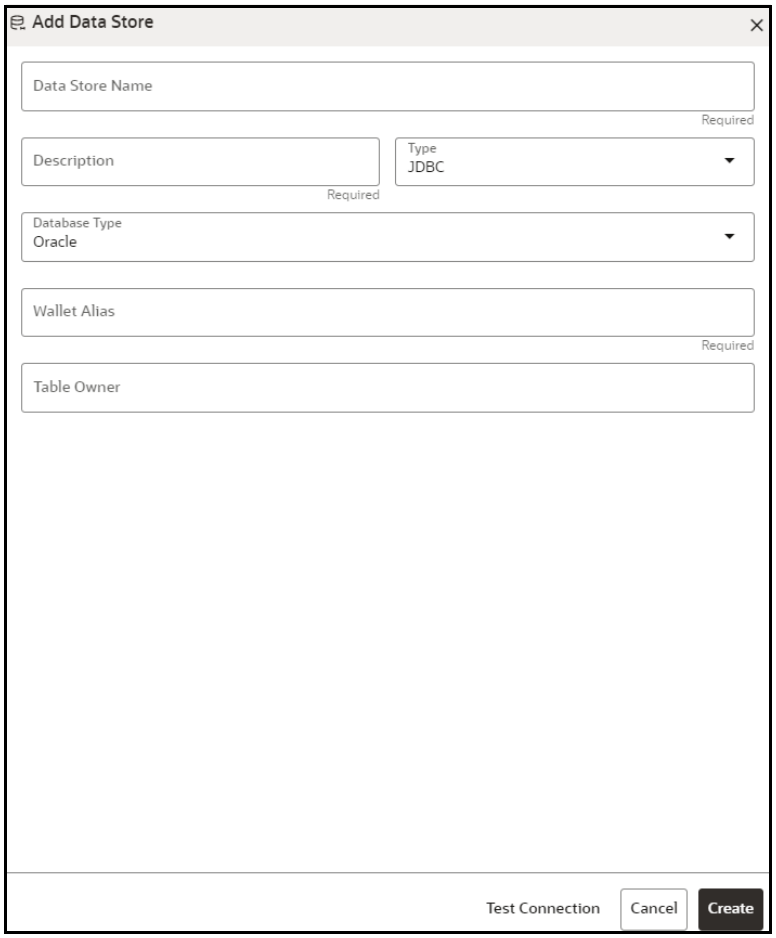


Figure 58: Add Data Store with Hive Database

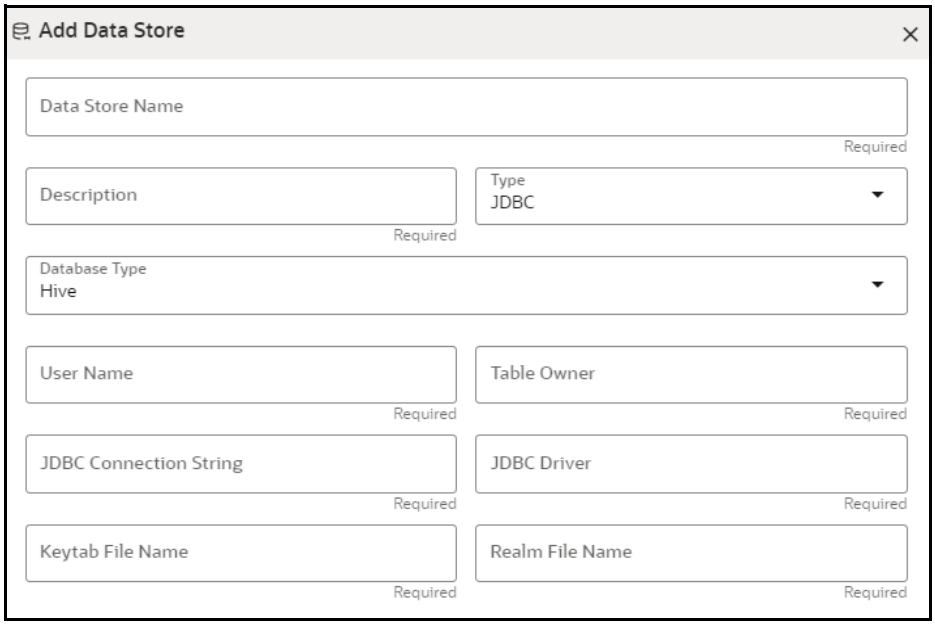
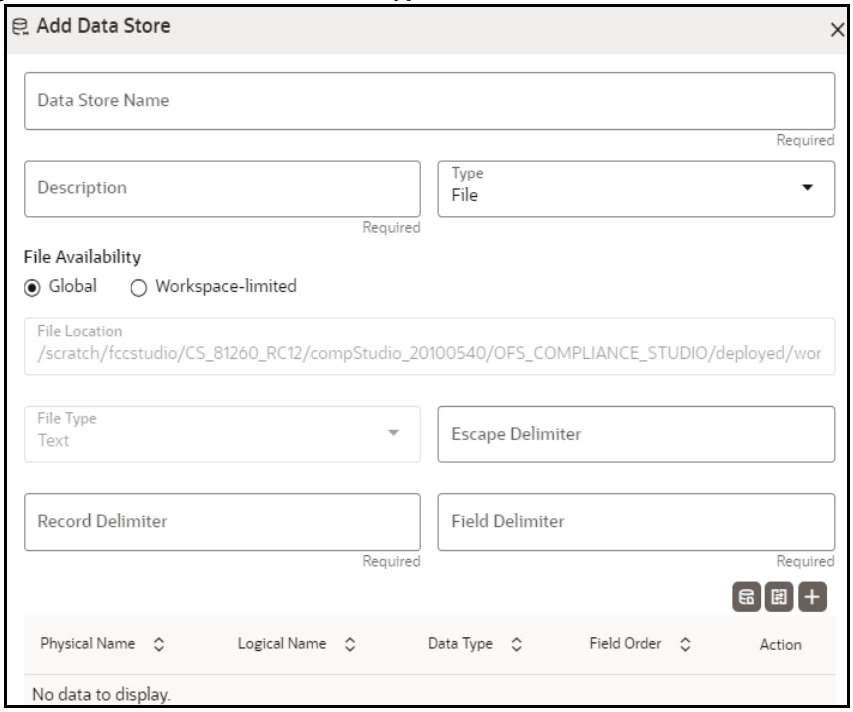


Figure 59: Add Data Store with File Type



4. Enter the required details as describe in the Table 21.

Table 21: Add Data Store

Field	Description
Database Source Name	Enter the connection URL to the database for the data schema.
Description	Enter the description of database connection.

Table 21: Add Data Store

Field	Description

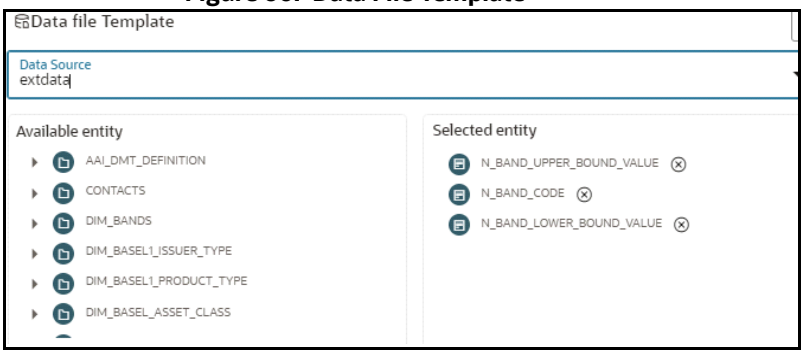
Type	Select the Type from the drop-down list. The available options are **JDBC** and **File**.
	• **JDBC** : If selected, the Database type options Oracle and Hive are displayed.
	• **File:** If selected, (see Figure 59), the following options are displayed.
	▪ **Global**: Select this option if you want to fetch the global level datasource details. You need to place the file with the datasource details in JSON format in the following location:
	`<Compliance_Studio_Installation_path>/deployed/workspace/mmg/Data Sources/datasources/##Data Source Name##/datafiles/##MISDATE##/*`
	▪ **Workspace-limited**: Select this option if you want to fetch the workspace level datasource details. You need to place the file with the datasource details in JSON format in the following location:
	`<Compliance_Studio_Installation_path>/deployed/workspace/mmg/##Workspace##/datasources/##Data Source Name##/datafiles/##MISDATE##/*`
	▪ **Record Delimiter**: There is a separation of the records using a delimiter character like a comma, semicolon, hyphen, and so on for the rows. Enter the delimiter in the Record Delimiter field. This is a mandatory field and limited to two characters.
	▪ **Field Delimiter**: There is a separation of the records using a delimiter character for the columns.Enter the delimiter in the Field Delimiter field. This is a mandatory field.
	You can either add the file details using data template or manually.
	Click **Data File Template** icon to select the Data Source entities and click **Save**.
	Figure 60: Data File Template
	 
	OR
	Click **Add** icon to add the details such as Physical Name, Logical Name, Data Type, and Field Order manually and click **Save**.

Table 21: Add Data Store

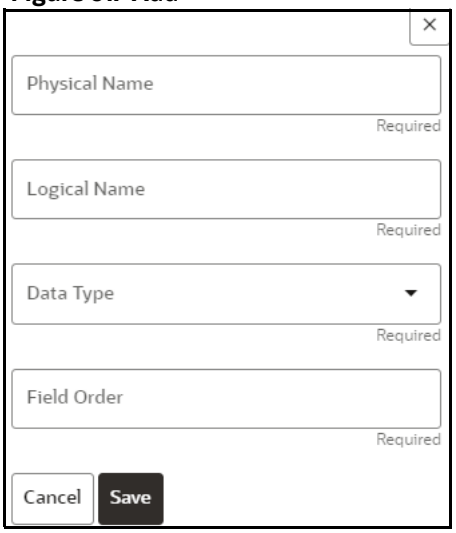
Field	Description
Type	**Figure 61: Add**  Click **Re-order Grid** option to reorder the field orders.
Database Type	If the **Type** field is selected as **JDBC**, the following options are displayed. Select the Database Type as **Oracle** or **Hive**. **NOTE:** Selected tables during Hive sourcing should be preexisting in the RDBMS data schema before the workspace population. If you select **Database Type** as **Oracle** (see Figure 57), then following additional fields are displayed to enter details: • **Wallet Alias** : Enter the Wallet Alias. This value should be same as configured using Oracle Wallet. • **Table Owner**: Enter the Oracle Database schema name. This is an optional field. If you select Database Type as **Hive** (see Figure 58), then following additional fields are displayed to enter details: • **User Name**: User Name / Principal is used for Kerberos authentication. **Example**: mmg/hostname@ORACLE.COM. • **Table Owner**: Enter the Hive schema. • **JDBC Connection String**: Enter the JDBC Connection String. **Example:** jdbc:hive2://hostname:10000/default;principal=hive/hive-service-hostname@ORACLE.COM. • **JDBC Driver**: Supports org.apache.hive.jdbc.HiveDriver and com.cloudera.hive.jdbc4.HS2Driver. • **Keytab File Name**: Enter the Name of the keytab file present in conf directory. • **Realm File Name**: Enter the Name of the configuration file present in conf directory. **Example**: krb5.conf

Table 21: Add Data Store

Field	Description
Database Type	**NOTE:** • Schema population for Hive as target is not supported. • This is applicable only for Sandbox Workspace.

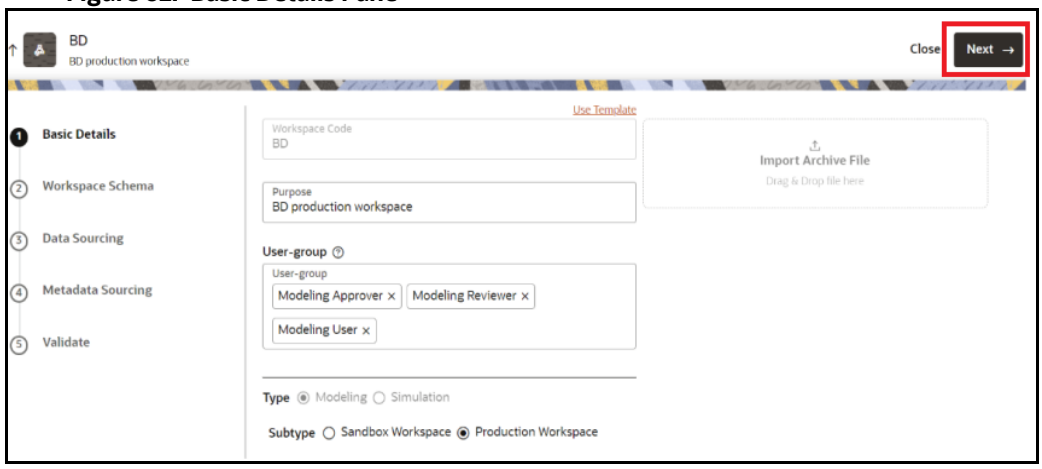
5. Click **Create** to create/add a new data store.

7.2 Updating Conda Environments in the BD Production Workspace

To update conda environments in the BD production workspace, follow these steps:

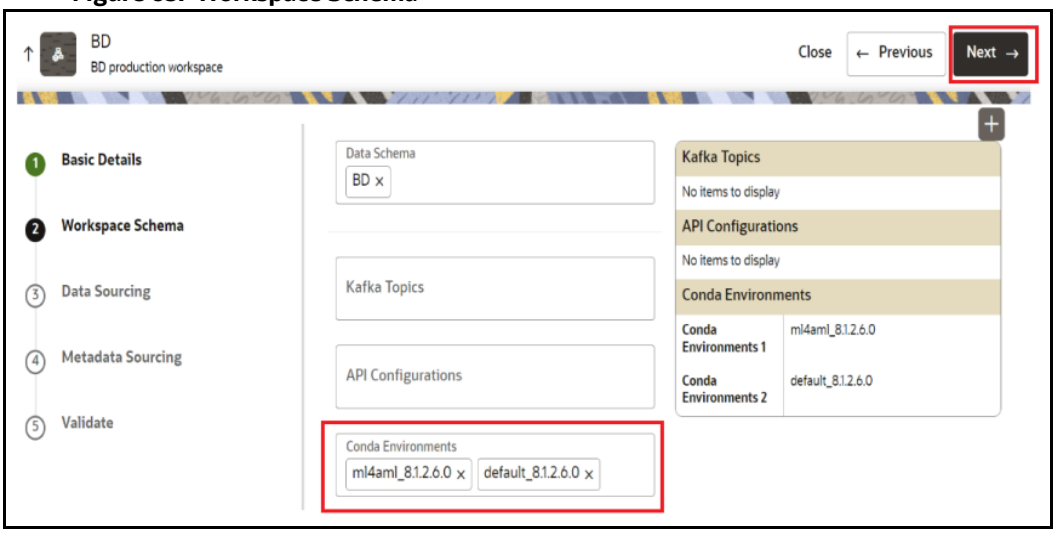
1. Navigate to **BD** workspace.
2. Click **Action** ••• icon and select **Edit**. The **Basic Details** pane is displayed.

Figure 62: Basic Details Pane



3. Click **Next** to navigate to the **Workspace Schema** pane.

Figure 63: Workspace Schema



4. From the **Conda Environments**, select **default_8.1.2.6.0** and **ml4aml_8.1.2.6.0**.
5. Click **Next** to navigate to the **Data Sourcing** pane.
6. Click **Next** to navigate to the **Metadata Sourcing** pane.
7. Click **Update**.

   The conda environments are updated in the BD production workspace.

7.3 Creating a Sandbox Workspace

Prerequisites

Before creating the sandbox workspace, the user should follow these steps:

1. Create the Tablespace
2. Create the Sandbox Schema
3. Assign Grants to the Sandbox Schema

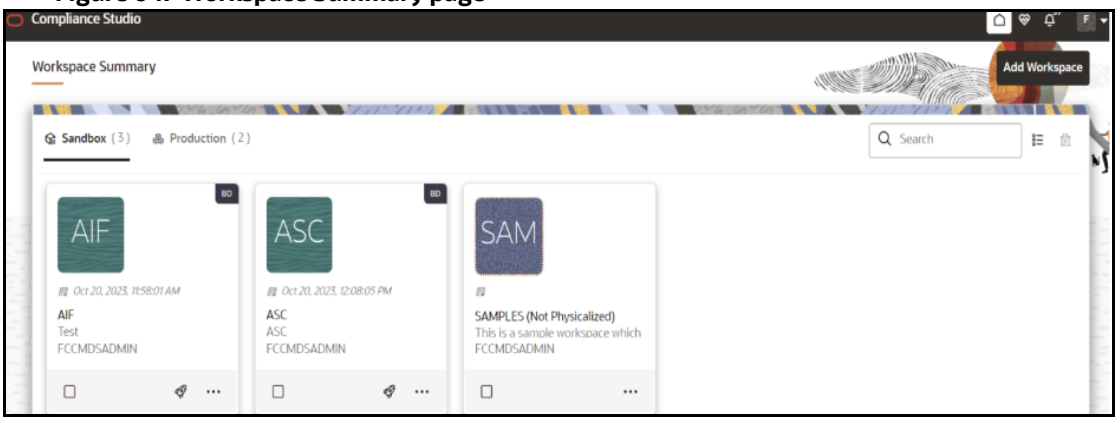
For more information on creating tablespace, sandbox schema and assigning grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

Topics:

- Basic Details
- Workspace Schema
- Data Sourcing
- Metadata Sourcing
- Validate Workspace
- Summary

After clicking **Add Workspace** in the **Workspace Summary page (CS Home Page)**, the Workspace Creation window is displayed.

Figure 64: Workspace Summary page



7.3.1 Basic Details

1. Provide the requested details for **Workspace Code** and **Purpose**.
2. Select the **User-group** from the drop-down list.
3. Select the subtype as **Sandbox Workspace**.
4. Enable the **Production Workspace** button.
5. Choose **BD** as workspace from the drop-down list (Production workspace).

   Or

   Click **Import Archive File** to import for basic details. If you use this feature, the other fields described in the preceding rows are auto populated.

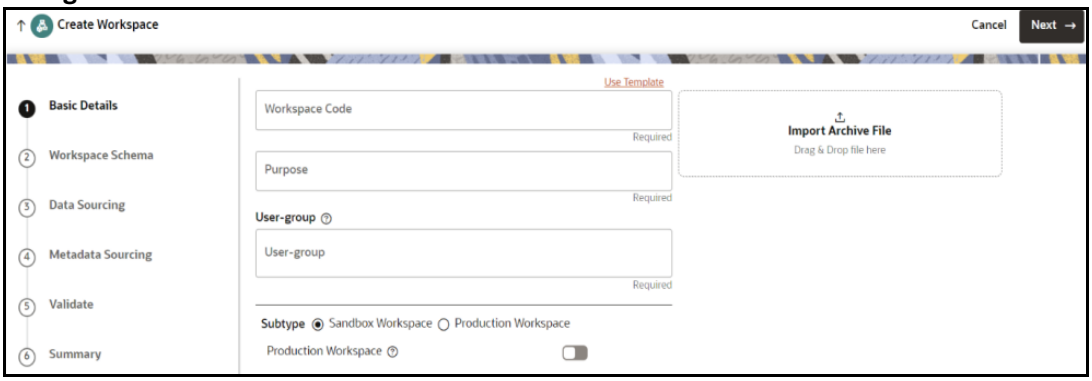
   Click on the box to open the file selector dialog and select the required configuration file or drag the file from its directory and drop it in the box.

   Or

   Click **Use Template** hyperlink to select workspace template and select workspace from the **Library** drop-down list. The **Update schema mapping** window is displayed.

   Select the **New Data Schema** and **New Data Store Name** drop-down list respectively, if required or use the existing data and click **Update**.
6. Click **Next**.

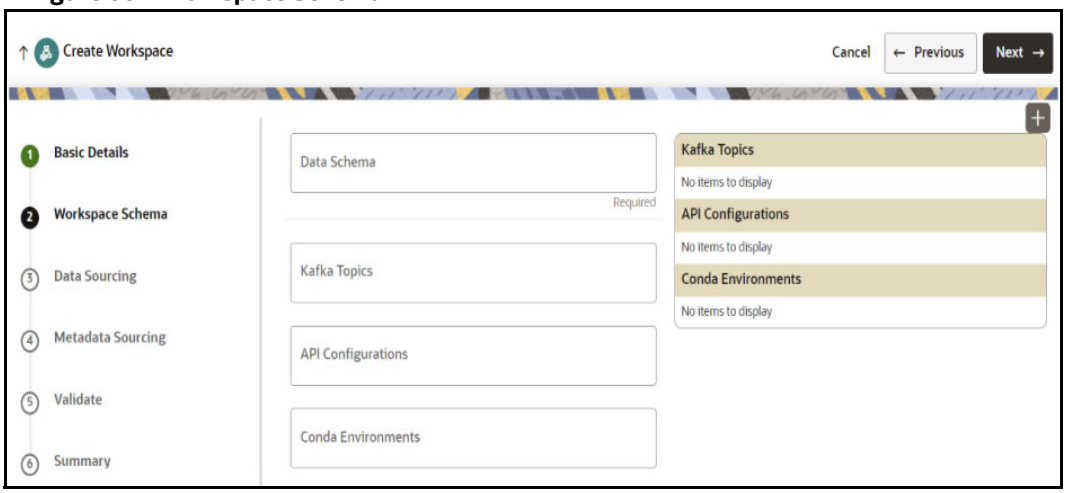
Figure 65: Basic Details



7.3.2 Workspace Schema

1. Select the **newly created data store** (see Basic Details section) as **Data Schema**.
2. Select the following **Conda Environments**:
   a. **default_8.1.2.6.0**
   b. **ml4aml_8.1.2.6.0**
3. Click **Next**.

Figure 66: Workspace Schema



7.3.2.1 For Automatic Scenario Calibration (ASC) Use Case

- The target schema used for the ASC workspace should be a valid BD atomic schema like BD **pre-prod**, BD **UAT**, BD **Dev**, etc., because we use BD packages and functionality to reproduce alerts as in BD.
- Historical data can come from a variety of Stores like Hive/another Oracle Schema, etc., (generally from an archived data store).
- ATL or Production alerts can come from actual BD production.
- ASC use case might need as many **data stores** to pull in the data required for the analysis (ATL/BTL).
- To create a data Store, see the Creating Data Store section.

> **NOTE** ASC runs scenarios to produce test alerts. Hence, the BD production schema should not be used as an ASC BD target.

7.3.2.2 For ML and Typology Use Case

The following use cases falls under this category:

- Customer Risk Scoring
- Customer Segmentation
- AML Event Scoring
- Typology Scenario for Shell Detection
- Scenario Model

Use any empty schema pointed by newly created data Stores (see Basic Details section) as **Meta** and **Data Schema**.

> **NOTE** Common workspace cannot serve for both **ML & Typology** and **ASC** use cases. So, you must create separate workspace for **ML & Typology** and **ASC** use cases.

7.3.3 Data Sourcing

Generally, BD Production does not hold enough history; hence data sourcing from other Stores will be required.

- Select the group of tables from an archived data store like **Hive Data Source/Other Oracle Data Sources/BD Production Data Source**.

  > **NOTE** The following tables are applicable for all Use Cases.

  - CUST
  - CUST_ACCT
  - CUST_SMRY_DAILY
  - CUST_SMRY_MNTH
  - ACCT
  - ACCT_BAL_POSN_SMRY
  - ACCT_SMRY_MNTH
  - ACCT_POSN
  - CASH_TRXN
  - WIRE_TRXN
  - MI_TRXN
  - BACK_OFFICE_TRXN
  - TRADE
  - TRADE_EXECUTION_EVENT
  - SCRTY_MKT_DAILY
  - SCRTY
  - ORDR
  - EXECUTION
  - NTCPTRY_PRFL
  - DERIVED_ADDRESS
  - WATCH_LIST
  - WIRE_TRXN_INSTN_LEG
  - STDO_ERROR_DETAILS
  - FCC_AM_EVENT_ENTITY_MAP

- FCC_AM_EVENTS
- FCC_AM_EVENT_BINDING
- FCC_AM_EVENT_DETAILS
- KDD_SCNRO
- CUST_ACCT_ROLE
- FCC_AM_PROCESS_LOG

NOTE
- This step is **optional** for **ASC** as the ASC **workspace** target is assumed to be a valid BD schema parallel to production. This step should be considered when the ASC BD schema does not have sufficient data. In that case, use this option to source the data.
- The following additional tables are applicable for **ASC** use case only.

- KDD_PARAM_SET
- KDD_PRCSNG_BATCH_HIST
- KDD_JOB
- KDD_RUN
- KDD_SCNRO
- KDD_PTTRN
- KDD_REVIEW
- KDD_REVIEW_SCNRO
- KDD_BREAK
- KDD_BREAK_BINDING

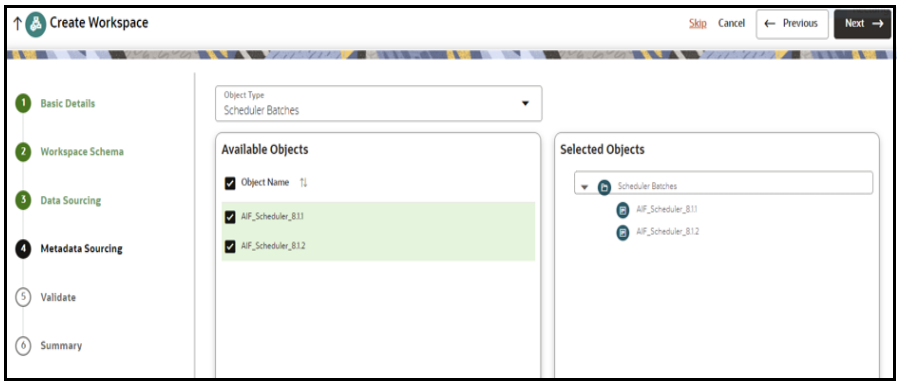
7.3.4 Metadata Sourcing

NOTE This section is not applicable for **ASC** use case.

1. Select **Scheduler Batches** from the **Object Type** drop-down list.
2. In the **Available Objects**, users need to select the following schedulers in sequential order based on the respective use case:
   - For Customer Risk Scoring and Customer Segmentation and Anomaly Detection use cases:
     - **AIF_Scheduler_8.1.1**
     - **AIF_Scheduler_8.1.2**
     - **AIF_Scheduler_8.1.2.1**
   - For AML Event Scoring use case:
     - **AMLES_Scheduler_8.1.1**
     - **AMLES_Scheduler_8.1.2.1**

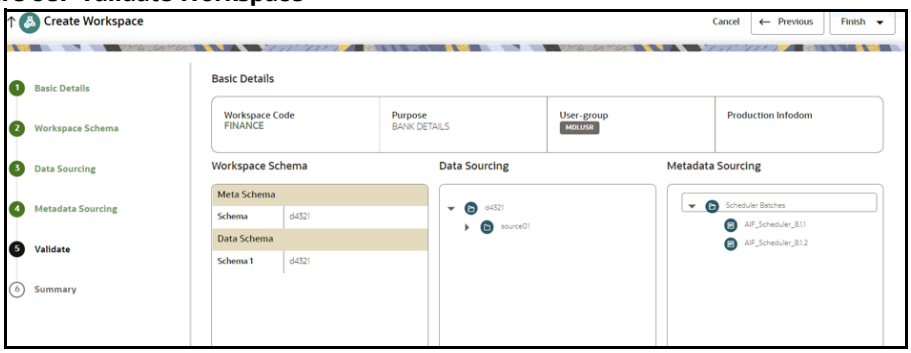
- For Typology Scenario use case:
  - **AML_Scenario_Scheduler_8.1.2.1**
- For Customer Screening use case:
  - **Sancations_Scheduler_8.1.2.4**
- For Scenario Model use case:
  - **SM_Scheduler_8.1.2.6.1**

Figure 67: Metadata Sourcing



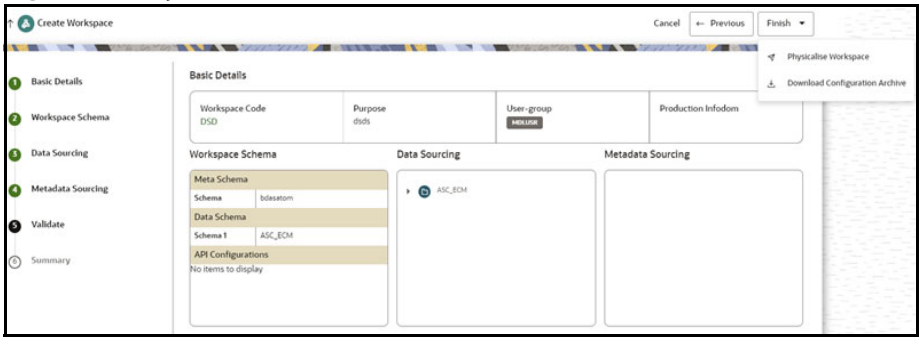
7.3.5 Validate Workspace

Figure 68: Validate Workspace



1. Click **Finish** and then select **Physicalize Workspace**.

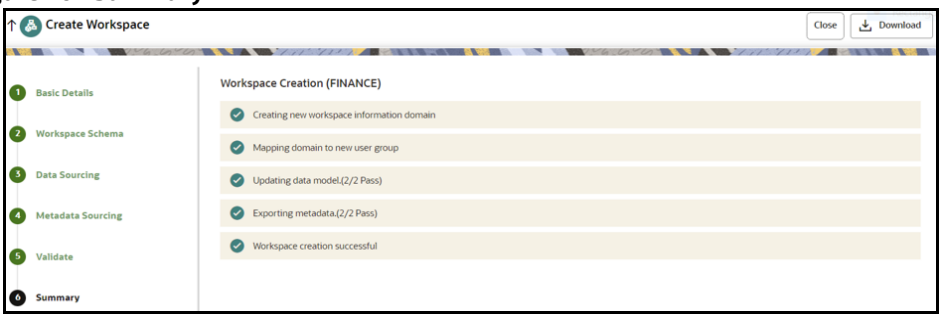
Figure 69: Physicalize Workspace



7.3.6 Summary

You can view summary of the created workspace.

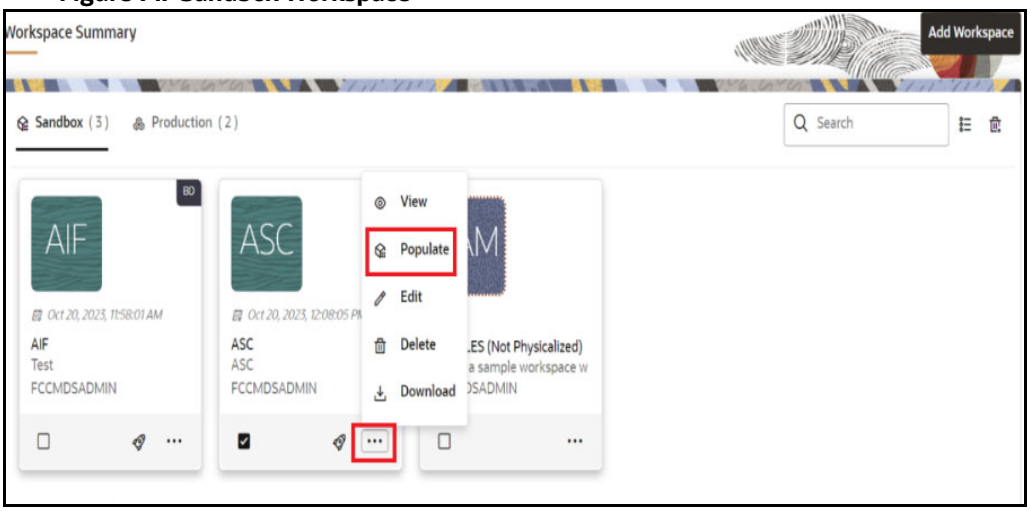
Figure 70: Summary



7.4 Populating the Sandbox Workspace

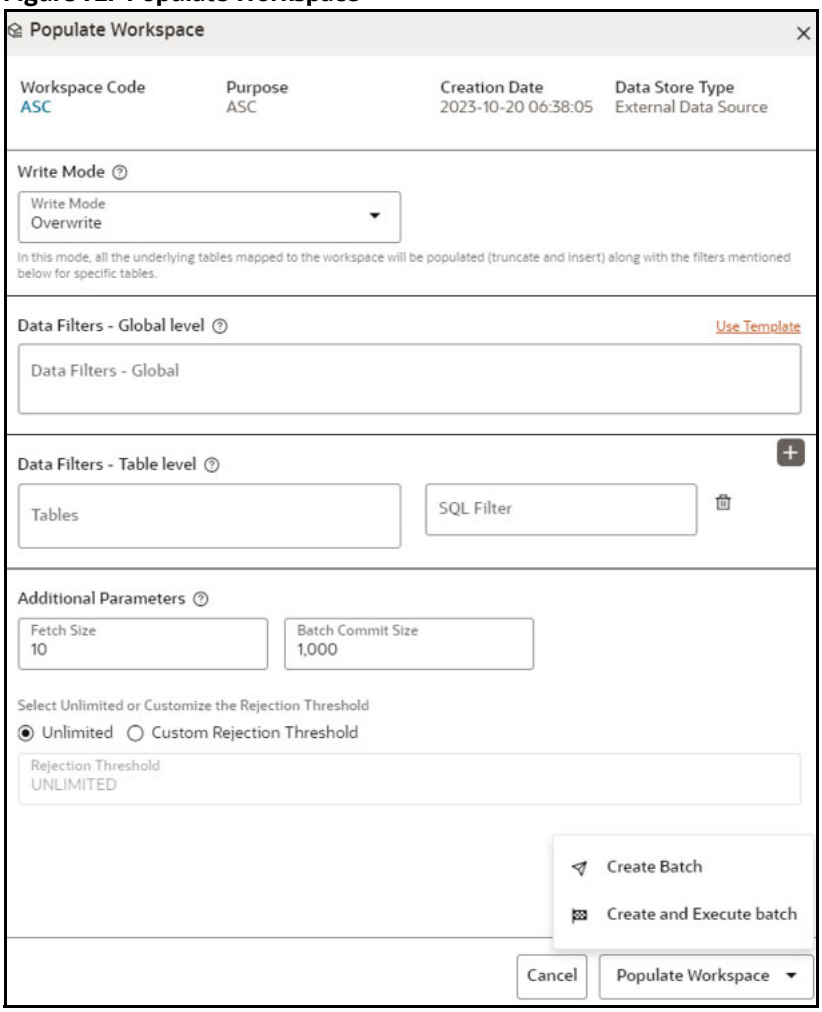
- From the workspace summary screen, choose to **populate sandbox** for the newly created sandbox.

Figure 71: Sandbox Workspace



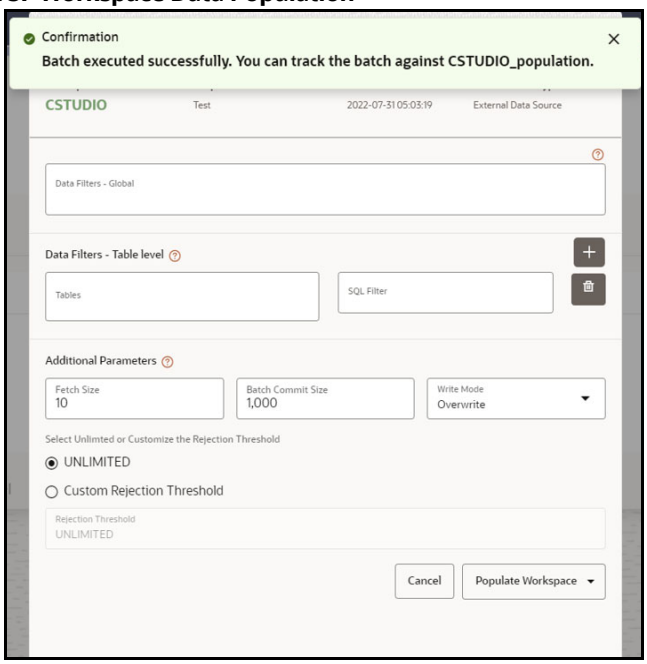
- Select **Create and Execute** batch option.

Figure 72: Populate Workspace



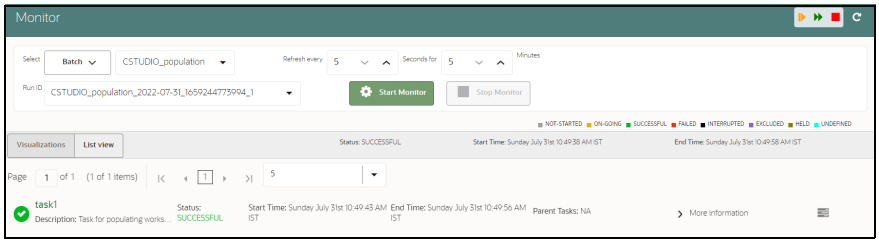
- Shows a Successful message on successfully triggering the **Workspace Data Population**.

Figure 73: Workspace Data Population



- Monitor the status of **Sandbox Workspace Population**.
  - `Launch the sandbox workspace.`
  - `Click` **`Orchestration`** `drop-down list and select` **`Monitor Batch.`**
- Select/Provide the Batch ID details using the drop-down to see the **status**.

Figure 74: Add/ Stop Monitor



NOTE If batch execution fails, check logs in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs/execution/<Batch_Executed_date>/<Sandbox_Workspace>/workspace-population` directory for debugging.

For example, `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/logs/execution/2024-02-20/AIF/workspace-population` directory.

7.5 Post Workspace Activity for ASC

Run the scenario conversion utility in **ASC BD** schema. For information about how to run, see **Using Scenario Conversion Utility** section in the OFS Compliance Studio User Guide.

7.6 Periodic Workspace Schema Cleanup for ASC

The system creates some intermediate temporary tables as part of the ASC workflow, which should be dropped periodically during cleanup activity. The following sample oracle statement will generate a drop table statement including all temp tables.

The generated drop table statement should be manually verified before using it as a drop table statement.

To generate drop table statement, execute the following:

```
select 'DROP TABLE '||TABLE_NAME||';' from user_tables where table_name like '%ASC_TEMP_%';
```

Example for the drop table statement:

DROP TABLE ASC_TEMP_1735;

7.7 Importing Workspace Metadata for ML4AML

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin.`

To identify the utilities and commands required for each use case, see Table 22.

Table 22 describes the utilities for the different use cases for both Sandbox and Production workspaces.

Table 22: Utilities for different Use Cases

Utility	Sandbox Workspace	Production Workspace	Command	Customer Risk Scoring	Customer Segmentation	AML Event Scoring	Shell Account Detection Scenario	ASC	Scenario Model
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh -w <workspace_wallet_alias>`	Yes	Yes	Yes	Yes	Yes	Yes
`importNotebooksAIF.sh`	Yes	Yes	`./importNotebooksAIF.sh -w <sandbox_workspace_code>`	Yes	Yes	No	No	No	No

Table 22: Utilities for different Use Cases

Utility	Sandbox Workspace	Production Workspace	Command	Customer Risk Scoring	Customer Segmentation	AML Event Scoring	Shell Account Detection Scenario	ASC	Scenario Model
importNotebooksAMLES.sh	Yes	Yes	./importNotebooksAMLES.sh -w <sandbox_workspace_code>	No	No	Yes	No	No	No
importNotebooksASC.sh	Yes	No	./importNotebooksASC.sh -w <sandbox_workspace_code	No	No	No	No	Yes	No
importNotebooksScenario.sh	Yes	Yes	./importNotebooksScenario.sh -w <sandbox_workspace_code>	No	No	No	Yes	No	No
enableRangeAutoPartition.sh (optional)	Yes	No	./enableRangeAutoPartition.sh -w <sandbox_wallet_alias>	Yes	Yes	Yes	Yes	Yes	No

Table 22: Utilities for different Use Cases

Utility	Sandbox Workspace	Production Workspace	Command	Customer Risk Scoring	Customer Segmentation	AML Event Scoring	Shell Account Detection Scenario	ASC	Scenario Model
`enableVPD.sh` (optional)	Yes	No	`./enableVPD.sh -w <sandbox_wallet_alias>`	No	No	No	No	Yes	No
`importNotebooksSM.sh`	Yes	Yes	`./importNotebookSM.sh -w <sandbox_workspace_code>`	No	No	No	No	No	Yes

NOTE

- **sandbox_wallet_alias** and **sandbox_workspace_code** are the place holders to be replaced with actual values used to create sandbox workspace.
- For more information about `enableRangeAutoPartition.sh`, see the Incremental Workspace Refresh section.
- For more information about `enableVPD.sh`, see the Fine Grain Data Access Control for Workspace section.
- `Sandbox.sh` should not be run on a schema, which is used as BD atomic schema in the Compliance Studio installer configuration file. The installer takes care of all the SQL objects.

7.8 Adding User Defined Transformation (UDT) as Python Module

The analyst user shares folder that contains python files to the administrator. To obtain the folder, see the **Feature Engineering of Scenario Model** section in the OFS Compliance Studio ML4AML Use Case Guide.

To add the UDT folder (python module), follow these steps:

1. Login to Unix machine where Compliance Studio is installed.
2. Navigate to `<MINICONDA_INSTALLATION_HOME>/miniconda3/envs/ml4aml_<version>/lib/python3.9/site-packages` directory.
3. Copy UDT folder and place it in the **site-packages** directory.

7.9 Optimizing SQL performance for ASC

You can further optimize SQL performance for ASC using this configuration. Users can configure SQL hints with PARALLEL or NO_PARALLEL hints. It comes with a default configuration as PARALLEL(8). Table **ml4aml_hint_config** holds the default configuration. Users can change these values as per database capacity and its DBA activity to come up with the best possible values that suit the database.

> **NOTE** Ensure all the tables are properly indexed per data growth experience. We assume this is a standard DBA activity as on when data keeps growing.

7.10 Incremental Workspace Refresh

As a part of incremental workspace refresh, all partitioned tables used in the workspace schema should be enabled to handle auto partition.

Enable partition table to auto partition, follow these steps:

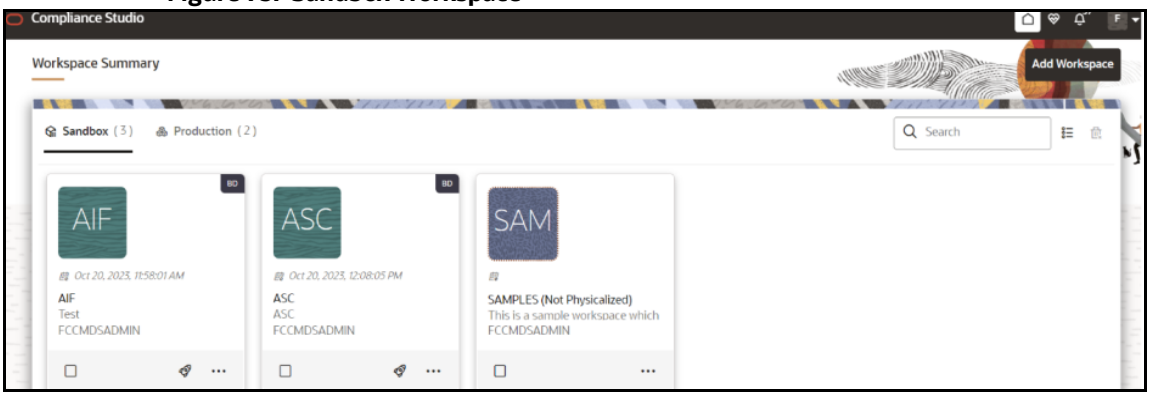
1. Configuring a list of partitioned tables to enable auto partition. Changes to be made in the **Sandbox** workspace schema are as follows:
   a. Update or insert the record in table "`ml4aml_range_auto_partition_config`" with `PARTITION_FLAG` as **Y**. Update other records which do not require to enable with `PARTITION_FLAG` as **N**.
2. Login to Compliance Studio installed UNIX Machine.
3. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin` directory.
4. Execute the following UNIX command:

```
./enableRangeAutoPartition.sh -w <sandbox_wallet_alias>
```

7.11 Launch the Sandbox Workspace

1. Click **Launch** icon from the workspace summary screen for launching the sandbox.

Figure 75: Sandbox Workspace



The Dashboard displays the following options:

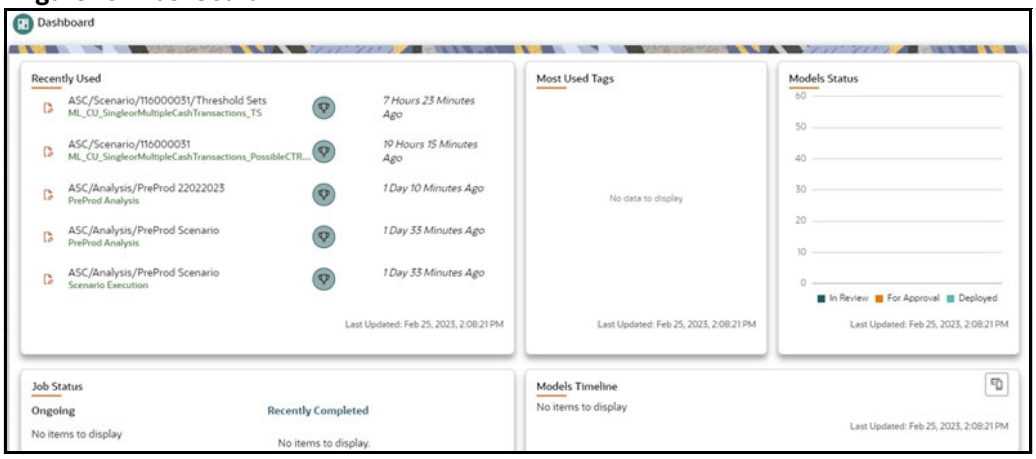
- Modeling
  - Datasets
  - Model Libraries
  - Model Techniques

- Model Catalog
- Pipelines
- Graphs

- Orchestration
  - Scheduler Service
- More
  - Ruleset Details
  - Merge and Split Global Entities
  - Model Actions
  - Audit Trail
  - Data Pipelines
  - Manual Decisioning

2. On **Modeling** menu, click **Pipelines** to start with ML.

Figure 76: Dashboard



7.12 Model Groups

OFS ML4AML is an application that provides foundational building blocks to train, deploy and monitor models tailored to address specific use cases relevant to the AML domain. It has a pre-defined set of transformations and over 300 attributes to help expedite the model development process.

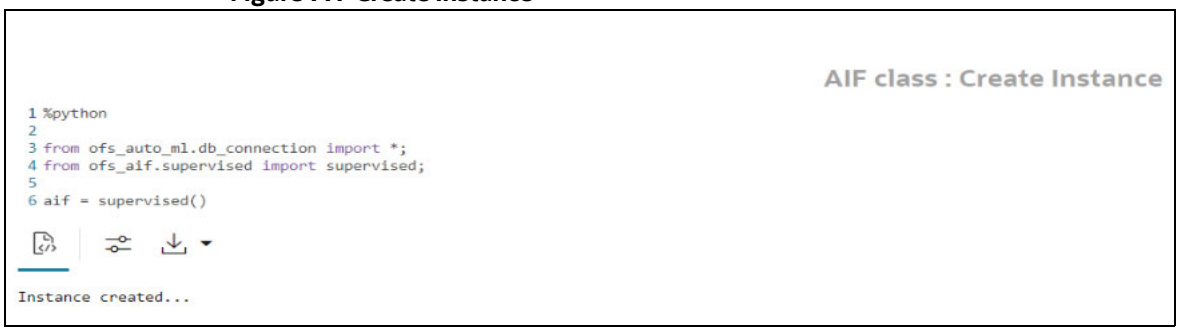
OFS ML4AML uses the Model Management and Governance (MMG) application to manage the various stages of the modeling lifecycle, such as sandbox creation, deployment to production, and ongoing monitoring.

7.12.1 Initialize the Session

Execute the following instructions in the Notebook to load the AIF4AML library:

```
import ofs_aif.supervised
```

Figure 77: Create Instance



7.12.2 Metadata to Create Model Group(s)

A model group is used to define the Line Of Business (LOB) of a model group. Six variables are provided in the model group, and the LOB value can be found in these variables. The model group can be used at the account and customer levels.

The following metadata is used to create model groups:

- **Account Type1 Code**: Client-specified account type classification for the usage of this account.
- **Account Type2 Code**: Client-specified account type classification for the usage of this account.
- **Business Domain(s)**: An account or customer (for example, institutional brokerage or retail brokerage).
- **Customer Type Code**: When a customer is involved in the execution, identify the type of customer.
- **Jurisdiction Code**: For an account or customer (for example, Americas, Europe, Middle East & Africa, India, and United States).
- **Account Status**: Account status (active, closed, and inactive).

Use the `aif.show_metadata_for_model_group_creation` API to view the metadata, which you can use to create model groups.

Execute the following paragraph to view the metadata for the model groups:

```
metadata_df = aif.show_metadata_for_model_group_creation()
z.show( metadata_df )
```

The output shows the default account and customer-level attributes enabled in the Table 23.

Table 23: Output Data for Model Groups

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer/Account	Business Domain(s)	Asset Management, Corporate or Wholesale Banking, Employee Information, General, Institutional Broker-Dealer, Other values as specified by the client, Retail Banking, Retail Brokerage, or Private Client.
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer/Account	Jurisdiction Code	Americas, Europe, Middle East & Africa, India, United States.

Table 23: Output Data for Model Groups

Account	Account Type1 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/Time/Certificate of Deposit.
Account	Account Type2 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/Time/Certificate of Deposit.

7.12.3 Create the Input Dataframe for Model Groups

Create the Input Dataframe as shown in the following example:

```
pdf = pd.DataFrame(
{'MODEL_GROUP_NAME'     : ["LOB13","LOB13"],
  'ENTITY_NAME'         : ["Account", "Account"],
   ATTRIBUTE_NAME'      : ["Business Domain(s)","Jurisdiction Code"],
  'ATTRIBUTE_VALUE'     : ["General","Europe Middle East & Africa"],
  'LABEL_FILTER'        : ["ACCT","ACCT"],
  'FEATURE_TYPE_FILTER' :
["CASH_TRXN,WIRE_TRXN,MI_TRXN","CASH_TRXN,WIRE_TRXN,MI_TRXN"]
                      })
z.show( pdf )
```

- **MODEL_GROUP_NAME**: The administrator-defined unique identifier for the model group. Only alphanumeric characters underscore, hyphens, and space are the special characters allowed.
- **ENTITY_NAME**: Logical Entity Name as displayed in the metadata section.
- **ATTRIBUTE_NAME**: Logical Attribute Name as displayed in the metadata section.
- **ATTRIBUTE_VALUE**: Logical Attribute Value as displayed in the metadata section.

7.12.3.1 Vertical and Horizontal Filters

The following filters are used as input data frames for model group creation:

- **LABEL_FILTER**: Use this filter to identify entities and labels from the table `AIF_INVESTIGATED_ENTITY`. It is a model group creation parameter that is mapped to the LABELLED_SCENARIO column in the `AIF_INVESTIGATED_ENTITY` table.
  - For **Unsupervised**, LABEL_FILTER to be passed as **UNSUPERVISED**
  - Foe **AMLES**, LABEL_FILTER to be passed as **AMLES**
- **FEATURE_TYPE_FILTER**: Use this filter to identify the features required for the model group. It is a model group creation parameter that is mapped to the ATTRIBUTE_NM column in the `aif_vertical_filter_lookup` table. Options include:
  - CASH_TRXN: Features specific to Cash Transactions
  - WIRE_TRXN: Features specific to Wire Transactions

- MI_TRXN: Features specific to Monitory Instrument
- TRADE: Features specific to Trading
- BACK_OFFICE_TRXN: Features specific to Back-office Transactions

NOTE

- A vertical filter (FEATURE_TYPE_FILTER) is applicable only for supervised model groups.
- You can provide the list of features in the FEATURE_TYPE_FILTER that must be used while creating the supervised model group in the Admin Notebook.
- By default, it considers all features in the filter.
- In the case of Unsupervised, this is not applicable.

Any above combination such as comma (,) separated CASH_TRXN, MI_TRXN, or MI_TRXN, and CASH_TRXN, WIRE_TRXN is also allowed. The FEATURE_TYPE_FILTER helps to reduce the memory requirement at the model group level, so ensure that you optimize the storage by choosing only the required features.

- **Table AIF_VERTICAL_FILTER_LOOKUP**: Use this filter as a lookup table for feature list to feature type.

  Execute the following paragraph to view data for the filters:

```
pdf = pd.DataFrame(
{'MODEL_GROUP_NAME'     : ["LOB13","LOB13"],
  'ENTITY_NAME'         : ["Account", "Account"],
   ATTRIBUTE_NAME'      : ["Business Domain(s)","Jurisdiction Code"],
  'ATTRIBUTE_VALUE'     : ["General","Europe Middle East & Africa"],
  'LABEL_FILTER'        : ["ACCT","ACCT"],
  'FEATURE_TYPE_FILTER' :
["CASH_TRXN,WIRE_TRXN,MI_TRXN","CASH_TRXN,WIRE_TRXN,MI_TRXN"]
                      })
z.show( pdf )
```

The output appears as shown in the Table 24.

Table 24: Output Data for Filters

MODEL_GROUP_NAME	ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE	LABEL_FILTER	FEATURE_TYPE_FILTER
LOB13	Account	Business Domain(s)	General	ACCT	CASH_TRXN, WIRE_TRXN, MI_TRXN
LOB13	Account	Jurisdiction Code	Europe Middle East & Africa	ACCT	CASH_TRXN, WIRE_TRXN, MI_TRXN

7.12.4 Show Unused Attributes for Model Group Creation

Use the `aif.show_unused_attributes_in_model_group_metadata` API to view the unused attributes after the model group is created. See the following sections to know how to enable the unused attributes.

Execute the following paragraph to view a list of unused attributes:

```
z.show( aif.show_unused_attributes_in_model_group_metadata())
```

The output appears as shown in the Table 25.

Table 25: Output Data for Unused Attributes

Entity	Attributes
Account	Account Status
Customer	Employee Relationship Type Code
Customer	Employer Industry
Customer	Occupation
Customer	Resident Country
Customer	Registration Type
Customer	Source System

7.12.5 Enable or Disabling Unused Attributes for Model Group Creation

Use the `aif.show_unused_attributes_in_model_group_metadata()`

API to view the unused attributes after the model group is created.

The following is the input value for the paragraph:

- **entity_attribute_df**: This is the input data frame formed with respect to the **show_unused_attributes_in_model_group_metadata()**. The Data frame with the ENTITY & ATTRIBUTES column must be provided.
- **disable**: This value has two options, that is, TRUE or FALSE. The value is FALSE by default, which means that the attributes are enabled under metadata for model group creation. If you enter TRUE, then the attributes are disabled.

Execute the following paragraph to view a list of unused attributes:

```
z.show( aif.show_unused_attributes_in_model_group_metadata())
```

The output appears as shown in the Table 26.

Table 26: Output Data for Unused Attributes

Entity	Attributes
Customer	Customer Status
Account	Account Status

7.12.5.1 Enable Unused Attributes

Execute the following paragraph to enable the unused attributes:

```
aif.enable_attributes_as_model_group_metadata(pdf , disable = False )
z.show( aif.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 27.

Table 27: Output Data showing Enabled Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	Asset Management, Corporate or Wholesale Banking, Employee Information, General, Institutional Broker-Dealer, Other values as specified by the client, Retail Banking, Retail Brokerage, or Private Client.
Customer	Customer Status	Active, Inactive, Not a Customer, Pending.
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer	Jurisdiction Code	Europe Middle East & Africa, India, United States.
Account	Account Status	Active, Closed, Dormant, Inactive, Purge.
Account	Account Type1 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/Time/Certificate of Deposit.
Account	Account Type2 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/Time/Certificate of Deposit.

7.12.5.2 Disable Unused Attributes

Execute the following paragraph to disable the unused attributes:

```
aif.enable_attributes_as_model_group_metadata(pdf , disable = True )
z.show( aif.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 28.

Table 28: Output Data showing Disabled Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	Asset Management, Corporate or Wholesale Banking, Employee Information, General, Institutional Broker-Dealer, Other values as specified by the client, Retail Banking, Retail Brokerage, or Private Client.
Customer	Customer Type	Financial Institution, Individual, Other Organization.

Table 28: Output Data showing Disabled Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Jurisdiction Code	Europe Middle East & Africa, India, United States.
Account	Account Type1 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/ Time/Certificate of Deposit.
Account	Account Type2 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/ Time/Certificate of Deposit.

7.12.6 Add or Remove Attributes to the Model Group Metadata

Use the `aif.add_new_attribute_values_for_model_group_metadata()` API to add or remove attributes after the model group is created.

The following are the input values for the paragraph:

- **entity_attribute_value_df**: The input data frame has the Data frame with the provided entities, Attributes, and Values columns.
- **remove**: This value has two options, that is, TRUE or FALSE. If you enter TRUE, then the attribute values are removed under metadata for model group creation.

Execute the following paragraph to view a list of unused attributes:

```
pdf = pd.DataFrame({'ENTITY' : ["Customer"],
  'ATTRIBUTE_NAME' : ["Jurisdiction Code"],
 'ATTRIBUTE_VALUE' : ["Australia"],
  'ATTRIBUTE_CODE' : ["AU"]
                   })


z.show(pdf)
```

The output appears as shown in the Table 29.

Table 29: Output Data for Adding or Removing Attributes

ENTITY	ATTRIBUTE_NAME	ATTRIBUTE_VALUE	ATTRIBUTE_CODE
Customer	Jurisdiction Code	Australia	AU

7.12.6.1 Add Attributes

Execute the following paragraph to add the attributes:

```
aif.add_new_attribute_values_for_model_group_metadata(pdf, remove = False)
z.show( aif.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 30.

Table 30: Output Data showing Added Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	Asset Management, Corporate or Wholesale Banking, Employee Information, General, Institutional Broker-Dealer, Other values as specified by the client, Retail Banking, Retail Brokerage, or Private Client.
Customer	Customer Status	Active, Inactive, Not a Customer, Pending.
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer	Jurisdiction Code	Australia, Europe, Middle East & Africa, India, United States.
Account	Account Status	Active, Closed, Dormant, Inactive, Purge.
Account	Account Type1 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/ Time/Certificate of Deposit.
Account	Account Type2 Code	Checking, Credit Card, Health Savings, Insurance Policy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/ Time/Certificate of Deposit.

7.12.6.2 Remove Attributes

Execute the following paragraph to remove the attributes:

```
aif.add_new_attribute_values_for_model_group_metadata(pdf, remove = True)
z.show( aif.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 31.

Table 31: Output Data showing Removed Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	Asset Management, Corporate or Wholesale Banking, Employee Information, General, Institutional Broker- Dealer, Other values as specified by the client, Retail Banking, Retail Brokerage, or Private Client.
Customer	Customer Status	Active, Inactive, Not a Customer, Pending.
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer	Jurisdiction Code	Europe Middle East & Africa, India, United States.
Account	Account Status	Active, Closed, Dormant, Inactive, Purge.

Table 31: Output Data showing Removed Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Account	Account Type1 Code	Checking, Credit Card, Health Savings, Insurance Pol-icy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/Time/Certificate of Deposit.
Account	Account Type2 Code	Checking, Credit Card, Health Savings, Insurance Pol-icy, Investment, Loan, Money Market, Other values as specified by the client, Others, Retirement, Savings, Stored Value Card, Term/Time/Certificate of Deposit.

7.12.7 Add Model Groups

Use the `aif.add_model_groups ()` API to view the list of available model groups.

The following is the input value for the paragraph:

meta_data_df: This is the input pandas data frame formed using the available metadata.

Execute the following paragraph to add Model Group(s):

```
aif.add_model_groups(pdf)
```

The preceding code returns a confirmation message on successfully adding model groups or error messages for failures.

7.12.8 Import User Model Templates

The steps for importing the user notebook into your workspace are:

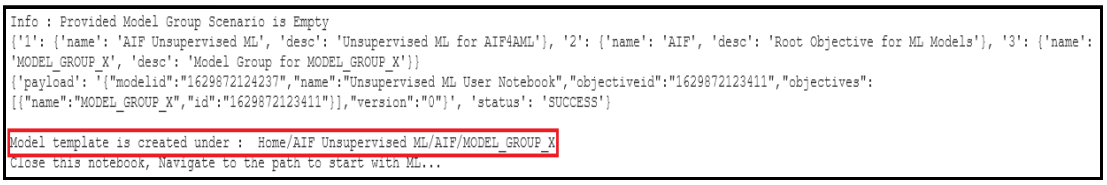
1. Execute the following line of code which contains the **aif.import_model_template** API. Here **meta_data_df** refers to the same pandas dataframe created during creation of your model group.

   ```
   aif.import_model_template( meta_data_df = pdf,

   model_group_scenario = None )
   ```

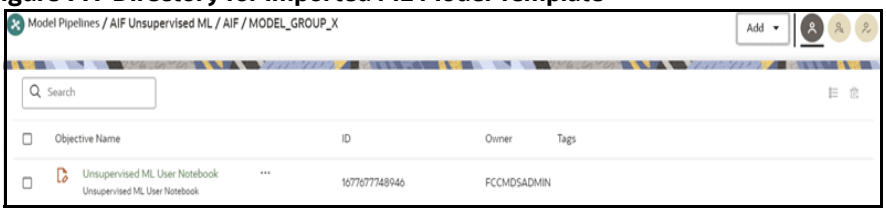
   A message will be displayed saying that the model template has been created under "this" particular path.

   Figure 78: Successful Message for Imported Template



2. Navigate to the directory mentioned in the output message to find the user notebook for your created model group.

Figure 79: Directory for Imported ML Model Template



7.12.9 View the List of Available Model Groups

Use the `aif.show_model_groups` API to view the list of available model groups.

Execute the following paragraph to view a list of available model groups:

```
z.show( aif.show_model_groups())
```

The output appears as shown in the Table 32.

Table 32: Output Data for Model Groups

MODEL_GROUP_ID	MODEL_GROUP_NAME	ENTITY_LOGICAL_NAME	ATTRIBUTE_LOGICAL_NAME	ATTRIBUTE_LOGICAL_VALUE
401	LOB1	Customer	Business Domain(s)	General
803	BUS_DMN_LIST_TX_E	Account	Business Domain(s)	General
1201	LOB13	Account	Business Domain(s)	General
1201	LOB13	Account	Jurisdiction Code	Europe Middle East & Africa

7.12.10 Modify Model Groups

Use the `aif.modify_model_groups` API to modify an existing model group.

The following is the input value for the paragraph:

meta_data_df: This is the input pandas data frame that is formed using the available metadata.

To view a list of available model group(s), use the following paragraph:

```
aif.modify_model_groups(pdf)
```

A successful message is displayed when you add model groups.

```
Successful: Model group modification
```

7.13 Model Groups for Behavioral Scenario Model

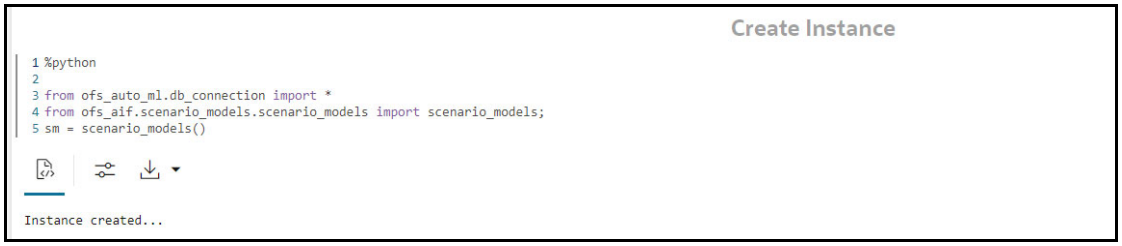
OFS ML4AML is an application that provides foundational building blocks to train, deploy and monitor models tailored to address specific use cases relevant to the AML domain. It has a pre-defined set of transformations and over 300 attributes to help expedite the model development process.

OFS ML4AML uses the Model Management and Governance (MMG) application to manage the various stages of the modeling lifecycle, such as sandbox creation, deployment to production, and ongoing monitoring.

7.13.1 Initialize the Session

Execute the following instructions in the Notebook to load the AIF4AML library:

Figure 80: Create Instance



7.13.2 Metadata to Create Model Group(s)

A model group is used to define the Line Of Business (LOB) of a model group. Six variables are provided in the model group, and the LOB value can be found in these variables. The model group can be used at the account and customer levels.

The following metadata is used to create model groups:

- **Account Type1 Code**: Client-specified account type classification for the usage of this account.
- **Account Type2 Code**: Client-specified account type classification for the usage of this account.
- **Business Domain(s)**: An account or customer (for example, institutional brokerage or retail brokerage).
- **Customer Type Code**: When a customer is involved in the execution, identify the type of customer.
- **Jurisdiction Code**: For an account or customer (for example, Americas, Europe, Middle East & Africa, India, and United States).
- **Account Status**: Account status (active, closed, and inactive).

Use the `sm.show_metadata_for_model_group_creation` API to view the metadata, which you can use to create model groups.

Execute the following paragraph to view the metadata for the model groups:

```
metadata_df = sm.show_metadata_for_model_group_creation()
z.show( metadata_df )
```

The output shows the default account and customer-level attributes enabled in the Table 33.

Table 33: Output Data for Model Groups

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Account	Business Domain (or Domains)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/Wholesale Banking, Employee Information, Asset Management, To Be Updated

Table 33: Output Data for Model Groups

Customer	Customer Type	Financial Institution, Individual, Other Organization.
Account	Jurisdiction Code	Australia, Americas
Account	Account Type1 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card, Term/ Time/Certificate of Deposit
Account	Account Type2 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card, Term/ Time/Certificate of Deposit
Customer	Occupation	Aero/Aviation/Defense, Agriculture, Forestry & Fishing, Airlines, Auto, Entertaiment, Others,Build & Grounds Maint, Construction, Electronics, Finance/Economics, Firm-specified, Banking
Customer	Business Domain (or Domains)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/Wholesale Banking, Employee Information, Asset Management, Other values as specified by the client
Customer	Jurisdiction Code	India, United States, Europe Middle East & Africa, Australia, Americas
Customer	Resident Country	Alpha country code

7.13.3 Show Unused Attributes for Model Group Creation

Use the `sm.show_unused_attributes_in_model_group_metadata` API to view the unused attributes after the model group is created. See the following sections to know how to enable the unused attributes.

Execute the following paragraph to view a list of unused attributes:

```
z.show( sm.show_unused_attributes_in_model_group_metadata())
```

The output appears as shown in the Table 34.

Table 34: Output Data for Unused Attributes

Entity	Attributes
Account	Account Status
Customer	Secondary Citizenship
Customer	Credit Rating Source
Customer	Industry
Customer	Customer Watch List Identifier Source
Customer	Customer NAICS Code
Customer	Customer Status

7.13.4 Enable or Disabling Unused Attributes for Model Group Creation

Use the `sm.show_unused_attributes_in_model_group_metadata()` API to view the unused attributes after the model group is created.

The following is the input value for the paragraph:

- **entity_attribute_df**: This is the input data frame formed with respect to the **show_unused_attributes_in_model_group_metadata()**. The Data frame with the ENTITY & ATTRIBUTES column must be provided.
- **disable**: This value has two options, that is, TRUE or FALSE. The value is FALSE by default, which means that the attributes are enabled under metadata for model group creation. If you enter TRUE, then the attributes are disabled.

7.13.4.1 Enable Unused Attributes

Execute the following paragraph to enable the unused attributes:

```
sm.enable_attributes_as_model_group_metadata(pdf , disable = False )
z.show( sm.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 35.

Table 35: Output Data showing Enabled Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/ Wholesale Banking, Employee Information, Asset Management, Other values as specified by the client

Table 35: Output Data showing Enabled Attributes

Customer	Occupation	Aero/Aviation/Defense, Agriculture, Forestry & Fishing, Airlines, Auto, Entertaiment, Others, Build & Grounds Maint, Construction, Electronics, Finance/ Economics, Firm-specified, Banking
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer	Jurisdiction Code	India, United States, Europe Middle East & Africa, Australia, Americas
Account	Jurisdiction Code	Australia, Americas
Account	Account Type1 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card, Term/Time/ Certificate of Deposit
Account	Account Type2 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card, Term/Time/ Certificate of Deposit
Account	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking,Corporate/ Wholesale Banking, Employee Information, Asset Management, To Be Updated

7.13.4.2 Disable Unused Attributes

Execute the following paragraph to disable the unused attributes:

```
sm.enable_attributes_as_model_group_metadata(pdf , disable = True )
```

z.show(sm.show_metadata_for_model_group_creation())

The output appears as shown in the Table 36.

Table 36: Output Data showing Disabled Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/ Wholesale Banking, Employee Information, Asset Management, Other values as specified by the client
Customer	Occupation	Aero/Aviation/Defense, Agriculture, Forestry & Fishing, Airlines, Auto, Entertaiment, Others,Build & Grounds Maint, Construction, Electronics, Finance/Economics, Firm-specified, Banking
Customer	Customer Type	Financial Institution, Individual, Other Organization.

Table 36: Output Data showing Disabled Attributes

Customer	Jurisdiction Code	India, United States, Europe Middle East & Africa, Australia, Americas
Account	Jurisdiction Code	Australia, Americas
Account	Account Type1 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy,Investment, Loan, Money Market,Others, Retirement, Savings,Stored Value Card, Term/ Time/Certificate of Deposit
Account	Account Type2 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy,Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card,Term/ Time/Certificate of Deposit
Account	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking,Corporate/ Wholesale Banking, Employee Information, Asset Management, To Be Updated

7.13.5 Add or Remove Attributes to the Model Group Metadata

Use the `sm.add_new_attribute_values_for_model_group_metadata()` API to add or remove attributes after the model group is created.

The following are the input values for the paragraph:

- **entity_attribute_value_df**: The input data frame has the Data frame with the provided entities, Attributes, and Values columns.
- **remove**: This value has two options, that is, TRUE or FALSE. If you enter TRUE, then the attribute values are removed under metadata for model group creation.

Execute the following paragraph to view a list of unused attributes:

```
pdf = pd.DataFrame({'ENTITY' : ["Customer"],

'ATTRIBUTE_NAME' : ["Processing Batch"],

 'ATTRIBUTE_VALUE' : ["Hong Kong"],

  'ATTRIBUTE_CODE' : ["HKDLY"]

   })

z.show(pdf)
```

The output appears as shown in the Table 37.

Table 37: Output Data for Adding or Removing Attributes

ENTITY	ATTRIBUTE_NAME	ATTRIBUTE_VALUE	ATTRIBUTE_CODE
Customer	Processing Batch	Hong Kong	HKDLY

7.13.5.1 Add Attributes

Execute the following paragraph to add the attributes:

```
sm.add_new_attribute_values_for_model_group_metadata(pdf, remove = False)
z.show( sm.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 38.

Table 38: Output Data showing Added Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking,Corporate/ Wholesale Banking, Employee Information, Asset Management, Other values as specified by the client
Customer	Occupation	Aero/Aviation/Defense, Agriculture, Forestry & Fishing, Airlines, Auto, Entertaiment, Others, Build & Grounds Maint, Construction, Electronics,Finance/Economics, Firm-specified, Banking
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer	Jurisdiction Code	India, United States, Europe Middle East & Africa, Australia, Americas
Account	Jurisdiction Code	Australia, Americas
Account	Account Type1 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement,Savings, Stored Value Card, Term/Time/ Certificate of Deposit
Account	Account Type2 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card,Term/Time/ Certificate of Deposit

Table 38: Output Data showing Added Attributes

Account	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/ Wholesale Banking, Employee Information, Asset Management, To Be Updated
Account	Processing Batch	Singapore Daily
Customer	Processing Batch	HING KONG, Singapore Daily

7.13.5.2 Remove Attributes

Execute the following paragraph to remove the attributes:

```
sm.add_new_attribute_values_for_model_group_metadata(pdf, remove = True)
z.show( sm.show_metadata_for_model_group_creation())
```

The output appears as shown in the Table 39.

Table 39: Output Data showing Removed Attributes

ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE
Customer	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/ Wholesale Banking, Employee Information, Asset Management, Other values as specified by the client
Customer	Occupation	Aero/Aviation/Defense,Agriculture, Forestry & Fishing, Airlines, Auto, Entertaiment, Others, Build & Grounds Maint, Construction, Electronics,Finance/Economics, Firm-specified, Banking
Customer	Customer Type	Financial Institution, Individual, Other Organization.
Customer	Jurisdiction Code	India, United States, Europe Middle East & Africa, Australia, Americas
Account	Jurisdiction Code	Australia, Americas

Table 39: Output Data showing Removed Attributes

Account	Account Type1 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card, Term/Time/ Certificate of Deposit
Account	Account Type2 Code	Credit Card, Checking, Other values as specified by the client, Health Savings, Insurance Policy, Investment, Loan, Money Market, Others, Retirement, Savings, Stored Value Card, Term/Time/ Certificate of Deposit
Account	Business Domain(s)	General, Institutional Broker Dealer, Retail Brokerage/Private Client, Retail Banking, Corporate/ Wholesale Banking, Employee Information, Asset Management, To Be Updated

7.13.6 Create a Model Group with a Single Data Segment

Create a single individual model group with required data filters. For example, Let's assume there is a need to create a model group for a single segment for Singapore.

To create a single segment for the model group follow these sections.

7.13.6.1 Add Model Groups

Use the `sm.add_model_groups ()` API to add a new model group.

The following is the input value for the paragraph:

meta_data_df: This is the input pandas data frame formed using the available metadata.

For creating a new model group, the input metadata dataframe is shown below as an example:

```
meta_data_df = pd.DataFrame({'MODEL_GROUP_NAME'   : ["Logicalgroup_Sin-
gapore"],

'ENTITY_NAME'          : ["Customer"],

'ATTRIBUTE_NAME'       : ["Processing Batch"],

'ATTRIBUTE_VALUE'      : ["Daily Batch"],

 'LABEL_FILTER'         : [""],

 'FEATURE_TYPE_FILTER' : [""]

 })

 z.show( meta_data_df )
```

Execute the following paragraph to add Model Group(s):

```
sm.add_model_groups(pdf)
```

The preceding code returns a confirmation message on successfully adding model groups or error messages for failures.

7.13.6.2 Specify Unique Model Information

This paragraph is used to store unique model information for a model group.

Unique information such as **Model Name, Model Display Name, and Model Description** are stored against (Model Group Name, Model Name, and Entity Name/Focus) as key.

The parameters are:

- **model_name**: Provide a name of the model that gives the purpose of the model. This field is limited to 15 characters; special characters are not allowed except for the underscore (_).

  For example;

  - If the purpose of the model within the model group/segment is to create an RMF scenario Model that internally uses dispositions/target data from the RMF scenario. In that case, the model name can be provided as RMF.
  - If the purpose of the model within the model group is to create a model for large reportable transactions, then the model name can be provided as LRT.
  - If the purpose of the model within the model group is to analyze cash transaction behavior, then the model name can be provided as CASH.

  > **NOTE** The model name is an additional identifier for the data's nature or the model's purpose.

- **model_display_name**: Provide the model's name displayed in the case management whenever the model produces alerts/events.
- **model_description**: Provide a description of the model displayed in the case management whenever the model produces alerts/events.
- **overwrite**: This flag is used to set the overwrite condition. By default, it is set to false.
  - If **overwrite = FALSE**, insert a new entry for model information or return an exception if the entry already exists.
  - If **overwrite = TRUE**, update the model_display_name and model_description of the existing entry for model information.

Other required parameters are **model_group_name** and **entity_name/focus.** These are provided while creating the input dataframe for model group in the Modify Model Groups section.

Figure 81: Specify Unique Model Information

The output of this paragraph is stored in the **ML4AML_MODEL_METADATA_MASTER** table.

This information is used as part of ECM integration for persisting Model Name, Model Description, and a **unique Model ID**.

For more information on these tables, see the OFS Compliance Studio Data Model Reference Guide.

7.13.6.3 Import User Model Templates

The steps for importing the user notebook into your workspace are:

1. Execute the following line of code which contains the **sm.import_model_template** API. Here **meta_data_df** refers to the same pandas dataframe created during creation of your model group.

   ```
   sm.import_model_template( meta_data_df = pdf, overwrite=False, model_name='RMF' )
   ```

   A message will be displayed saying that the model template has been created under this particular path.

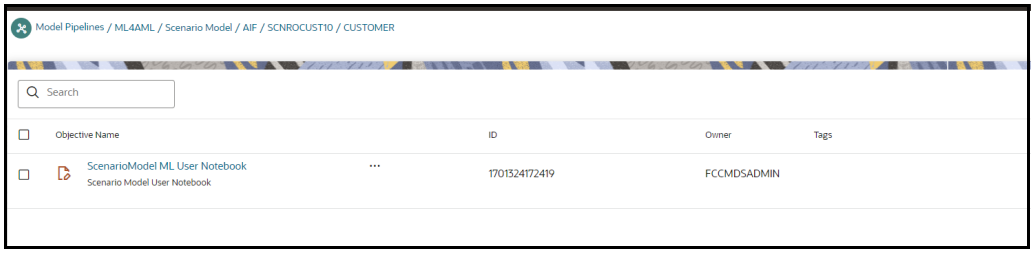
   Figure 82: Successful Message for Imported Template

   ```
   Template imported successfully - 'ScenarioModel ML User Notebook' .

   Model template is created under :  Model Pipelines/ML4AML/Scenario Model/AIF/SCNROCUST10/RMF_CUSTOMER
   Close this notebook, Navigate to Scheduler Dashboard & execute ML4AML/Scenario Model/Batch/Base Features/Base Features
   Post successful batch execution, Navigate to the path mentioned above to start with ML...
   ```

2. Navigate to the directory mentioned in the output message to find the user notebook for your created model group.

   Figure 83: Directory for Imported ML Model Template



7.13.6.4 View the List of Available Model Groups

Use the `sm.show_model_groups` API to view the list of available model groups.

Execute the following paragraph to view a list of available model groups:

```
z.show( sm.show_model_groups())
```

The output appears as shown in the Table 40.

Table 40: Output Data for Model Groups

MOEL_GROUP_ID	MODEL_GROUP_NAM E	ENTITY_LOGICAL _NAME	ATTRIBUTE_LOGIC AL_NAME	ATTRIBUTE_LOGICAL_VALUE
501	MG1	Customer	Business Domain(s)	General
501	MG1	Customer	Jurisdiction Code	Americas

7.13.7 Create a Model Group with Multiple Data Segments

Often, a single model must be created for multiple segments because there may not be sufficient labels for each segment individually or the segment is similar enough that a single model is sufficient.

In such cases, pooling data across multiple segments and building a single model that caters to all these segments is preferable.

For example, consider customer segments in Singapore, Malaysia, and Thailand. The business may determine that the behavior of customers across these three countries are similar and a single model should be built for all three segments.

To do this, first, a model group should be created for each of the segments (Singapore, Malaysia, and Thailand) by following the Create a Model Group with a Single Data Segment section.

Now create a logical model group e.g., Singapore, Malaysia, and Thailand that combines these three segments. Once the logical model template is imported, navigate to the model template path to start creating the model.

> **NOTE** The data preparation batch will be run only for the constituent model groups (Singapore, Malaysia, and Thailand) and not for this new logical model group.

For creating a logical model group, the input metadata dataframe is shown below as an example:

```
meta_data_df = pd.DataFrame({'MODEL_GROUP_NAME'   : ["Logicalgroup_Sin-
gapore_Malaysia_Thailand"],

'ENTITY_NAME'         : ["Customer"],

'ATTRIBUTE_NAME'      : ["Processing Batch"],

'ATTRIBUTE_VALUE'     : ["Daily Batch"],

 'LABEL_FILTER'        : [""],
```

```
'FEATURE_TYPE_FILTER' : [""]

})

z.show( meta_data_df )
```

7.13.8 Modify Model Groups

Use the `sm.modify_model_groups` API to modify an existing model group.

The following is the input value for the paragraph.

meta_data_df: This is the input pandas data frame that is formed using the available metadata.

```
pdf = pd.DataFrame({'MODEL_GROUP_NAME'     : ["Group_Singapore"],

'ENTITY_NAME'          : ["Customer"],

'ATTRIBUTE_NAME'       : ["Business Domain(s)"],

'ATTRIBUTE_VALUE'      : ["General"],

'LABEL_FILTER'         : [""],

'FEATURE_TYPE_FILTER'  : [""],

 'ACTION_TYPE'          : ["ADD"],

 'DISABLE_GROUP'        : ["N"]

 })

z.show( pdf )

sm.modify_model_groups(pdf)
```

A successful message is displayed when you modified the model groups.

```
Successful: Model group modification
```

The output appears as shown in the Table 41.

Table 41: Output Data for Model Group Modification

MODEL_GROUP_NAME	ENTITY_NAME	ATTRIBUTE_NAME	ATTRIBUTE_VALUE	ACTION_TYPE	DISABLE_GROUP
Group_Singapore	Customer	Business Domain(s)	General	ADD	N

7.14 Obtaining SAR Labels for Supervised Use Cases

7.14.1 Obtain the SAR Information for Sandbox

7.14.1.1 Populate Investigated Entity Details

7.14.1.1.1 Obtain the SAR from CRR/ECM

Use `aif.load_sar_data ()` API to load the Suspicious Activity Report (SAR) entities details from the Compliance Regulatory Reporting (CRR) application and Non-SAR entities from ECM into AIF.

The data will be loaded into the AIF `table aif_investigated_entity` table.

Figure 84: Aif Load SAR Data

```
3 CRR_conn = cx_Oracle.connect('/@CRR_Atomic_Wallet_Alias')
4 ECM_conn = cx_Oracle.connect('/@ECM_Atomic_Wallet_Alias')
5
6 aif.load_sar_data(20010101, 20991231, CRR_conn, ECM_conn)
7
```

The following parameters are the input value for the paragraph:

- **from_date**: From date range in **YYYYMMDD** format for SAR/Alert creation date.
- **to_date**: To date range in **YYYYMMDD** format for SAR/Alert creation date.
- **CRR_conn**: CRR Connection object.
- **ECM_conn**: ECM Connection object.

> **NOTE**
> - Register Oracle wallet entries/aliases for CRR & ECM Atomic schema to connect within Compliance Studio.
> - Use the aliases mentioned here to create/register entries. If aliases are being created with some other name, use them accordingly in the Admin Notebook.

7.14.1.1.2 Obtain the SAR from the CSV file

Use `aif.load_sars_from_csv()` API to load the SAR and Non-SAR entities into a CSV file.

Figure 85: Aif Load Sars from CSV

```
3 INVdata = aif.load_sars_from_csv('/scratch/fccstudio/SARCSV.csv', 'Y')
4
```

The following parameters are the input value for the paragraph:

- **filename**: Complete path of the CSV file.
- **headerIncluded**: This parameter has two options: **Y** or **N**. If the file has data with the header, then Y or N.

> **NOTE**
> - The date should be in `YYYYMMDD HH24:MI:SS` format.
> - Records should be comma-separated (CSV).

Ensure that the following columns are available in the CSV files with the required values:

- **ENTITY_ID**: Customer Id or Account Id
- **SUSPICIOUS_FLAG**: This parameter has two options: **Y** or **N**. If E-file for Regulatory body has been sent for Customer or Account, then Y or N.
- **ALERT_DATE**: SAR/EVENT generated to date from Customers and Accounts
- **CREATED_ON**: CSV file creation date
- **CREATED_BY**: CSV file created by
- **UPDATED_ON**: CSV file updated date
- **UPDATED_BY**: CSV file updated by
- **LABELLED_SCENARIO**: This value has the following options:
  - **CUST**: `For customer-level SAR`
  - **ACCT**: `For account level SAR`
- **ENTITY_CD**: This value has the following options:
  - `If entity type is customer`
  - `If entity type is the account`

7.14.1.2 Obtain the SAR classification from the CRR database

The `aif.get_case_data_and_sar_classification()` API gets SAR classification from CRR schema, merge with entity ID (Customer ID) in ECM, and stores as metadata in AIF schema table, `aif_case_information`.

Figure 86: Aif Get Case Data

```
CRR_conn = cx_Oracle.connect('/@CRR_Atomic_Wallet_Alias')
ECM_conn = cx_Oracle.connect('/@ECM_Atomic_Wallet_Alias')

aif.get_case_data_and_sar_classification(20010101, 20991231, CRR_conn, ECM_conn)
```

The `aif_case_information` table columns are as follows:

- ENTITY_ID
- CASE_ID
- SAR_CLASSIFICATION
- FILING_AM
- CONTINUING_SAR
- FILING_DATE

The following parameters are the input value for the paragraph:

- **from_date**: From date range in **YYYYMMDD** format.
- **to_date**: To date range in **YYYYMMDD** format.
- **CRR_conn**: CRR Connection object.
- **ECM_conn**: ECM Connection object.
- **AIF_conn:** AIF Connection object.

Format: `cx_Oracle.connect(<db_user/db_password@tns>)`

On successful execution of the paragraph, the details will be loaded in the `aif_case_information` table.

NOTE

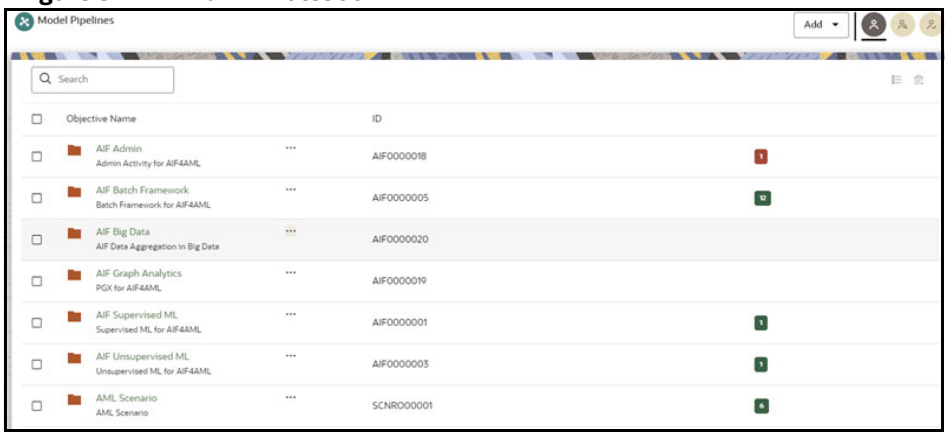
- Register Oracle wallet entries/aliases for CRR and ECM Atomic schema to connect within Compliance Studio.
- Use the aliases mentioned here to create/register entries. If aliases are being created with some other name, use them accordingly in the Admin Notebook.

7.14.2 Obtain SAR information for Production

To get Investigated Labels in Production, perform the following:

1. Login to Compliance Studio.
2. Launch the Sandbox workspace using the **launch** button.
3. On **Modeling** menu, click **Pipelines**.
4. Select **AIF Admin** Folder from the **Model Pipelines** summary page.

Figure 87: AIF Admin notebook



5. Open the Notebook with the **Pipeline Designer** option and switch to **Notebook** Tab.

Figure 88: Open Notebook in Pipeline Designer

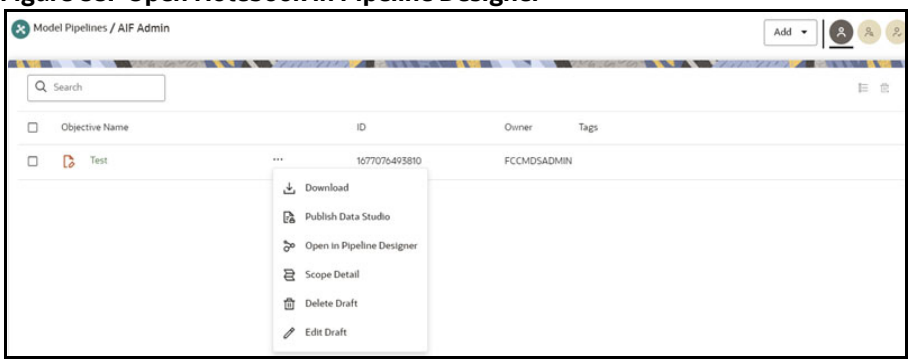
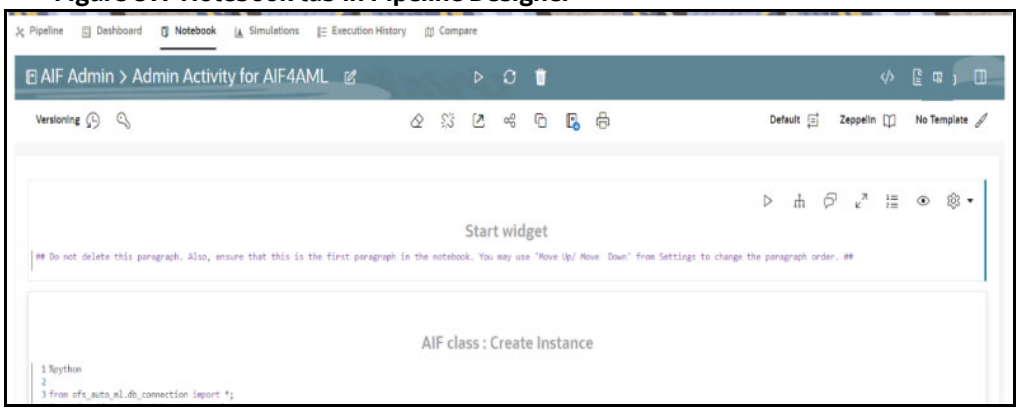


Figure 89: Notebook tab in Pipeline Designer



- Admin notebook facilitates the following functionalities to build Machine Learning Models:
  - Manage Model Groups
  - Import Model Templates
  - Obtain Investigated Labels
  - Configure Investigation Guidance
- As mentioned above, Notebook has paragraphs for **Obtaining Investigated Labels** from Enterprise Case Management (**ECM**) and Compliance Regulatory Reporting (**CRR**) or **CSV** file.

Figure 90: Obtaining Investigated Labels from CRR-ECM

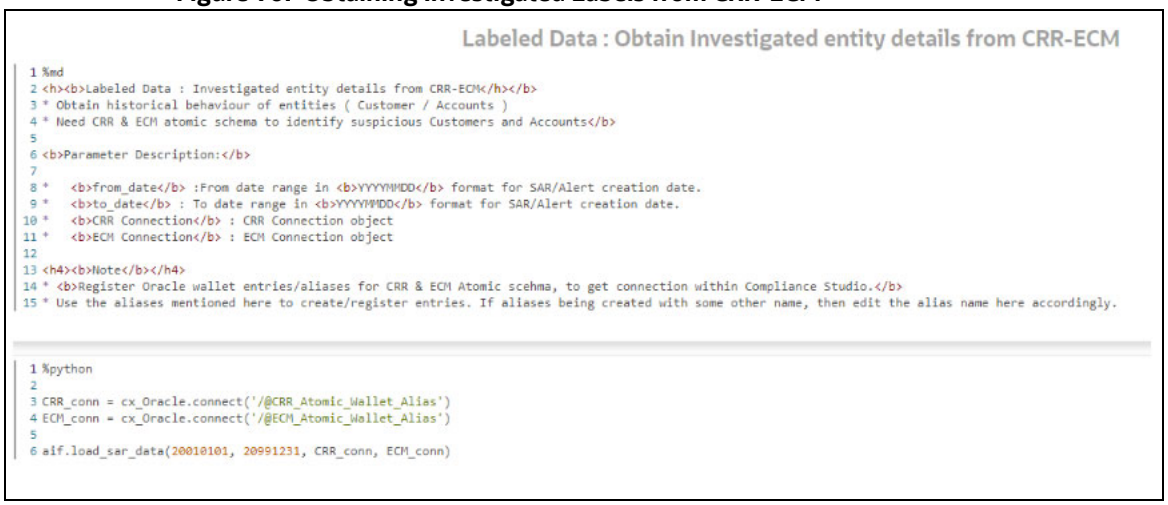


Figure 91: Obtaining Investigated Labels from CSV file

Users can select the above options to get the investigated labels into the workspace. See the Initialize the Session section for the usage of the paragraphs and Interactively executing the paragraphs in the sandbox workspace gets the labels in the Sandbox workspace.

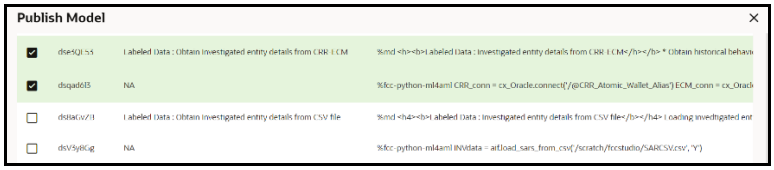
7.14.2.1 Obtain Labels in Production Workspace

To obtain labels in the production workspace, paragraphs must be deployed to Production and executed via Batch.

Perform the following:

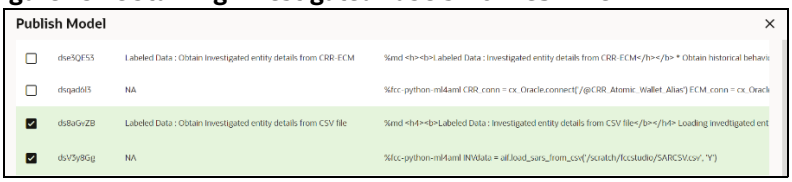
1. Obtaining labels for the following:
   - From **CRR-ECM, Publish and Deploy** the following two paragraphs:

Figure 92: Obtaining Investigated Labels from CRR-ECM



- From the **CSV file, Publish and Deploy** following two paragraphs:

Figure 93: Obtaining Investigated Labels from CSV file



For more details on Publish and Deploy, see the **How to Deploy the Model section in** OFS Compliance Studio Use Case Guide (ML4AML)**.**

2. Post successful deployment, create a New Batch and Execute the Batch to obtain investigated labels into the production workspace.

   Use the following task parameters while creating a new batch task:

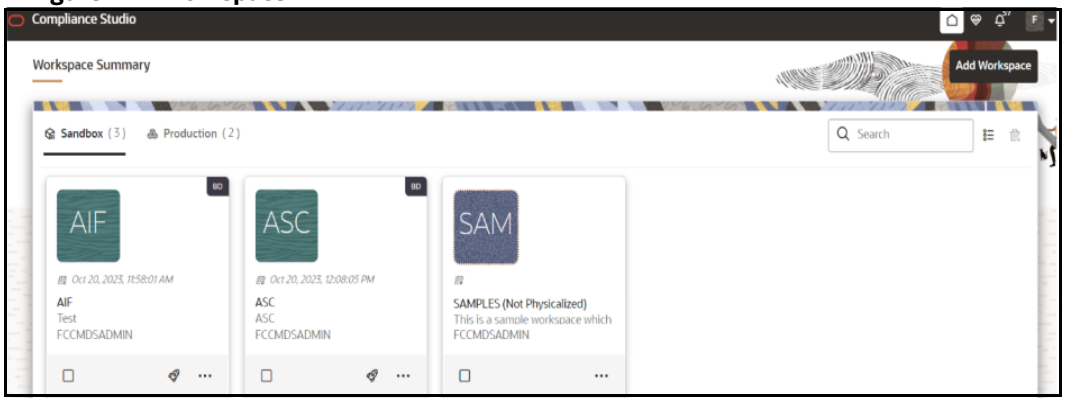
   - Objective: **AIF Admin**
   - Model: **CHAMPION**
   - Link Types: **Training + Scoring**
   - Synchronous Execution: **Yes**
   - Optional Parameters: **You can retain as-is/Leave it blank**

   For more information, see **Using Schedule Service** section in OFS Compliance Studio User Guide.

7.14.2.2 Create a New Batch for Obtaining Investigated Entities

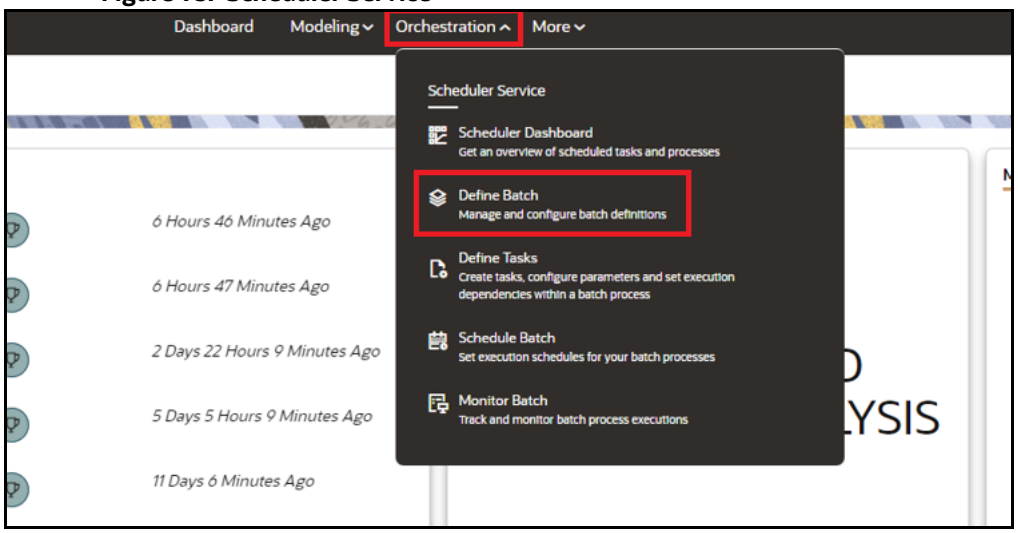
1. Launch **BD Production** workspace from the workspace summary screen.

Figure 94: Workspace



2. On **Orchestration** menu, click **Define Batch**.

Figure 95: Scheduler Service

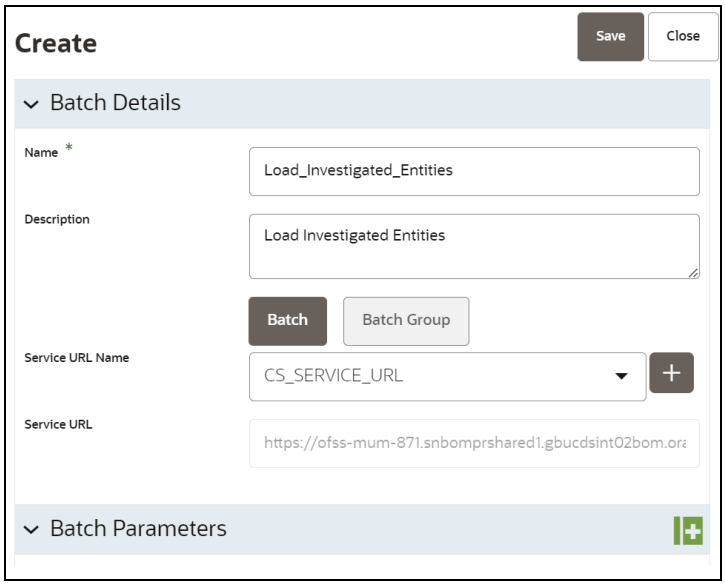


3. Click **Create** button on the top-right corner. The Create window is displayed.

Figure 96: Define Batch

4. Enter the **Name**, **Description,** and **Service URL** specified in the following figure.

Figure 97: Create Batch

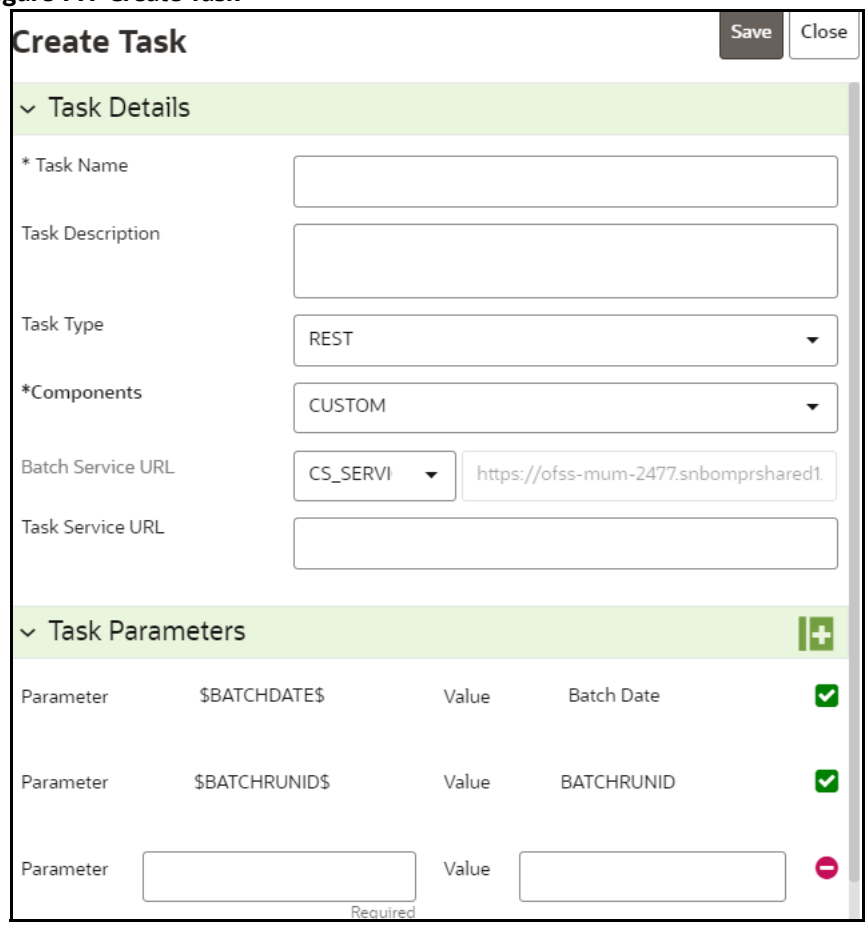


5. Click **Save** to **create a new batch**.
6. Navigate to **Scheduler Services** on the LHS pane and Click **Define Tasks** to create **New Task** in the newly created Batch.

Figure 98: Define Task

7. Select the **Batch** from the drop-down to create new tasks. Click **Add** to add tasks. The Create Task window is displayed.
8. Enter the following details to add task details and Parameters.

Figure 99: Create Task



9. Click **Save**. The task is created for the batch.

7.14.3 Obtain the SAR Information

7.14.3.1 Populate Investigated Entity Details

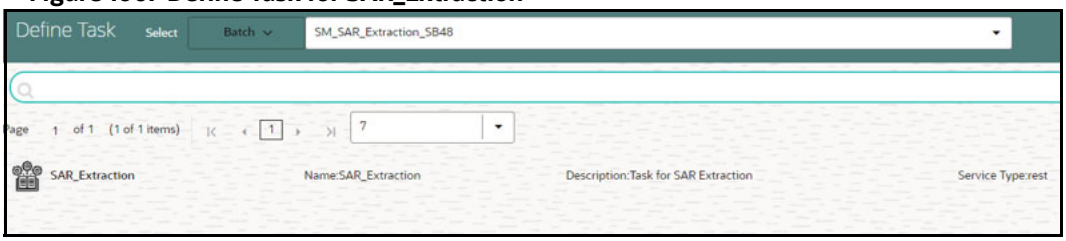
SM_SAR_Extraction batch is available in the out-of-the-box for the Scenario Model framework. This is a pre-seeded batch and will be available in all the workspaces.

This batch loads SAR Information to AIF_INVESTIGATED_ENTITY table.

7.14.3.1.1 Batch and Task Parameters

The batch contains a single task named **SAR_Extraction**.

Figure 100: Define Task for SAR_Extraction



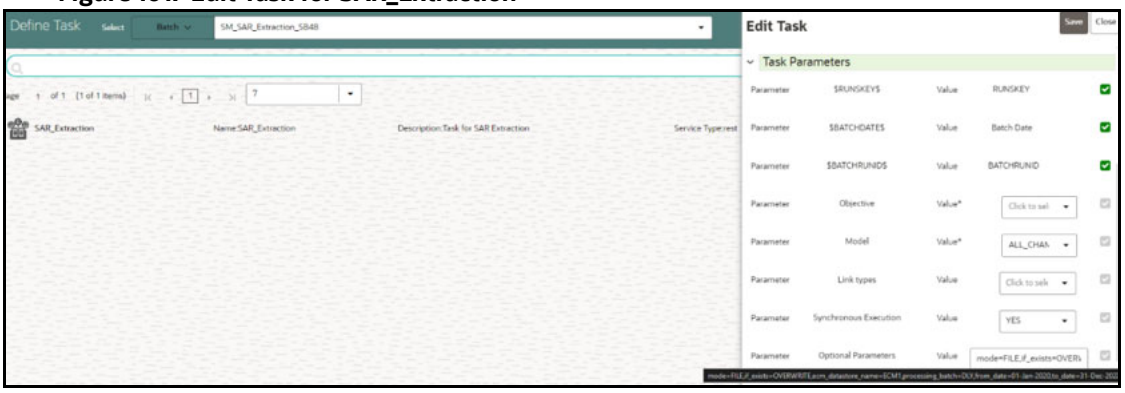
7.14.3.1.1.1 Task: SAR_Extraction, Task Parameters

- Objective folder for this task:

  `Home / Model Pipelines / ML4AML / Scenario Model / Batch / SAR Extraction`

- Do not change any parameter, except **Optional Parameters.**
- Optional Parameters:
  - **mode**: Extraction Mode to be used. This parameter is case-sensitive, and the option is either FILE or ECM.
  - **if_exists**: This parameter is used to set the behavior of data insertion. This parameter is case-sensitive, and the option is either OVERWRITE or APPEND.
    - **OVERWRITE**: Overwrites the rows where ENTITY_ID, ALERT_DATE, and LABELLED_SCENARIO are matched and inserts the rest of the rows.
    - **APPEND**: Ignores the rows where ENTITY_ID, ALERT_DATE, and LABELLED_SCENARIO are matched and inserts the rest of the rows.
  - **ecm_datastore_name**: Data Store created in the Compliance Studio UI for ECM atomic schema from where we need to extract the investigated labels.
  - **processing_batch**: Value for v_data_origin column from the fcc_events table in ECM.
  - **from_date**: Value for d_mis_date from the fcc_events table in ECM. The format should be DD-Mon-YYYY.
  - **to_date**: Value for d_mis_date from fcc_events table in ECM. The format should be DD-Mon-YYYY.
- **Example**: mode=**ECM**,if_exists=**OVERWRITE**,ecm_datastore_name=**SM_ECM**, processing_batch=**DLY**,from_date=**01-Nov-2015**,to_date=**30-Dec-2015**
- Edit **Task Parameters** & **Save**.

Figure 101: Edit Task for SAR_Extraction



7.14.3.1.2 Obtain the SAR from the CSV file

For loading data using a CSV file, the **SM_SAR_Extraction** batch should be executed using the following parameters:

mode = FILE, if_exists = OVERWRITE or **APPEND.**

> **NOTE** The remaining parameters can be ignored but should not be removed while running the batches.

A sample CSV is shipped with Compliance Studio named `sar.csv` in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ml4aml/demodata/sar.csv` directory.

This sample CSV is shipped with headers that resemble the structure of the **AIF_INVESTIGATED_ENTITY** table and two sample rows showing the format of each column.

Figure 102: Snapshot of sar.csv

When running the **SM_SAR_Extraction** batch with **mode = FILE**, the user should ensure that the following columns are available with the required values in the CSV files:

- **ENTITY_ID**: Customer ID or Account ID.
- **SUSPICIOUS_FLAG**: This flag has two options and they are **1 (Suspicious)** and **0 (Non-suspicious)**.
- **ALERT_DATE**: SAR/EVENT generation date. The format should be **YYYY-MM-DD**.
- **CREATED_ON**: CSV file creation date. The format should be **YYYY-MM-DD**.
- **CREATED_BY**: CSV file created by
- **UPDATED_ON**: CSV file updated date. The format should be **YYYY-MM-DD**.
- **UPDATED_BY**: CSV file updated by
- **LABELLED_SCENARIO**: Scenario ID corresponding to the entity_id and alert_date.
- **ENTITY_CD**: This parameter has the following options:

- CUSTOMER
- ACCOUNT
- EXTERNAL ENTITY
- CLIENT_BANK

The batch will read this file from its default location and load data to **AIF_INVESTIGATED_ENTITY** based on the **if_exists** condition**.**

> **NOTE** In the CSV file, the user is expected to populate Non-Null data for all the columns except UPDATED_ON and UPDATED_BY.

7.14.3.1.3 Obtain the SAR from ECM

For loading data from ECM, the **SM_SAR_Extraction** batch should be executed using **mode = ECM** along with all the other parameters.

For example,

mode=**ECM**, if_exists=**OVERWRITE**, ecm_datastore_name=**SM_ECM**, processing_batch=**DLY**, from_date=**01-Nov-2015** to_date=**30-Dec-2015**

The SM_SAR_Extraction batch runs with **mode = ECM**, will fetch data from ECM tables and load data to **AIF_INVESTIGATED_ENTITY** based on the **if_exists** condition.

The query used for fetching the data from ECM can be found in the **proc_ecm_sar_query** procedure under the **pkg_scenario_model** package.

The query expects the following ECM tables to have data:

- FCC_EVENTS
- FCC_EVENT_ENTITY_MAP
- FCC_EVENT_DETAILS
- FCC_SCENARIO_MASTER
- FCC_EVENT_INVESTIGATION_STATUS
- FCC_EVENT_STATUS_B
- KDD_CASE_LINKS
- KDD_CASES
- KDD_REVIEW_OWNER
- KDD_STATUS

7.15 Batch Framework

Batch Schedulers are available for the following use cases for ML4AML:

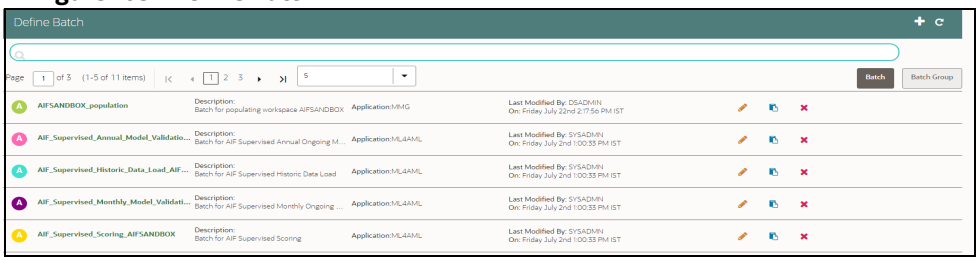
1. Supervised ML Batch Framework
2. Unsupervised ML Batch Framework
3. AMLES Batch Framework
4. Typology Scenario Batch Framework
5. Scenario Model Batch Framework

7.15.1 Supervised ML Batch Framework

Following Batches are available out of the box for the Supervised ML framework:

1. Supervised Historic Data Load
2. AIF Supervised Scoring
3. AIF Supervised Annual Model Validation
4. AIF Supervised Monthly Model Validation

Figure 103: Define Batch



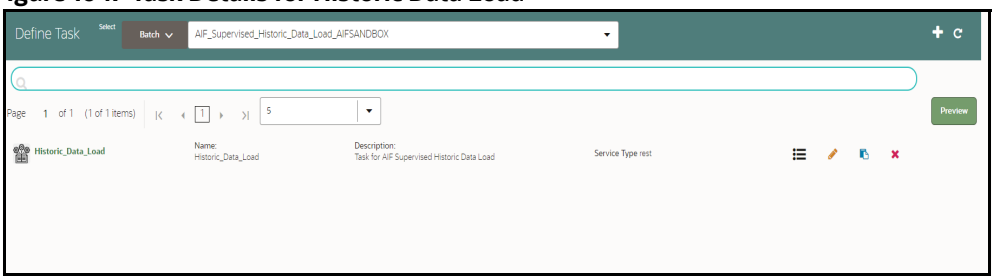
7.15.1.1 Supervised Historic Data Load

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the Sandbox workspace.
3. This Batch creates Historical Data Aggregates for ML Model training in the sandbox.

7.15.1.1.1 Batch and Task Parameters

The batch contains a single task named **Historic_Data_Load.**

Figure 104: Task Details for Historic Data Load



7.15.1.1.1.1 Task: Historic_Data_Load, Task Parameters

- Objective folder for this task:

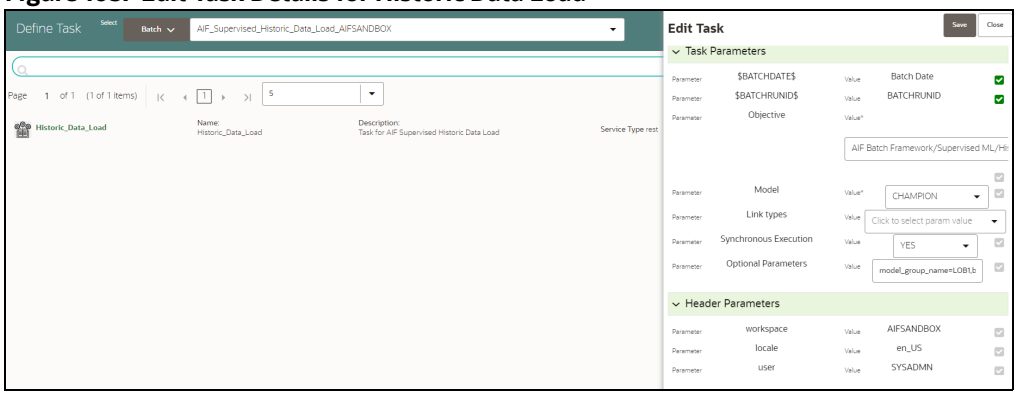
  `Home / Modeling / Pipelines / AIF Batch Framework / Supervised ML / Historical Data`

- Do not change any parameter, except **Optional Parameters.**
- Optional Parameters:
  - **model_group_name**: Name of the Model Groups for which Data Aggregation is to be created. Example **LOB1**
  - **benford_flag**: Flag indicates whether **Benford Law** Computation is required or not. Options **Y** or **N.**
  - **benford_digit**: Parameter to Benford law, **Benford Digit**. Options **1** or **2** or **3**

- **from_date**: Start date for Historic Data lookup in **DD-Mon-YYYY format.**
- **to_date**: End Date for Historic Data lookup in **DD-Mon-YYYY format.**

- **Example** : model_group_name=**LOB1**,benford_flag=**Y**,benford_digit=**1**,from_date=**01-Jul-2020**,to_date=**31-Jul-2021**
- **Edit** Task Parameters & **Save**.

Figure 105: Edit Task Details for Historic Data Load



7.15.1.2 Supervised Scoring

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the Production workspace.

7.15.1.2.1 Batch and Task Parameters

The batch contains the following tasks:

- Task 1: Scoring_Data_Load
- Task 2: ML_Scoring
- Task 3: ECM_Event_Processing

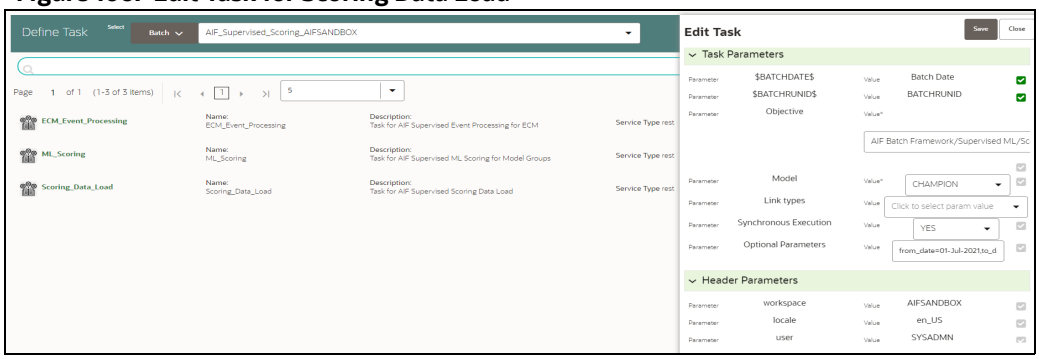
7.15.1.2.1.1 Task 1: Scoring_Data_Load, Task Parameters

- Objective folder for this task:

  `Home / Modeling / Pipelines / AIF Batch Framework / Supervised ML / Scoring Data`

- Optional Parameters:
  - **from_date**: Start date for Scoring Data lookup in **DD-Mon-YYYY format**.
  - **to_date**: End Date for Scoring/New Data lookup in **DD-Mon-YYYY format**.
- **Example**: from_date=**01-Jul-2020**,to_date=**31-Jul-2021**
- Optional Parameters can be edited from the **Schedule Batch** option.
- Change any other batch /task parameters, except **Optional Parameters.**

Figure 106: Edit Task for Scoring Data Load



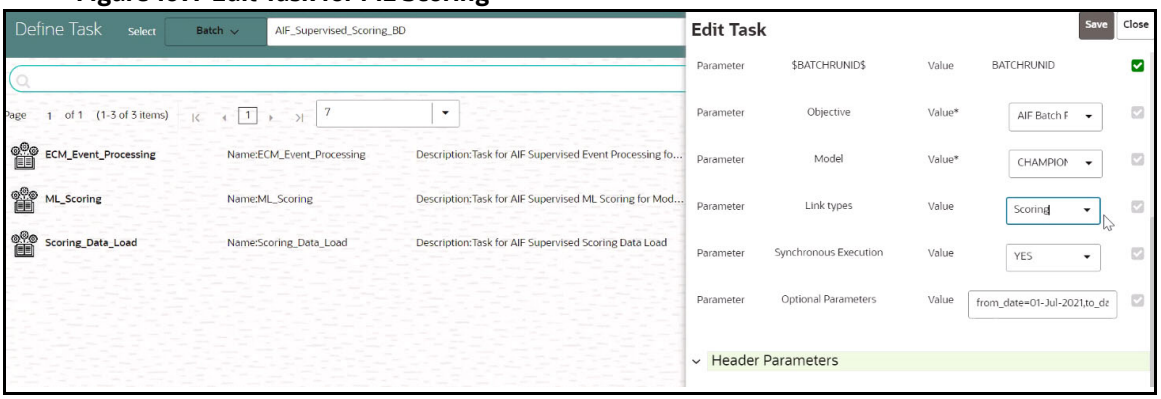
7.15.1.2.1.2 Task 2: ML Scoring, Task Parameters

- Objective folder for this task:

  `Home / Modeling / Pipelines / AIF Supervised ML / AIF`

  - Navigate to respective model group/scenario folders for actual model templates.
  - Optional Parameters:
    - **osot_end_month:** Specify the scoring data month in **YYYYMM format**. If not specified by default latest month data available in the table will be picked up for scoring.
    - **threshold:** Input threshold or cutoff to create events. Events will be created if the score of an entity exceeds the threshold. **Example: 0.7**
    - **from_date**: Start date for Scoring Data lookup in **YYYYMM format.**
    - **to_date**: End Date for Scoring/New Data lookup in **YYYYMM format. Example** : from_date=**202007**,to_date=**202007**
- Optional Parameters can be edited from the **Schedule Batch** option.
- Choose **Link Types** as **Scoring**.
- Do not change any batch/task parameters, except **Optional Parameters.**

Figure 107: Edit Task for ML Scoring



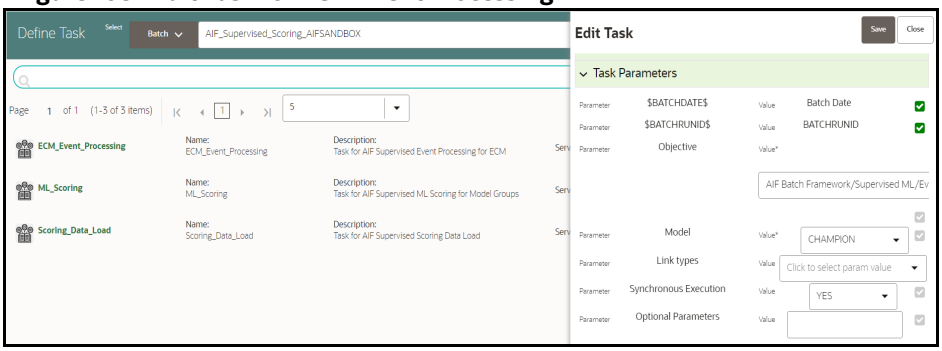
7.15.1.2.1.3 Task 3: ECM_Event_Processing, Task Parameters

- Objective folder for this task:

  `Home / Modeling / Pipelines / AIF Batch Framework / Supervised ML / Event Processing`

- This task does not take any optional parameters.
- Do not change any other batch/task parameters.

Figure 108: Edit Task for ECM Event Processing



- After scoring for supervised customer risk scoring, the outputs are stored in the AIF_ENTITY_SCORE table.
- Alerts generated above thresholds are moved to the following tables for case management integration:
  - FCC_AM_EVENTS
  - FCC_AM_EVENT_DETAILS
  - FCC_AM_EVENT_ENTITY_MAP
  - FCC_AM_EVENT_BINDING

7.15.1.2.2 Cleanup Steps in case of Running the Scoring Process Twice

In case the user wants to run the Scoring Process for the same FIC_MIS_DATE and same MODEL_GROUP_NAME twice, the following cleanup steps should be performed first:

1. Remove the existing events:

```
delete from fcc_am_event_binding where v_event_cd in (select v_event_cd
from fcc_am_events where prcsng_dt='DD-Mon-YYYY');

delete from fcc_am_event_entity_map where v_event_cd in (select
v_event_cd from fcc_am_events where prcsng_dt='DD-Mon-YYYY');

delete from fcc_am_event_details where n_event_cd in (select v_event_cd
from fcc_am_events where prcsng_dt='DD-Mon-YYYY');

delete from fcc_am_events where prcsng_dt='DD-Mon-YYYY');
```

2. Get the child tables which contain scoring results:

```
select D_FIC_MIS_DATE, V_MODEL_GROUP, V_OUTPUT_TABLE_NAME,
V_OUTPUT_TABLE_NAME_ALL_ENTITY

from aif_entity_score

where d_fic_mis_date ='DD-Mon-YYYY'

and model_group_name='<Model_Group_Name>';
```

3. Drop all child tables manually listed in V_OUTPUT_TABLE_NAME and V_OUTPUT_TABLE_NAME_ALL_ENTITY columns from the result of the above query :

```
drop <Child_Table_Name>;
```

4. Delete the parent entry from aif_entity_score:

```
delete from aif_entity_score where d_fic_mis_date='DD-Mon-YYYY'
```

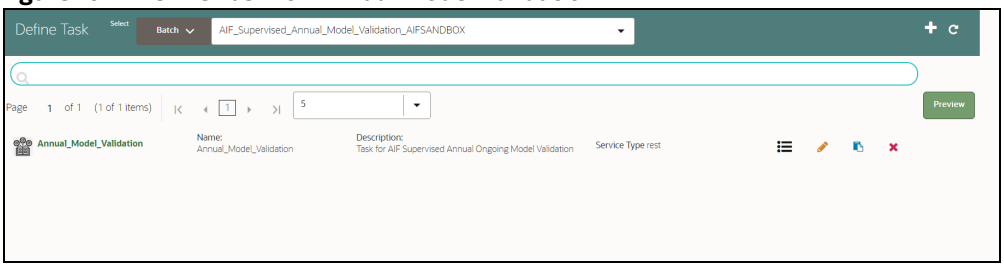
7.15.1.3 Annual Model Validation

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the **Production** workspace.
3. This Batch shows ongoing model performance annually.

7.15.1.3.1 Batch and Task Parameters

The batch contains a single task named Annual_Model_Validation

Figure 109: Define Task for Annual Model Validation



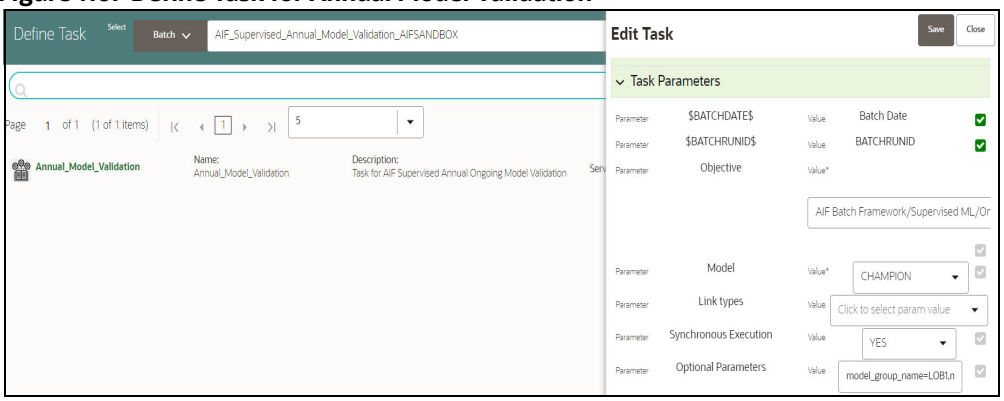
7.15.1.3.1.1 Task: Annual_Model_Validation, Task Parameters

- Objective folder for Data Quality:

```
Home / Modeling / Pipelines / AIF Batch Framework / Supervised ML /
 Ongoing Model Validation / Annual
```

- Do not change any parameter, except **Optional Parameters.**
- Optional Parameters:
  - **model_group_name**: Name of the Model Groups for which Model has been trained. Example **LOB1**
  - **model_group_scenario_name:** Name of the Model Groups Scenario for which Model has been trained. Example **Cash**
  - **osot_end_month**: Specify the data month in **YYYYMM format**. If not specified by default latest month data available in the table will be picked up for monthly validations as scoring data / new data.
- **Example**: model_group_name=**LOB1**,model_group_scenario_name=**None**,osot_end_month=**None**
- Optional Parameters can be edited from the **Schedule Batch** option.
- Do not change any batch/task parameters, except **Optional Parameters**.

Figure 110: Define Task for Annual Model Validation



7.15.1.4 Monthly Model Validation

1. This pre-seeded batch will be available in all workspaces (production & sandboxes).
2. This Batch is to be executed in the **Production** workspace.
3. This Batch shows ongoing model drift and data quality with respect to new data every month (monthly).

7.15.1.4.1 Batch and Task Parameters

The batch contains a single task named Monthly_Model_Validation.

7.15.1.4.1.1 Task: Monthly_Model_Validation, Task Parameters

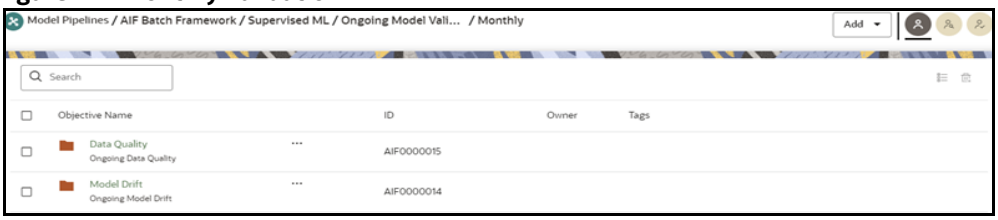
- Objective folder for Data Quality :

  ```
  Home / Modeling / Pipelines / AIF Batch Framework / Supervised ML /
  Ongoing Model Validation / Monthly / Data Quality
  ```

- Objective folder for Model Drift :

  ```
  Home / Modeling / Pipelines / AIF Batch Framework / Supervised ML /
   Ongoing Model Validation / Monthly / Model Drift
  ```

Figure 111: Monthly Validation



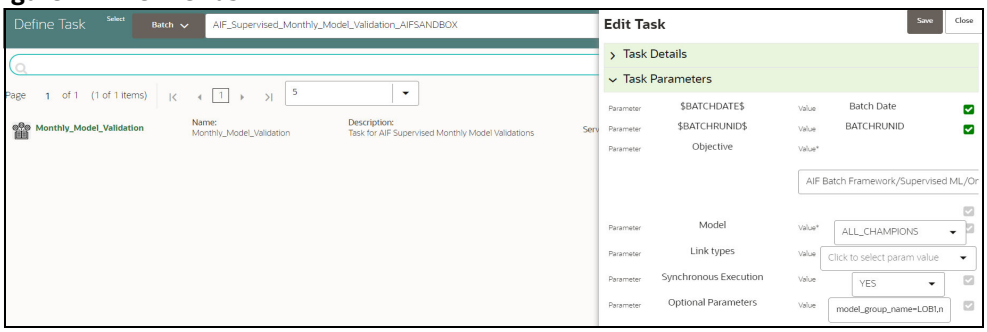
- Do not change any parameter, except **Optional Parameters.**
- Optional Parameters:
  - **model_group_name**: Name of the Model Groups for which Model has been trained. Example **LOB1**
  - **model_group_scenario_name:** Name of the Model Groups Scenario for which Model has been trained. Example **Cash**
  - **osot_end_month**: Specify the data month in **YYYYMM format**. If not specified by default latest month data available in the table will be picked up for monthly validations as scoring data / new data.

- **FEATURE_INCLUDE:** List of features to be included for **data quality**. Default **None** means everything.
- **FEATURE_EXCLUDE:** List of features to be excluded for **data quality**. Default **None** means exclude nothing.
  - When both include & exclude is provided. Include takes precedence over exclude.
  - **Example 1** : feature_include="Feature1~Feature2"
  - **Example 2** : feature_exclude="Feature3~Feature4~Feature5"
- **look_back_months:** No of periods to look back for getting drift history. Default is 5**.**
- **Number_Of_Bins:** Number of bins to be used in discretizing (scalar). **Default is 9.**
- **Boot_Strap_Samples:** Number of bootstrap samples on which to estimate thresholds. **Default is 5.**
- **Standard_Deviation_Band_Sigma:** Number of standard deviation band (sigma band). Threshold setting to be used. **Default is 2 sigma.**

Example: model_group_name=LOB1,model_group_scenario_name=None,osot_end_month=None,Number_Of_Bins=9,Boot_Strap_Samples=5,Standard_Deviation_Band_Sigma=2,look_back_months=5,FEATURE_INCLUDE=None,FEATURE_EXCLUDE=None

- Optional Parameters can be edited from the **Schedule Batch** option.
- Do not change any batch/task parameters, except **Optional Parameters.**

Figure 112: Define Task

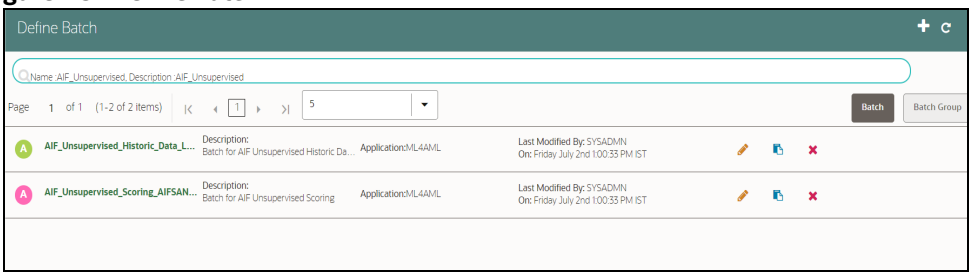


7.15.2 Unsupervised ML Batch Framework

The following batches are available out of the box:

1. Unsupervised Historic Data Load
2. Unsupervised Scoring

Figure 113: Define Batch



7.15.2.1 Unsupervised Historic Data Load

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the **Sandbox** workspace.

The historic data batch fetches 12 months or more of transactional data for all entities and will be aggregated for each entity. These aggregated features are used to populate the tables in the following with just one row for each entity.

The following tables that this batch will populate.

- AIF_BEHAVIORAL_DATA_UNSUP
- AIF_NON_BEHAVIORAL_DATA

These tables will be used for customer segmentation.

This batch has only one task defined under it:

- Historic_Data_Load

Figure 114: Define Task

7.15.2.1.1 Historic_Data_Load

- The objective folder for this task is

  `Home/Modeling/Pipelines/AIF Batch Framework/Unsupervised ML/Historical Data.`
- Do not change the parameters **Objective**, **Model**, **Link types**, and **Synchronous Execution**.
- The values in "Optional Parameters" can be edited:
  - **model_group_name**: Name of the model group the batch has to be run for as created in the admin notebook.
  - **model_group_scenario_name**: Name of the model group scenario under this model group for which the batch has to be run.
  - **from_date**: From date in DD-MON-YYYY format. Example: 01-Jul-2021
  - **to_date**: To date in DD-MON-YYYY format. Example: 31-Jul-2021
- **Example**: model_group_name=MODEL_GROUP_X,model_group_scenario_name=None,from_date =01-Jan-2020,to_date=31-Jan-2021

7.15.2.2 Unsupervised Scoring

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the **Production** workspace.

The scoring data batch is used to fetch one month or more of transactional data for previously segmented customers and also 12 months or more of transactional data for new entities who are now eligible for segmentation.

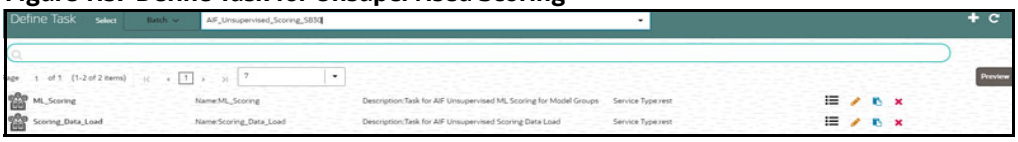
The following tables that this batch will populate.

- **AIF_BEHAVIORAL_DATA_UNSUP_PROD**

- **AIF_NON_BEHAVIORAL_DATA_PROD**

> **NOTE**
>
> 1. This batch has 2 tasks defined under it:
>   - **Scoring_Data_Load**
>   - **ML_Scoring**
> 2. In Sandbox, Cluster Information will be stored in the **AIF_ENTITY_CLUSTER** table.

Figure 115: Define Task for Unsupervised Scoring



Data for new entities is populated into these tables:

- AIF_BEHAVIORAL_DATA_UNSUP
- AIF_NON_BEHAVIORAL_DATA

7.15.2.2.1 Scoring_Data_Load

- The objective folder for this task is:

  `Home/Modeling/Pipelines/AIF Batch Framework/Unsupervised ML/Scoring Data.`

- Do not change the parameters **Objective**, **Model**, **Link types**, and **Synchronous Execution**.
- The values in "Optional Parameters" can be edited:
  - **from_date**: From date in DD-MON-YYYY format. Example: 01-Jul-2021
  - **to_date**: To date in DD-MON-YYYY format. Example: 31-Jul-2021
- **Example**: from_date=01-Jan-2021,to_date=31-Jan-2021

7.15.2.2.2 ML_Scoring

- The objective folder for this task is **Home/Model Pipelines/AIF Unsupervised ML/AIF**.
- Do not change the parameters **Objective**, **Model**, **Link types**, and **Synchronous Execution**.

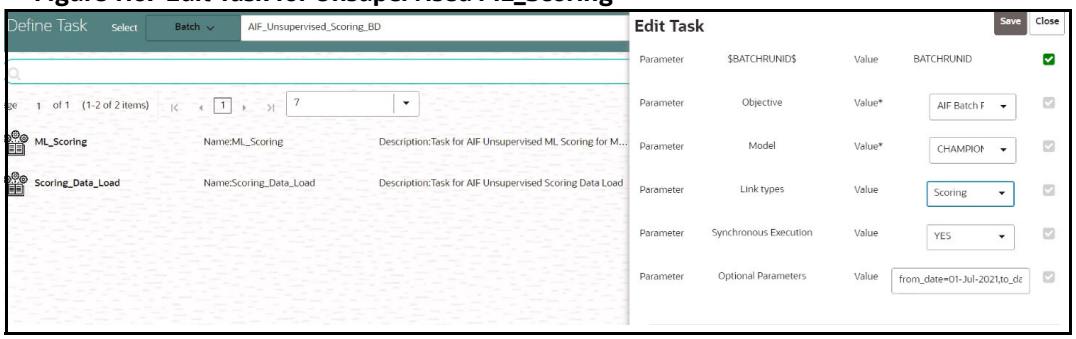
The values in "Optional Parameters" can be edited:

- **osot_end_month_anomaly_scoring**: Specify the scoring data month in **YYYYMM format**. If it is not specified, then by default the latest month data available in the table will be picked up for anomaly scoring.
- **debug**: Assign **True** if debug mode is to be switched on. Default is **False**.
- **data_start_date**: Start date for Scoring Data lookup in **YYYYMM format.**
- **data_end_date**: End Date for Scoring/New Data lookup in **YYYYMM format.**
- **method_anomaly_scoring**: String indicating which anomaly scoring method to use. Currently **"NNLOF", "PCAREC"** and **"ISOFOR"** are supported and the default is **"NNLOF"**.
- **cutoff_pctl_anomaly_scoring**: Cutoff percentile for anomaly flags. Ranges from **0 to 100**. Defaults to **99**.

- **osot_end_month_deviation_scoring**: Specify the scoring data month in **YYYYMM** format. If it is not specified, then by default the latest month data available in the table will be picked up for deviation scoring.
- **cutoff_pctl_deviation_scoring**: Cutoff percentile for deviation scoring. Ranges from **0 to 100**. Defaults to **99**.
- **method_deviation_scoring**: String indicating which deviation scoring method to use. Currently **"LDCOF"** and **"CBLOF"** are supported and the default is **"CBLOF"**.
- Choose **Link Types** as **Scoring**.

  Example:
  ```
  osot_end_month_anomaly_scoring=None,debug=False,data_start_date=202207,data_end_date=202207,method_anomaly_scoring=NNLOF,cutoff_pctl_anomaly_scoring=99,osot_end_month_deviation_scoring=None,cutoff_pctl_deviation_scoring=99,method_deviation_scoring=LDCOF
  ```

Figure 116: Edit Task for Unsupervised ML_Scoring



After scoring for unsupervised, data is stored in the following tables:

- AIF_ANOMALY_SCORE
- AIF_ANOMALY_SCORE_DETAILS
- AIF_ANOMALY_SCORE_ECM_DETAILS
- AIF_ENTITY_CLUSTER_DEVIATION

Table 42 describes AIF_ANOMALY_SCORE.

Table 42: AIF_ANOMALY_SCORE

COLUMN NAME	COLUMN TYPE	DESCRIPTION
D_FIC_MIS_DATE	DATE	Batch execution date
D_DATA_START_DATE	DATE	Scoring month in YYYYMM format
D_DATA_END_DATE	DATE	Scoring month in YYYYMM format
V_MODEL_GROUP	VARCHAR2(100 CHAR)	Name of the model group / segment
N_MODEL_GROUP_ID	NUMBER	Model group ID / Segment ID
V_DEFINITION_ID	VARCHAR2(100 CHAR)	Customer segmentation and anomaly detection model definition ID
V_TECHNIQUE	VARCHAR2(500 CHAR)	Deployed model technique
V_STATUS	VARCHAR2(50 CHAR)	Status of scoring like SUCCESS / FAILED

Table 42: AIF_ANOMALY_SCORE

COLUMN NAME	COLUMN TYPE	DESCRIPTION
V_UN_PREDICTED_ID	CLOB	Unpredicted entities during scoring due to various reasons
V_LOG	CLOB	Execution log
V_MODEL_GROUP_SCENARIO	VARCHAR2(100 CHAR)	Name of the model group scenario / sub segment
V_MODEL_GROUP_SCENARIO_ID	VARCHAR2(30 CHAR)	Model group scenario ID
V_TRAIN_NOTEBOOK_ID	VARCHAR2(30 CHAR)	Studio notebook ID used for model training
V_SCORE_NOTEBOOK_ID	VARCHAR2(30 CHAR)	Studio notebook ID used for model scoring
V_BATCH_RUN_ID	VARCHAR2(50 CHAR)	Unique batch execution ID

Table 43 describes AIF_ANOMALY_SCORE_DETAILS.

Table 43: AIF_ANOMALY_SCORE_DETAILS

COLUMN NAME	COLUMN TYPE	DESCRIPTION
ENTITY_ID	VARCHAR2(50 CHAR)	Entity ID for which anomaly is detected
AGGREGATION_END_DATE	NUMBER	Scoring month in YYYYMM format
MODEL_GROUP_NAME	VARCHAR2(4000 CHAR)	Name of the model group / segment
ANOMALY_SCORE	NUMBER	Anomaly score
PEERGROUP_ID	VARCHAR2(10)	Assigned Peer Group to the entity
CLUSTER_ID	VARCHAR2(5)	Assigned Cluster Id to the entity

Table 44 describes AIF_ANOMALY_SCORE_ECM_DETAILS.

Table 44: AIF_ANOMALY_SCORE_ECM_DETAILS

COLUMN NAME	COLUMN TYPE	DESCRIPTION
D_FIC_MIS_DATE	DATE	Batch execution date
V_MODEL_GROUP	VARCHAR2(100 CHAR)	Name of the model group / segment
ENTITY_ID	VARCHAR2(50 CHAR)	Entity ID for which anomaly is detected
PREDICTION	NUMBER	Anomaly score
PREDICTION_PERCENTILE	NUMBER	Anomaly score as percentile
TILE	VARCHAR2(10 CHAR)	Risk bucket like High, Medium, and Low
INPUT_FEATURE	CLOB	Input ML model features
FEATURE_CONTRIBUTION	CLOB	Individual ML feature contributions to form final score
CASE_INFORMATION	CLOB	Additional details for investigation in iHUB only

Table 44: AIF_ANOMALY_SCORE_ECM_DETAILS

COLUMN NAME	COLUMN TYPE	DESCRIPTION
ASSIGNED_PEER_GROUP	VARCHAR2(100)	Name of the Peer group assigned to entity

Table 45 describes AIF_ENTITY_CLUSTER_DEVIATION.

Table 45: AIF_ENTITY_CLUSTER_DEVIATION

COLUMN NAME	COLUMN TYPE	DESCRIPTION
ENTITY_ID	VARCHAR2(50 CHAR)	Entity ID for which anomaly is detected
AGGREGATION_END_DATE	NUMBER	Scoring month in YYYYMM format
MODEL_GROUP_NAME	VARCHAR2(4000 CHAR)	Name of the model group / segment
CLUSTER_ID	VARCHAR2(5)	Assigned Cluster Id to the entity
DEVIATION_SCORE	NUMBER	Deviation Score for entity
DEVIATION_FEATURE_CONTRIBUTION	CLOB	Individual ML feature contributions to form final score

The application can consume anomaly scores from the above tables for downstream integrations.

- **AIF_ANOMALY_SCORE_DETAILS** is a parent table with three entries (anomaly scoring, deviation scoring, and prediction of new entities) per scoring execution of the model.
- **AIF_ANOMALY_SCORE_ECM_DETAILS** is a child table that holds detailed outputs at the entity level.

7.15.2.2.2.1 Feature Contributions JSON Format

Feature contributions and expected values for anomalies are given under the column **CASE_INFORMATION** in the **AIF_ANOMALY_SCORE_ECM_DETAILS** table. Feature contributions provide a general idea of which features contributed how much to an anomaly's behavior. Expected values are a range of values given for every feature that indicates the feature's expected value based on the behavior of the entity's peer group. The **CASE_INFORMATION** column has JSONs as values, and their format should be as follows:

```
{
  "Feature Description": {
    "Feature": [
      "MAX_AVG_CR_AMT",
      "TOTAL_AVG_CR_AMT",
      "TOTAL_AVG_DB_CNT",
      "MAX_AVG_DB_CNT"
    ],
    "Contribution": [
      5.46667,
      5.06002,
```

```
    1.57681,
    1.42856
  ],
  "Impact on Risk": [
    "Increase",
    "Increase",
    "Increase",
    "Decrease"
  ],
  "Significance": [
    "Very High",
    "Very High",
    "Very High",
    "Very High"
  ],
  "Feature Description": [
    "Unusual incoming amounts through a single transaction channel",
    "Unusual amounts of deposits when compared to peers",
    "Unusual volume of withdrawals when compared to peers",
    "Feature Tag/Feature Description Not found"
  ],
  "Feature Tag": [
    "Maximum Average Credit Amount",
    "Total Average Credit Amount",
    "Total Average Debit Count",
    "Feature Tag/Feature Description Not found"
  ],
  "Expected_Feature_Values": [
    "16733.97 - 22725.54",
    "27105.79 - 37305.04",
    "12 - 14",
    "6 - 9"
  ],
  "Observed_Feature_Values": [
    146742.01,
    215230.35,
```

```
        38,
        18
      ]
    },
    "Investigation Guidance": {},
    "Investigation Summary": {}
}
```

NOTE

- If **PCAREC** is the method used during anomaly scoring, expected values will not be given as minimum and maximum values of a range.
- When using deviation scoring, the outputs are written to the table **AIF_ENTITY_CLUSTER_DEVIATION**.
- The anomaly percentile score of an entity is a relative metric with respect to other entities within the same peer group. An entity flagged as an anomaly based on its percentile score cutoff value but having its observed values within the ranges of expected values should be treated as non-anomalous. This could happen if there are actually no entities with anomalous activity within a peer group but still the algorithm has to identify at least a certain number of anomalies based on input parameters.

7.15.3 AMLES Batch Framework

Following Batches are available out of the box for the Supervised ML framework

1. **AMLES Historic Event Load**
2. **AMLES Scoring**
3. **AMLES Update Event Labels**

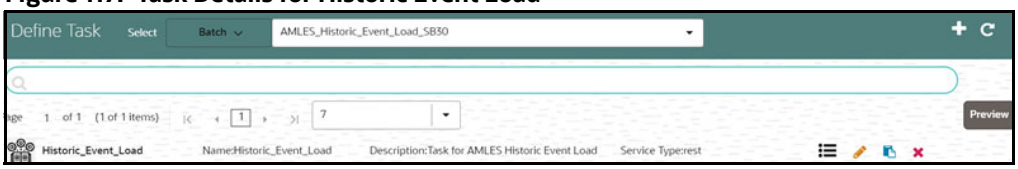
7.15.3.1 AMLES Historic Event Load

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the **Sandbox** workspace.
3. This Batch pulls data from the ECM system used for ML Model training in the sandbox.

7.15.3.1.1 Batch and Task Parameters

The batch contains a single task named **Historic_Event_Load.**

Figure 117: Task Details for Historic Event Load



7.15.3.1.1.1 Historic_Event_Load, Task Parameters

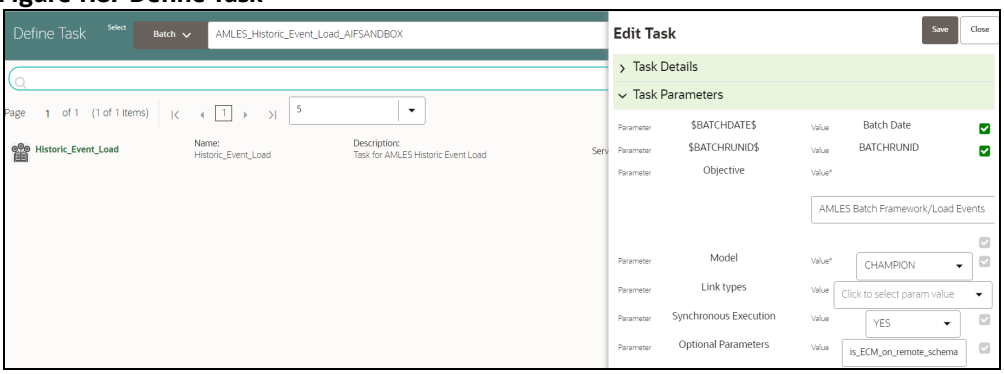
- Objective folder for this task :

  `Home / Modeling / Pipelines / AMLES Batch Framework / Load Events / AMLES Data Load`

- Do not change any parameter, except **Optional Parameters.**
- **Optional Parameters:**
  - **Event date range: from_date=YYYY-MM-DD,to_date=YYYY-MM-DD**
  - **is_ECM_on_remote_schema**: Flag indicates **ECM Schema** is on different schema or not. Options **True** or **False**
  - **enable_debug_mode**: enable debug mode or not. Options **True** or **False**

  Example : is_ECM_on_remote_schema=True,from_date=2001-01-01,to_date=2022-01-01
- **Edit** Task Parameters & **Save**.

Figure 118: Define Task



7.15.3.2 AMLES Scoring

1. This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
2. This Batch is to be executed in the **Production** workspace.

7.15.3.2.1 Execution Frequency

Scenario frequency gives the flexibility to schedule event-scoring solution at appropriate frequency so that daily, weekly and monthly events can easily be handled by event-scoring notebook.

As a solution, raw data which is input for event-scoring is pulled on daily basis. It consists of daily, weekly and monthly alerts.

Since alerts are pulled from ECM on daily basis, it is possible weekly and monthly alerts are not pulled daily. In this case, weekly and monthly event-scoring notebook exits gracefully and makes one entry in amles_event_score table with status as **No Data** and with the status as successful.

Output of AMLES event-scoring is stored in following static tables in BD schema.

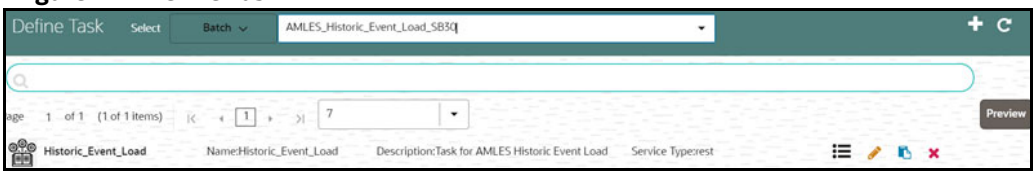
- amles_event_score
- amles_event_score_details

7.15.3.2.2 Batch and Task Parameters

The batch contains the following tasks:

- **Scoring_Event_Data_Load**
- **ML_Scoring**

Figure 119: Define Task



7.15.3.2.2.1 Scoring_Event_Data_Load, Task Parameters

- Objective folder for this task :

  `Home / Modeling / Pipelines / AMLES Batch Framework / Load Events / AMLES Data Load`
- Do not change any parameter, except **Optional Parameters.**
- Optional Parameters:
  - **is_ECM_on_remote_schema**: Flag indicates **ECM Schema** is on different schema or not. Options **True** or **False**
  - **enable_debug_mode**: enable debug mode or not. Options **True** or **False**

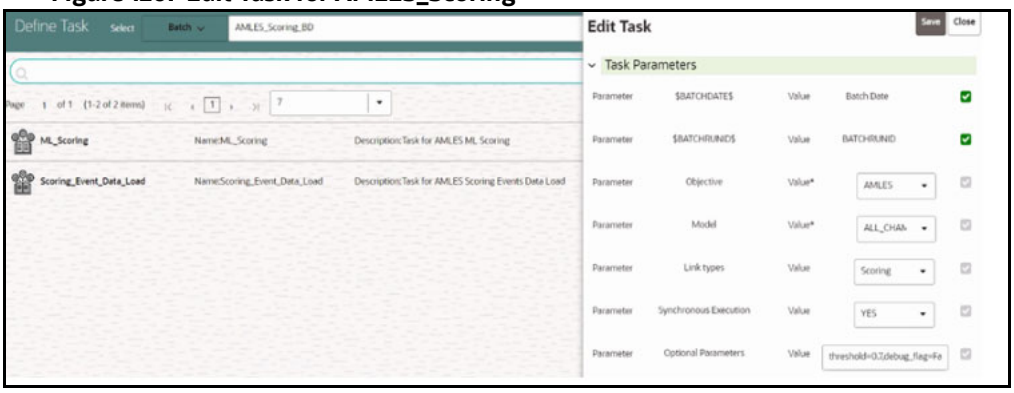
  Example : is_ECM_on_remote_schema=True
- Optional Parameters can be edited from **Schedule Batch** option.
- Do not change any other batch /task parameters, except **Optional Parameters.**

7.15.3.2.2.2 ML_Scoring, Task Parameters

- Objective folder for this task :

  `Home / Modeling / Pipelines / AMLES`
- Navigate to respective model group/scenario folders for actual model templates.
- Optional Parameters:
  - **threshold:** Input threshold or cutoff to create events. Events will be created if the score of an entity exceeds the threshold. **Example: 0.7**
  - **debug_flag**: flag to set for debugging purpose**.** Few records will be selected. **Options**: **True** or **False**
  - Optional Parameters can be edited from the **Schedule Batch** option.
  - Do not change any batch/task parameters, except **Optional Parameters.**
  - Choose **Link Types** as **Scoring**.

Figure 120: Edit Task for AMLES_Scoring

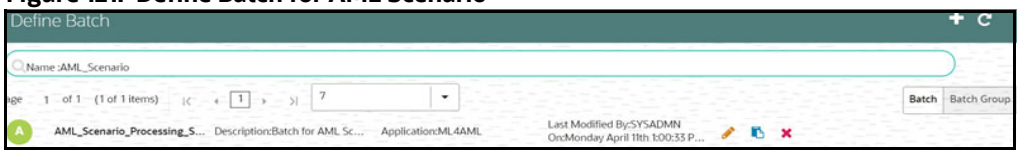


7.15.4 Typology Scenario Batch Framework

Following Batch available out of the box for the Typology scenario batch framework.

AML_Scenario_Processing

Figure 121: Define Batch for AML Scenario



7.15.4.1 AML Scenario Processing batch

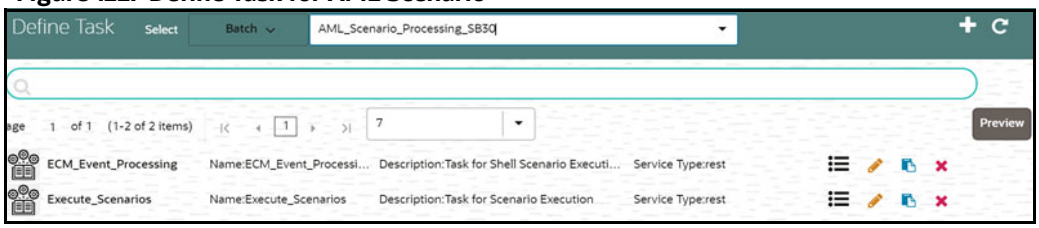
1. This is a pre-seeded batch and will be available in all workspaces (Production and Sandboxes).
2. This Batch can be executed in the Sandbox and Production workspaces.
3. This Batch executes scenario logic and generates events in **fcc_am*** tables.
4. Sandbox is mainly used for scenario tuning, and what-if analysis and main execution are done in Production.

7.15.4.1.1 Batch and Task Parameters

The Batch contains the following task named as:

1. Execute_Scenario
2. ECM_Event_Processing

Figure 122: Define Task for AML Scenario

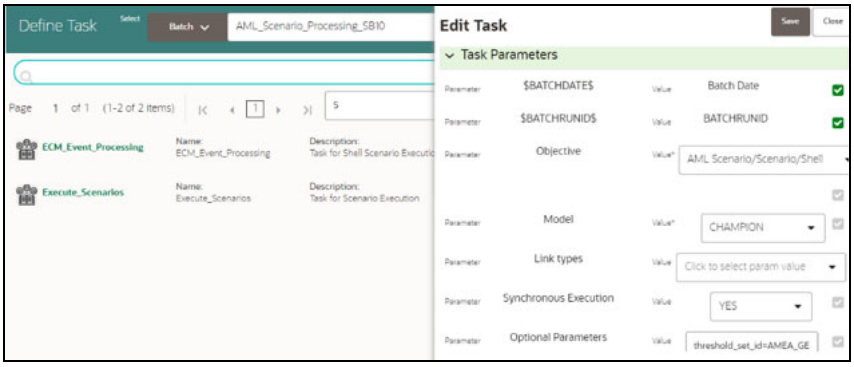


7.15.4.1.1.1 Task 1: Execute_Scenario, Task Parameters

- Objective folder for this task:
- `Home / Modeling / Pipelines / AML Scenario / Scenario / Shell / Customer`
  - The Shell or Human Trafficking folder needs to change based on execution requirements.
- The objective parameter and Optional parameter can be changed based on the requirement. No other parameter needs to change.
- Optional Parameters:
  - **threshold_set_id**: ID of the threshold set, Example AMEA_GENERAL.
  - **lookback**: Number of days to look back for data. Example 30

  Example: threshold_set_id=AMEA_GENERAL,lookback=30
- Edit Task Parameters and **Save**.

Figure 123: Define Task Parameter

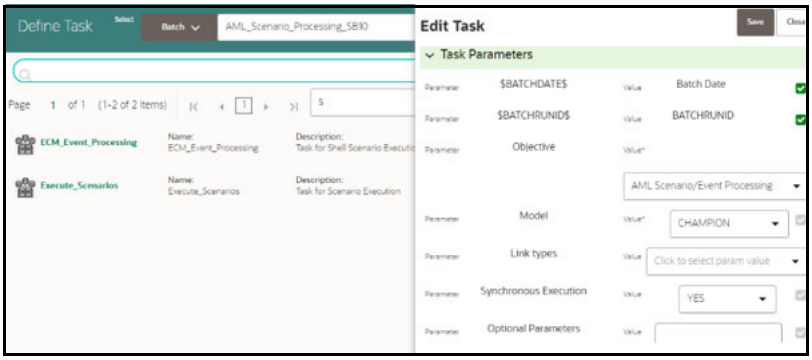


7.15.4.1.1.2 Task 2: ECM_Event_Processing, Task Parameters

- Objective folder for this task:
- `Home / Modeling / Pipelines / AML Scenario / Event Processing`

  This task does not take any optional parameters.
- Do not change any other batch/task parameters.

Figure 124: Edit Task Parameter



- AMLES event score outputs are available in the following tables:
  - AMLES_EVENT_SCORES

- AMLES_EVENT_SCORE_DETAILS

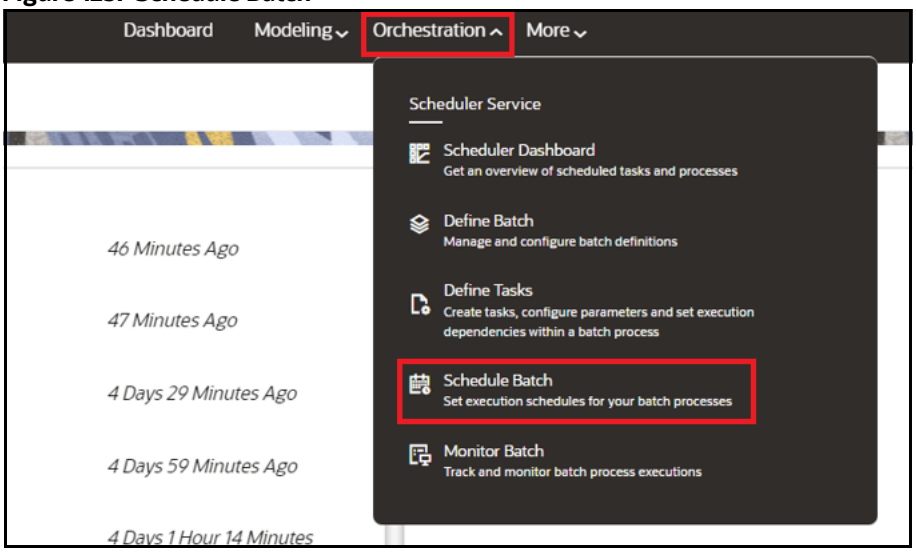
- Use the following schema for the table structure to insert into the document:

  TNS: ML4AMLPRODREST/password@ofss-mum-3629.snbomprshared1.gbucdsint02bom.oraclevcn.com:1521/fccmdb

7.15.5 Execute Batch

1. On **Orchestration** menu, click **Schedule Batch**.
2. Select the **Batch** from the drop-down.
3. Click **Edit Parameters** to select **MIS Date** and other parameters for the various tasks. **Save** changes.
4. Click **Execute** to Execute/Trigger the Batch.

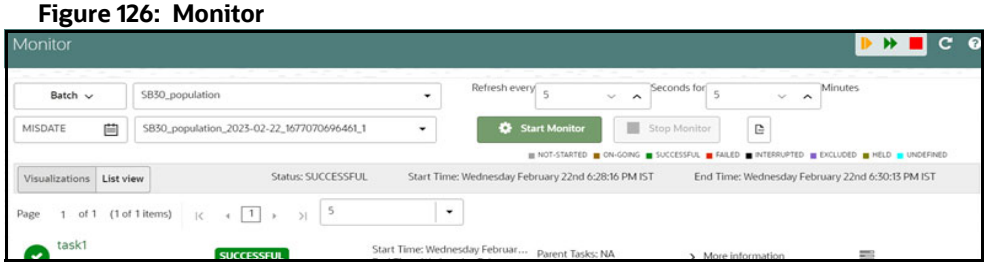
Figure 125: Schedule Batch



7.15.6 Monitor Batch

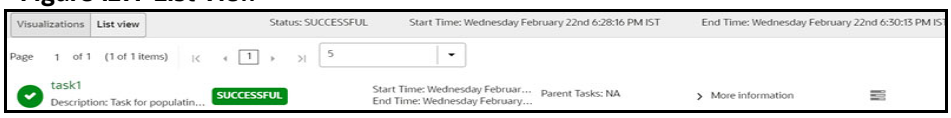
1. On **Orchestration** menu, click **Monitor Batch**.
2. Select the desired batch name from the drop-down list.
3. Choose the batch ID that has to be monitored.
4. Click **Start Monitor** to start monitoring the batch.

Figure 126: Monitor



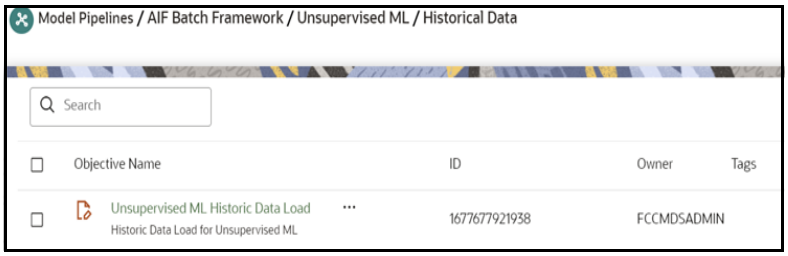
5. Click **List View** to view the status of the batch.
6. After the batch has been successfully executed, the status for the batch will be "successful".

Figure 127: List View



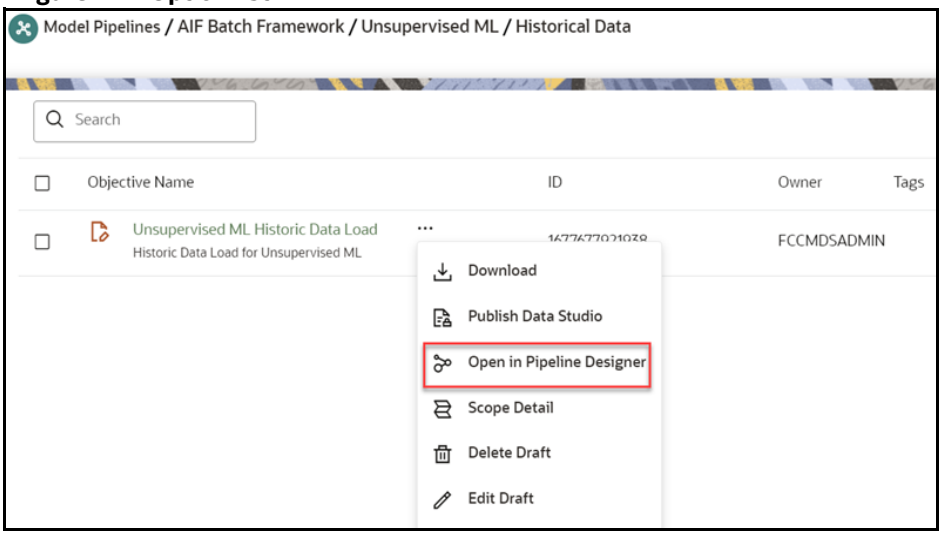
7. For further verification of the successful batch execution, navigate to "`Home > /Modeling / Pipelines/AIF Batch Framework/Unsupervised ML/Historical Data`," where the draft is located.

Figure 128: Historical Data



8. Click **Action** ••• icon next to <Objective Name> to view the list of options. The following page is displayed.

Figure 129: Option list



9. Click **Open in Pipeline Designer** and click **Notebook** tab**.**
10. Verify if all the draft paragraphs have been executed successfully and displayed no failure messages.

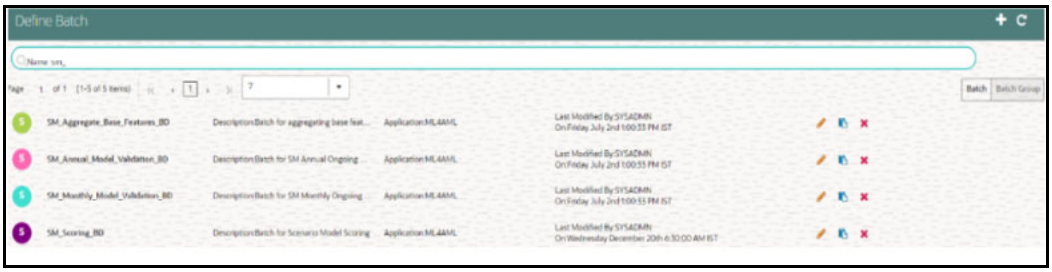
Figure 130: Batch Parameters

7.15.7 Scenario Model Batch Framework

The following batches are available in the out-of-the-box for the scenario model framework:

- **SM Aggregate Base Features**
- **SM Scoring**
- **Annual Model Validation**
- **Monthly Model Validation**

Figure 131: Define Batch for Scenario Model



7.15.7.1 SM Aggregate Base Features

- This pre-seeded batch will be available in all the workspaces (Production and Sandboxes).

> **NOTE** This batch is to be executed in the **Sandbox** workspace.

- This batch creates base features for scenario model training in the sandbox workspace.

7.15.7.1.1 Batch and Task Parameters

The batch contains a single task named **Aggregate_Base_Features.**

Figure 132: Define Task for Aggregate_Base_Features

7.15.7.1.1.1 Task: Aggregate_Base_Features, Task Parameters

- Objective folder for this task:

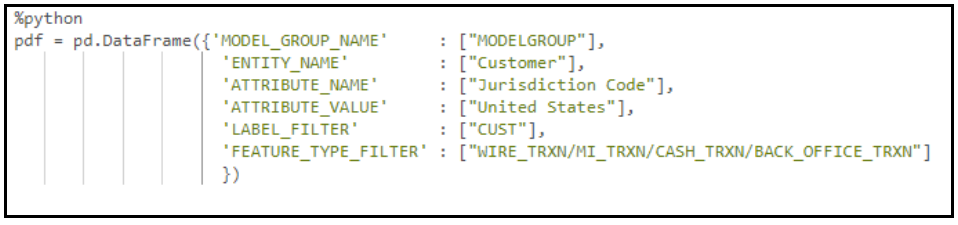
  `Home / Modeling / Pipelines / ML4AML / Scenario Model / Batch / Base Features`

  NOTE Do not change any parameter except **Optional Parameters.**

- Optional Parameters:
  - **model_group_name:** Name of the Model Group for which Base Feature Aggregation is to be created. Example: **LOB1**.
  - **model_name**: Name of the Model used while importing the model template using Admin Notebook. Example: **RMF**.
  - **from_date:** The start date for the Historic Data lookup is in **DD-MM-YYYY** format.
  - **to_dat**e: End Date for Historic Data lookup in **DD-MM-YYYY** format.
  - **prod_flag:** Flag to indicate Training/Scoring scenario. The option is **Y** or **N**.
  - For sandbox/historic training scenarios, the **prod_flag** should be set to **N**.
  - **include_full_lookback:** Flag to indicate whether the lookback should consider data beyond the from_date to aggregating base features. The option is **Y** or **N**.
  - **last_run_date:** The last run date within the from_date and to_date range, which exactly matches the scenario run date in **DD-MM-YYYY** format**.**
  - **frequency:** The frequency of the scenario execution.

    For example: **1** (Daily), **7** (Weekly), **14** (Bi-weekly), **30/31** (Monthly).
  - **look_back:** The lookback period for the scenario. For example: **30**.
  - **focus:** The model entity name is provided in the Admin notebook dataframe while creating the model group. The option is **CUSTOMER** or **ACCOUNT**.

    Figure 133: Parameters for Aggregate Base Features

    
    ```
    %python
    pdf = pd.DataFrame({'MODEL_GROUP_NAME'     : ["MODELGROUP"],
                        'ENTITY_NAME'          : ["Customer"],
                        'ATTRIBUTE_NAME'       : ["Jurisdiction Code"],
                        'ATTRIBUTE_VALUE'      : ["United States"],
                        'LABEL_FILTER'         : ["CUST"],
                        'FEATURE_TYPE_FILTER' : ["WIRE_TRXN/MI_TRXN/CASH_TRXN/BACK_OFFICE_TRXN"]
                        })
    ```

    filters: Scenario specific parameters that are used to give additional control for the base feature aggregation. The format to be provided is as follows:

    Param1 : Value1 ~ Param2 : Value2a | Value2b | Value2c

    For example: PRIMARY_CUST_FL : Y ~ MANTAS_BUSINESS_ACCT_TYPES : RBK | RBR ~ INCL_CASH_TRXN_PRDCT_TYPE_LST:DEBIT-CARD|SVC|CREDIT-CARD|CURRENCY|PHYS

Figure 134: Edit Task for Aggregate Base Features

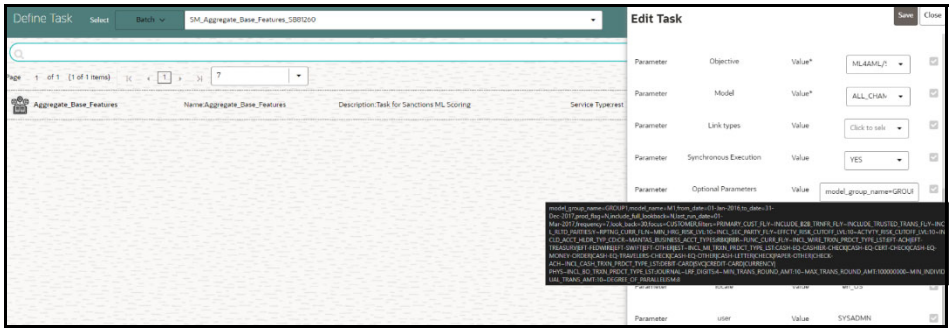


Table 46 describes the task parameter and its description for the scenario model aggregate base features.

Table 46: Task Parameters for Scenario Model Aggregate Base Features

Parameter	Description
PRIMARY_CUST_FL	It indicates what accounts are included by customer focus. The values are: • **Y**: Cover only accounts for which a customer plays a primary role. • **N**: Cover accounts over which a customer has discretion.
INCLUDE_B28_TRNFR_FL	It controls the inclusion or exclusion of bank-to-bank transactions. The values are: • **Y**: Includes transactions with a bank-to-bank transfer. • **N**: Excludes transactions with a bank-to-bank transfer, and the originator or beneficiary is the ultimate originator or beneficiary of the funds (i.e., Pass Through Indicator is set to No).
INCLUDE_TRUSTED_TRANS_FL	It controls the inclusion or exclusion of transactions designated as trusted transactions. Trusted transactions are those considered trusted based upon the presence of one or more trusted pairs (parties identified as enjoying a trusted relationship) on the transaction. The values are: • **Y**: Include trusted transactions • **N**: Exclude trusted transactions
INCL_RLTD_PARTIES	It allows coverage of all transactions between related parties. The values are: • **Y**: Covers all transactions. • **N**: Excludes transactions between related parties.
RPTNG_CURR_FL	The value is **Y** or **N**. If Y, then all aggregation is to be done on reporting currency.
MIN_HRG_RISK_LVL	Minimum list risk level greater than or equal to (>=) a transaction considered high risk.
INCL_SEC_PARTY_FL	It controls the inclusion or exclusion of secondary parties. The value is **Y** or **N**.

Table 46: Task Parameters for Scenario Model Aggregate Base Features

Parameter	Description
EFFCTV_RISK_CUTOFF_LVL	The effective risk level is specified for the conditional thresholds, which will be decided for overall risk.
ACTVTY_RISK_CUTOFF_LVL	The activity risk level is specified for the conditional thresholds, which will be decided for overall risk.
INCLD_ACCT_HLDR_TYP_CD	List of Account Types included by the scenario.
MANTAS_BUSINESS_ACCT_TYPES	Codes that identify the business purpose or usage of this account for scenarios.
FUNC_CURR_FL	The value is **Y** or **N**. If Y, all aggregation will be done on the functional currency. **Note:** If both reporting and functional currency are passed as "N", then it will be considered as the base currency.
INCL_WIRE_TRXN_PRDCT_TYPE_LST	A list of transaction product type codes for wire transactions is included in the scenario.
INCL_MI_TRXN_PRDCT_TYPE_LST	A list of transaction product type codes for monetary instrument transactions is included in the scenario.
INCL_CASH_TRXN_PRDCT_TYPE_LST	A list of transaction product type codes for cash transactions is included in the scenario.
INCL_BO_TRXN_PRDCT_TYPE_LST	A list of transaction product type codes for back-office transactions is included in the scenario.
LRF_DIGITS	Considering the number of the last digit as zero for the round amount.
MIN_TRANS_ROUND_AMT	Considering the minimum amount for round amount.
MAX_TRANS_ROUND_AMT	Considering the maximum amount for round amount.
MIN_INDIVIDUAL_TRANS_AMT	Minimum supported amount for LRT scenario.
DEGREE_OF_PARALLELISM	This should be configured properly for performance gain for SQL execution in parallel degree.

For example : model_group_name=**VALIDATION**, model_name=RMF_**LRT**, from_date=**01-Jan-2012**, to_date=**31-Dec-2017**, prod_flag=**N**, include_full_lookback=**N**, last_run_date=**09-May-2016**, frequency=**7**, look_back=**30**, focus=**CUSTOMER**, filters=**PRIMARY_CUST_FL:Y~INCLUDE_B2B_TRNFR_FL:Y~INCLUDE_TRUSTED_TRANS_FL:Y~INCL_RLTD_PARTIES:Y~RPTNG_CURR_FL:N~MIN_HRG_RISK_LVL:10~INCL_SEC_PARTY_FL:Y~EFFCTV_RISK_CUTOFF_LVL:10~ACTVTY_RISK_CUTOFF_LVL:10~INCLD_ACCT_HLDR_TYP_CD:CR~MANTAS_BUSINESS_ACCT_TYPES:RBK|RBR~FUNC_CURR_FL:Y~INCL_WIRE_TRXN_PRDCT_TYPE_LST:EFT-ACH|EFT-TREASURY|EFT-FEDWIRE|EFT-SWIFT|EFT-OTHER|EST~INCL_MI_TRXN_PRDCT_TYPE_LST:CASH-EQ-CASHIER-CHECK|CASH-EQ-CERT-CHECK|CASH-EQ-MONEY-ORDER|CASH-EQ-TRAVELERS-CHECK|CASH-EQ-OTHER|CASH-LETTER|CHECK|PAPER-OTHER|CHECK-ACH~INCL_CASH_TRXN_PRDCT_TYPE_LST:DEBIT-CARD|SVC|CREDIT-CARD|CURRENCY|PHYS~INCL_BO_TRXN_PRDCT_TYPE_LST:JOURNAL~LRF_DIGITS:4~MI**

N_TRANS_ROUND_AMT:10~MAX_TRANS_ROUND_AMT:100000000~MIN_INDIVIDUAL_TRANS_AMT:10~DEGREE_OF_PARALLELISM:8

- **Edit** Task Parameters and **Save**.

7.15.7.2 SM Scoring

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **NOTE** This batch is to be executed in the **Production** workspace.

7.15.7.2.1 Batch and Task Parameters

The batch contains the following tasks:

- Task 1: Aggregate_Scoring_Base_Features
- Task 2: ML_Scoring
- Task 3: Event_Processing
- Task 4: Output Overlays (**Optional**)
- Task 5: Aggregate_Base_Features for Additional Segments (**Optional**)

Figure 135: Define Task for SM Scoring

7.15.7.2.1.1 Task 1: Aggregate_Base_Features, Task Parameters

- Objective folder for this task:

  `Home / Modeling / Pipelines / ML4AML / Scenario Model / Batch / Base Features`

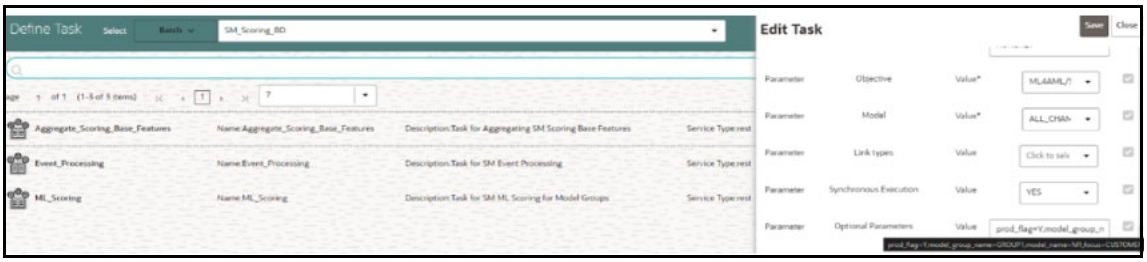
> **NOTE** Do not change any parameter except **Optional Parameters.**

- Optional Parameters:
  - **prod_flag:** Flag to indicate Training/Scoring scenario. The option is **Y** or **N**. For production/ scoring scenarios, the **prod_flag** should be set to **Y**.
  - **model_group_name:** Name of the Model Group for which Base Feature Aggregation is created. Example: **LOB1**.
  - **model_name:** Name of the Model used while importing the model template using Admin Notebook. Example: **RMF**.
  - **focus:** The model entity name is provided in the Admin notebook dataframe while creating the model group. The option is **CUSTOMER** or **ACCOUNT**.

    For example:
    prod_flag=**Y**,model_group_name=**GROUP1**,model_name=**M1**,focus=**CUSTOMER**

- **Edit** Task Parameters and **Save**.

Figure 136: Edit Task for SM Scoring



7.15.7.2.1.2 Task 2: ML_Scoring, Task Parameters

- Objective folder for this task:

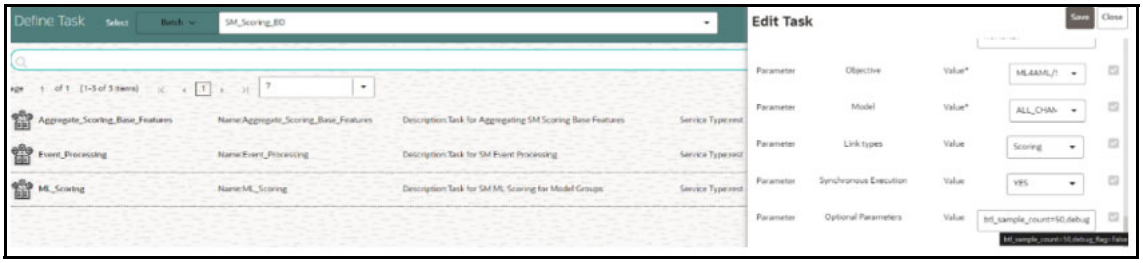
```
Home / Modeling / Pipelines / ML4AML / Scenario Model / AIF
```

> **NOTE** Do not change any parameter except **Optional Parameters.**

- Optional Parameters:
  - **btl_sample_count:** Number of random samples below the cutoff that should be considered while scoring.
  - **debug_flag:** Used for debugging purposes only. By default, set it to **False**.
  - **n_top_contrib:** Top N features contributing to model score. By default, set it to **None**.

    For example: btl_sample_count=**50**,debug_flag=**False**,n_top_contrib=**None**
- **Edit** Task Parameters and **Save**.

Figure 137: Edit Task Parameter for ML Scoring



The output for the ML_Scoring task is stored in the **SM_EVENT_SCORE** and **SM_EVENT_SCORE_DETAILS** tables. For more information on these tables, see the OFS Compliance Studio Data Model Reference Guide.

7.15.7.2.1.3 Task 3: Event_Processing Task Parameters

- Objective folder for this task:

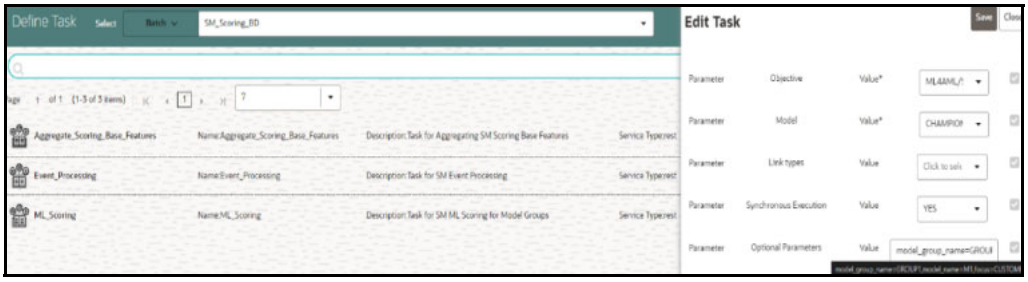
```
Home / Modeling / Pipelines / ML4AML / Scenario Model / Batch / Event
Processing
```

> **NOTE** Do not change any parameter except **Optional Parameters.**

- Optional Parameters:
  - **model_group_name:** Name of the Model Group for which Base Feature Aggregation is created. Example: **LOB1**.
  - **model_name:** Name of the Model used while importing the model template using Admin Notebook. Example: **RMF**.
  - **focus:** The model entity name is provided in the Admin notebook dataframe while creating the model group. The option is **CUSTOMER** or **ACCOUNT**.

    For example: model_group_name=**GROUP1**,model_name=**M1**,focus=**CUSTOMER**
  - **Edit** Task Parameters and **Save**.

Figure 138: Edit Task Parameter for Event Processing



7.15.7.2.1.4 Task: Output Overlays

This is an optional task added manually for running the score update notebook with static logic to update scores generated by the ML Scoring task.

This new task will be placed after the **ML_Scoring** task and before the **Event_Processing** task in the **SM_Scoring** batch.

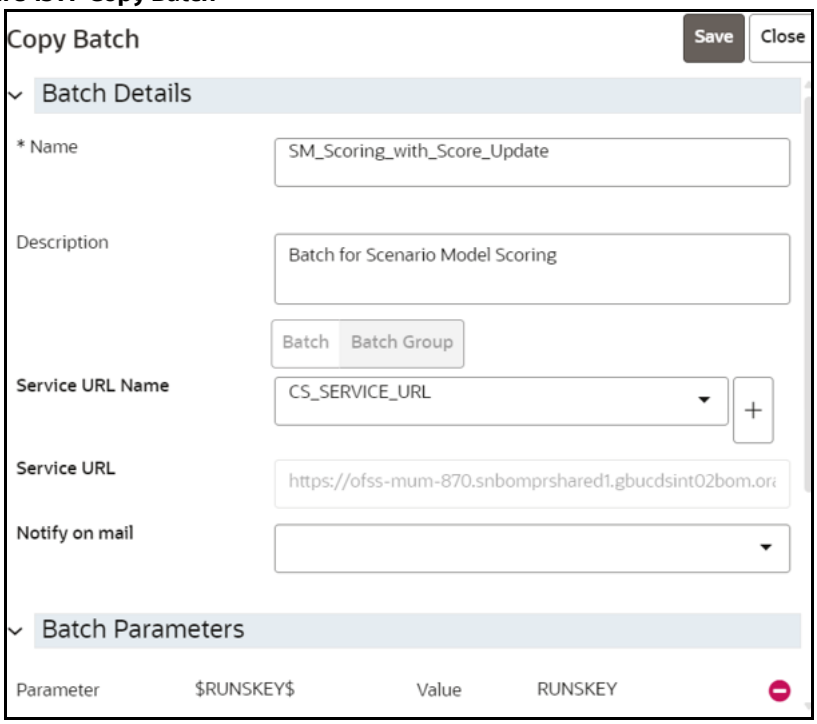
> **NOTE** **Prerequisites**: See the **Score Update Notebook for Scenario Model** section in the OFS Compliance Studio ML4AML Use Case Guide.

In the Production workspace, the score update notebook can be executed via batch framework.

For executing the score update notebook via batch framework, follow these steps:

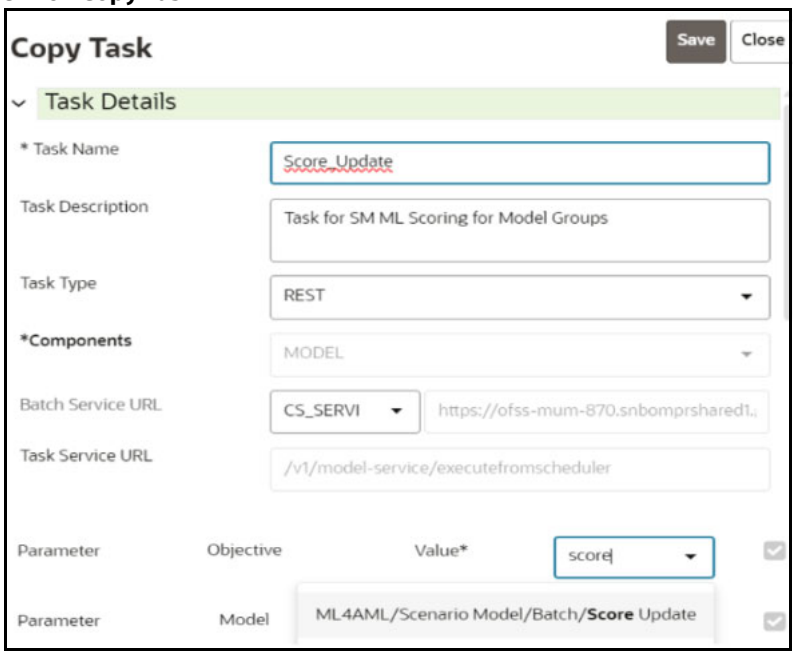
1. On the **Orchestration** mega menu, click **Define Batch**.
2. Search **SM_Scoring** Batch, and clone the batch using the icon. The Copy Batch page is displayed.

Figure 139: Copy Batch



3. Provide a new name to the batch and click **Save**.
4. On the **Orchestration** mega menu, click **Define Tasks** and select the newly created batch.
5. Copy any existing task using the icon. The Copy Task page is displayed.

Figure 140: Copy Task



6. Create a new task and provide the name as **Score_Update**.
7. Select the **Model** parameter where the draft notebook is present.

8. Click **Save**.
9. After the new Task is created, use the icon and adjust the Precedence Mapping of tasks.
10. Place the new task after **ML_Scoring** and before **Event_Processing** tasks as shown below.

Figure 141: Precedence Mapping

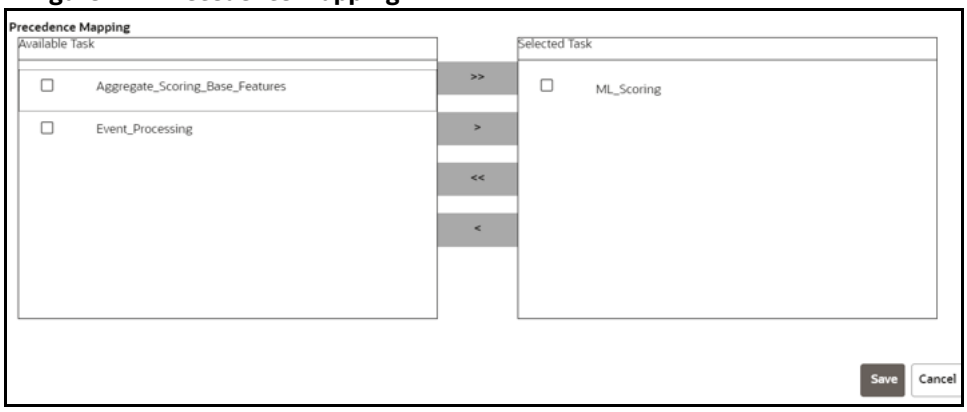
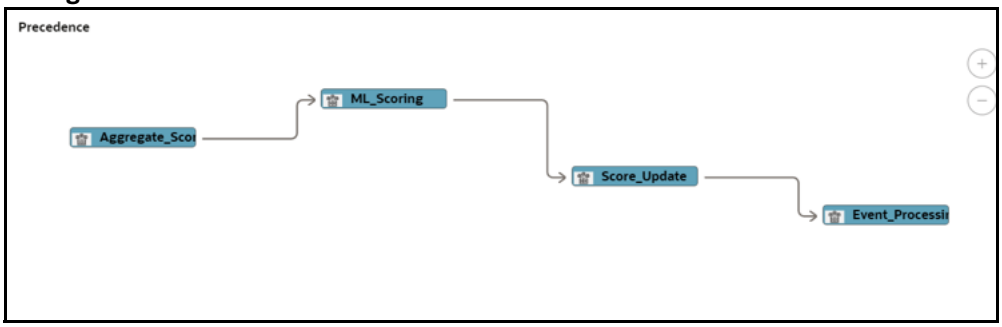


Figure 142: Precedence



11. On the **Orchestration** mega menu, click **Schedule Batches**.
12. Select the newly created batch, provide the parameters for each task, and trigger the batch.

    The newly created task will pass the control to the new notebook.

Figure 143: Event Score Update

```
-------------------------------------------------
N_AVG_TRXN_AM < 1000 and N_DLY_AMOUNT_VELOCITY < 120    : 0 records updated
-------------------------------------------------
N_MAX_TRXN_AM > 100000 and N_MIN_TRXN_AM > 5000         : 3 records updated
-------------------------------------------------
OCPTN_NM == "ENGINEER" and N_TOT_DR_TRXN_AM > 30000     : 48 records updated
-------------------------------------------------
OCPTN_NM == "Lawyer"    : 0 records updated
-------------------------------------------------
Event Scores Updated Successfully
```

The code in the new notebook will update the scores directly into the production table (SM_EVENT_SCORE_DETAILS).

7.15.7.2.1.5 Task: Aggregate_Base_Features for Additional Segments

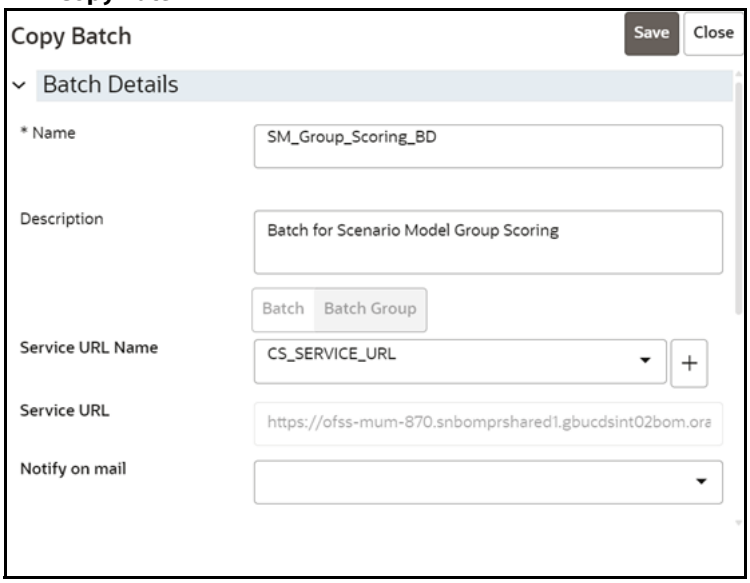
This is an optional task added manually for aggregating base features for additional segments.

This new task will be placed before the **ML_Scoring** task in the **SM_Scoring** batch.

To create additional tasks, follow these steps:

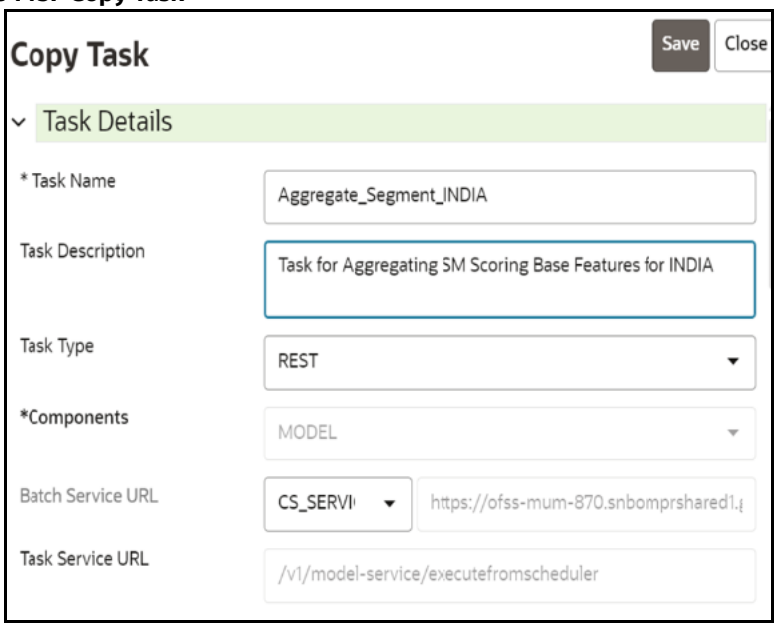
1. On the **Orchestration** mega menu, click **Define Batch**.
2. Search **SM_Scoring** Batch, and clone the batch using the icon. The Copy Batch page is displayed.

Figure 144: Copy Batch



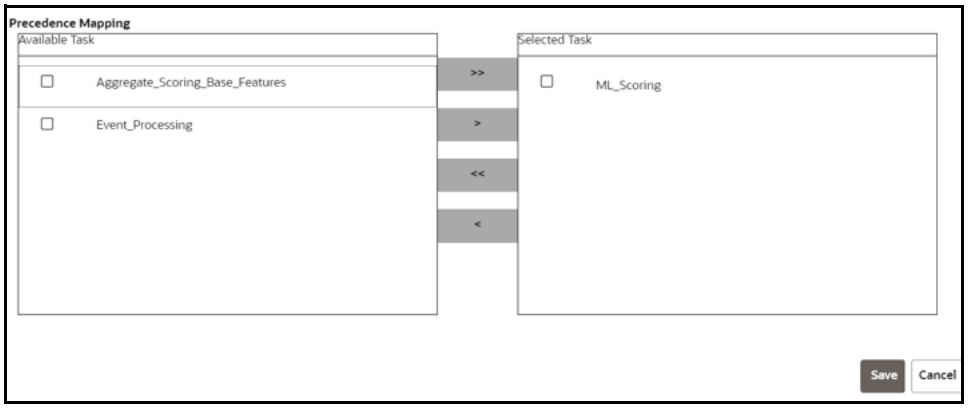
3. Provide a new name and and description to the batch and click **Save**.
4. On the **Orchestration** mega menu, click **Define Tasks** and select the newly created batch.
5. The pre-seeded **Aggregate_Scoring_Base_Features** task can be used for one of the additional segments.
6. For more segments, copy the existing **Aggregate_Scoring_Base_Features** task using the icon. The Copy Task page is displayed.

Figure 145: Copy Task



7. Provide the appropriate **Task Name** and **Task Description** for the additional segment.
8. Scroll down and navigate to **Task Parameters** in the **Optional Parameters** field and update the **model_group_name**, **model_name** or **focus** parameter values if needed.
9. Click **Save**.
10. After the new Task is created, use the icon and adjust the Precedence Mapping of tasks.
11. Place the new task before the **ML_Scoring** task as shown below.

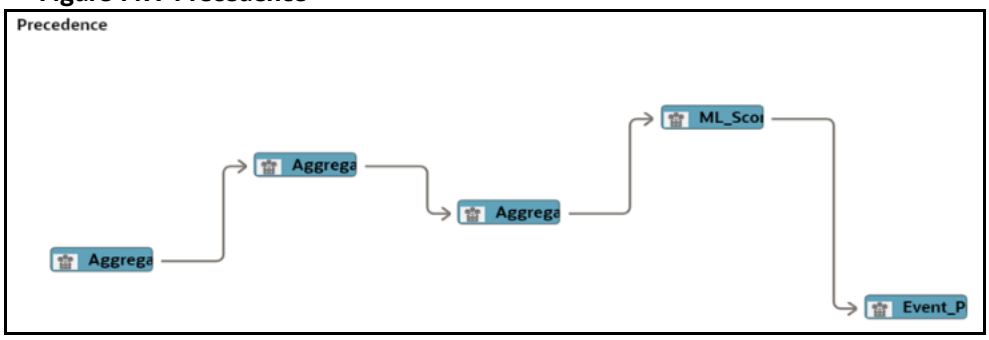
Figure 146: Precedence Mapping



NOTE If you want to create additional segments, then follow the steps 6-10 again.

12. Make sure to adjust the Precedence Mapping of tasks so that all the Data Aggregation Tasks are placed before the **ML_Scoring** task.

Figure 147: Precedence



13. On the **Orchestration** mega menu, click **Schedule Batches**.
14. Select the newly created batch, provide the parameters for each task, and trigger the batch.

7.15.7.3 Annual Model Validation

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

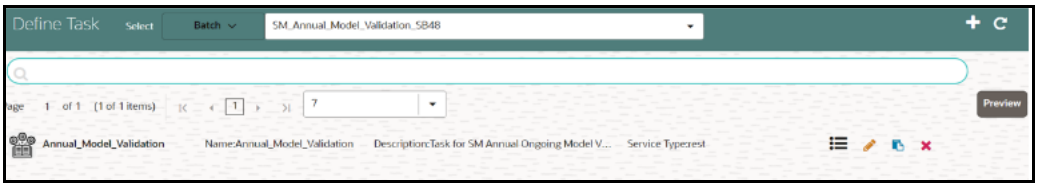
> **NOTE** This batch is to be executed in the **Production** workspace.

- This batch shows ongoing model performance annually.

7.15.7.3.1 Batch and Task Parameters

The batch contains a single task named Annual_Model_Validation.

Figure 148: Annual Model Validation for SM



7.15.7.3.1.1 Task: Annual_Model_Validation, Task Parameters

- Objective folder for this task:

  `Home / Modeling / Pipelines / ML4AML / Ongoing Model Validation / Annual`

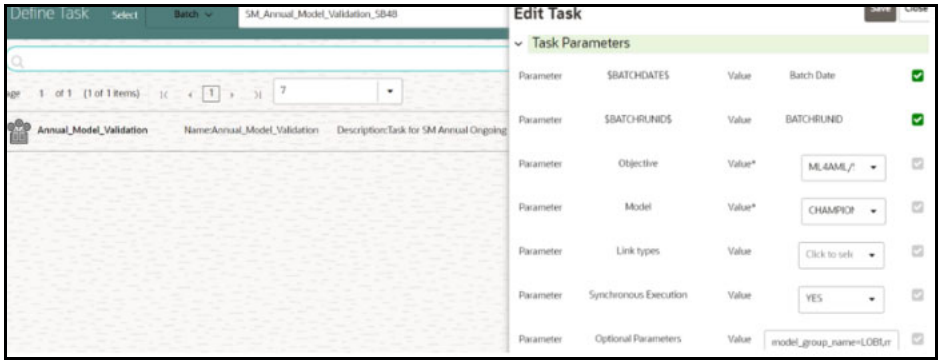
> **NOTE**
> - Do not change any batch/task parameter except **Optional Parameters.**
> - Optional Parameters can be edited from the **Schedule Batch** option.

- Optional Parameters:
  - **model_group_name**: Name of the Model Groups for which the model has been trained. Example **LOB1**.
  - **model_name:** Name of the Model for which the model has been trained. Example **RMF**.

- **focus:** Name of the entity type or segment. Example **CUSTOMER**.
- **model_id_list:** The user passes the parameter as deployed to use the deployed model. Example: **Deployed**.
- **from_date:** Start Date for Historic Data lookup in **DD-MM-YYYY**. Example **01-Jan-2016**.
- **to_date:** End Date for Historic Data lookup in **DD-MM-YYYY**. Example **31-Dec-2017**.

Example: model_group_name=**LOB1**,model__name=**RMF**,focus=**CUSTOMER,** from_date**=01-Jan-2016,**to_date**=31-Dec-2017**.

Figure 149: Edit Task for Annual Model Validation



7.15.7.4 Monthly Model Validation

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **NOTE** This batch is to be executed in the **Production** workspace.

- This batch shows ongoing model drift and data quality with respect to new data every month (monthly).

7.15.7.4.1 Batch and Task Parameters

The batch contains a single task named Monthly_Model_Validation.

7.15.7.4.1.1 Task: Monthly_Model_Validation, Task Parameters

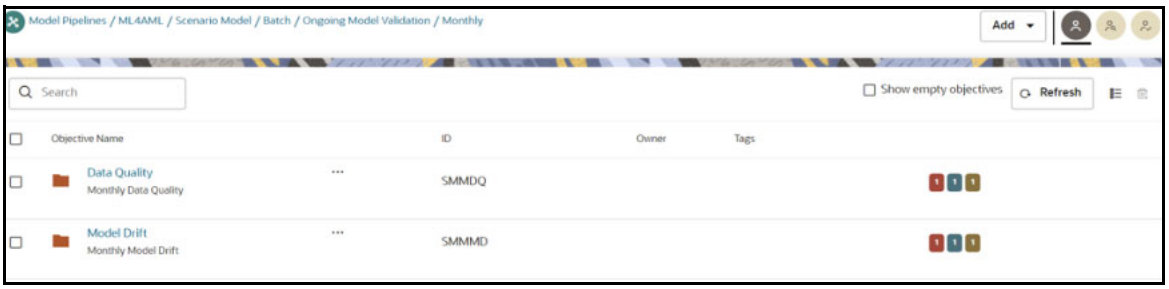
- Objective folder for Data Quality:

```
Home / Modeling / Pipelines / ML4AML / Ongoing Model Validation / Monthly
/ Data Quality
```

- Objective folder for Model Drift :

```
  Home / Modeling / Pipelines / ML4AML / Ongoing Model Validation /
Monthly    / Model Drift
```

Figure 150: Monthly Model Validation



NOTE

- Do not change any batch/task parameter except **Optional Parameters.**
- Optional Parameters can be edited from the **Schedule Batch** option.

- Optional Parameters:
  - **model_group_name:** Name of the Model Groups for which the model has been trained. Example: **LOB1**.
  - **model_name:** Name of the Model for which the model has been trained. Example: **RMF**.
  - **focus:** Name of the entity type or segment. Example: **CUSTOMER**.
  - **model_id:** User passes parameter as Deployed to use the deployed model. Example: **Deployed**.
  - **FEATURE_INCLUDE:** List of features to be included for data quality. The default **None** means which includes everything.
  - **FEATURE_EXCLUDE:** List of features to be excluded for data quality. The default **None** means which excludes nothing.

NOTE

If both include and exclude actions are provided, then include takes precedence over exclude action.

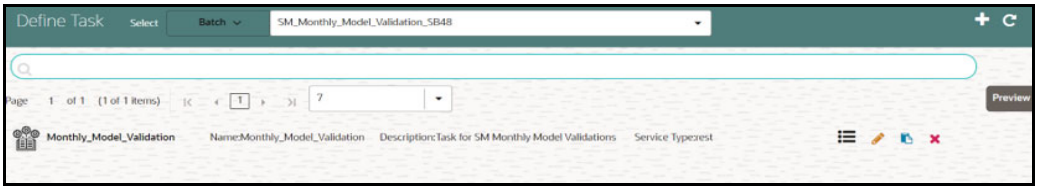
Example 1: feature_include="Feature1~Feature2"

Example 2: feature_exclude="Feature3~Feature4~Feature5"

  - **look_back_months:** Number of periods to look back for getting drift history. The default value is **5**.
  - **Number_Of_Bins:** Number of bins to be used in discretizing (scalar). The default value is **9**.
  - **Boot_Strap_Samples:** Number of bootstrap samples on which to estimate thresholds. The default value is **5**.
  - **Standard_Deviation_Band_Sigma:** Number of standard deviation bands (sigma band) for threshold setting to be used. The default value is **2 sigma**.

    For example:
    model_group_name=LOB1,model_name=RMF,focus=CUSTOMER,Number_Of_Bins=9,Boot_Strap_Samples=5,Standard_Deviation_Band_Sigma=2,look_back_months=5,FEATURE_INCLUDE=None,FEATURE_EXCLUDE=None

Figure 151: Define Task for Monthly Model Validation



7.16 Data Movement

7.16.1 Supervised

NOTE

- You must drop the partition before re-deployment for the particular model group.
- To drop a partition, run the following SQL commands:

  ```
  ALTER TABLE AIF_NON_BEHAVIORAL_DATA_PROD DROP PARTITION <MODEL_GROUP_NAME>;
  ```

  ```
  ALTER TABLE AIF_BEHAVIORAL_DATA_PROD DROP PARTITION <MODEL_GROUP_NAME>;
  ```
- Import/Export utility is available under the folder `$<Compliance_Studio_HOME>/deployed/ml4aml/datamovement`

7.16.1.1 Export from Sandbox

NOTE This section is intended for DBA/UNIX Admin.

1. Provide read/write/execute permissions to Export_Sandbox_Data.sh
2. Execute following Unix command

   ```
   dos2unix Export_Sandbox_Data.sh
   ```
3. Following grants are needed on Sandbox_Schema / Export_Schema (using sysdba)

   ```
   grant read, write on directory DATA_PUMP_DIR to export_schema_name;
   ```

   ```
   grant export full database to export_schema_name;
   ```
4. Execute the export utility using the following command

   ```
   ./Export_Sandbox_Data.sh
   ```

   a. Provide Oracle schema details when prompted

   b. Model Group Name will also be captured as part of inputs.

7.16.1.1.1 Outputs

`AIF_DATA.dmp` will be created as part of successful execution.

7.16.1.1.2 Execution Logs

`EXP_AIF_DATA.log` will be created as part of the execution in case of any issues.

> **NOTE** Oracle Drive Compatibility:
>
> 1. This utility can be executed from the same BD folder if the oracle drivers for the BD client and sandbox database server are compatible.
> 2. If not compatible, this utility can be copied to the database UNIX server of the sandbox schema under the folder DATA_PUMP_DIR.
> 3. DATA_PUMP_DIR for any oracle database server can be found out using the following query (using sysdba)
> 4. `select * from dba_directories where directory_name = 'DATA_PUMP_DIR'`

7.16.1.2 Import into Production

> **NOTE** This section is intended for DBA/UNIX Admin.

1. Copy `AIF_DATA.dmp` (output of export) and `Import_Sandbox_Data.sh` to DATA_PUMP_DIR of BD Production Database server.
2. Provide read/write/execute permissions to `AIF_DATA.dmp` and `Import_Sandbox_Data.sh`
3. Execute following Unix command

   ```
   dos2unix Import_Sandbox_Data.sh
   ```

4. Following grants are needed on BD Production Schema / Import Schema (using sysdba)

   ```
   GRANT read, write on directory DATA_PUMP_DIR to import_schema_name;
   GRANT import full database to import_schema_name;
   ```

5. Execute the import utility using the following command

   ```
   ./Import_Sandbox_Data.sh
   ```

   a. Provide Oracle schema details of the importing schema when prompted

   b. The Export schema user name / ID will also be captured as part of inputs.

7.16.1.2.1 Outputs

On successful execution, AIF_BEHAVIORAL_DATA & AIF_NON_BEHAVIORAL_DATA will be populated for the model group.

7.16.1.2.2 Execution Logs

`IMP_AIF_DATA.log` will be created as part of the execution in case of any issues.

> **NOTE** DATA_PUMP_DIR for any oracle database server can be found out using the following query (using sysdba)
>
> ```
> select * from dba_directories where directory_name =
> 'DATA_PUMP_DIR'
> ```

7.16.2 Unsupervised

> **NOTE**
> - You must drop the partition before re-deployment for the particular model group.
> - To drop a partition, run the following SQL commands:
>
>   ```
>   ALTER TABLE AIF_NON_BEHAVIORAL_DATA_PROD DROP
>   PARTITION <MODEL_GROUP_NAME>;
>   ```
>
>   ```
>   ALTER TABLE AIF_BEHAVIORAL_DATA_UNSUP_PROD DROP
>   PARTITION <MODEL_GROUP_NAME>;
>   ```
> - Import/Export utility is available under the folder `$<Compliance_Studio_HOME>//deployed/ml4aml/datamovement`

7.16.2.1 Export from Sandbox

> **NOTE** This section is intended for DBA/UNIX Admin.

1. Provide read/write/execute permissions to `Export_Sandbox_Data.sh`
2. Execute following Unix command

   ```
   dos2unix Export_Sandbox_Data.sh
   ```
3. Following grants are needed on Sandbox_Schema / Export_Schema (using sysdba)

   ```
   grant read, write on directory DATA_PUMP_DIR to export_schema_name;
   ```

   ```
   grant export full database to export_schema_name;
   ```
4. Execute the export utility using the following command

   ```
   ./Export_Sandbox_Data.sh
   ```

   a. Provide Oracle schema details when prompted

   b. Model Group Name will also be captured as part of inputs.

7.16.2.1.1 Outputs

`AIF_DATA_UNSUP.dmp` will be created as part of successful execution.

7.16.2.1.2 Execution Logs

`EXP_AIF_DATA_UNSUP.log` will be created as part of the execution in case of any issues.

> **NOTE** Oracle Drive Compatibility:
>
> 1. This utility can be executed from the same BD folder if the oracle drivers for the BD client and sandbox database server are compatible.
> 2. If not compatible, this utility can be copied to the database UNIX server of the sandbox schema under the folder DATA_PUMP_DIR.
> 3. DATA_PUMP_DIR for any oracle database server can be found out using the following query (using sysdba)
>
> ```
> select * from dba_directories where
> directory_name = 'DATA_PUMP_DIR'
> ```

7.16.2.2 Import into Production

> **NOTE** This section is intended for DBA/UNIX Admin.

1. Copy `AIF_DATA.dmp` (output of export) and `Import_Sandbox_Data.sh` to `DATA_PUMP_DIR` of BD Production Database server.
2. Provide read/write/execute permissions to `AIF_DATA.dmp` and `Import_Sandbox_Data.sh`
3. Execute following Unix command

   ```
   dos2unix Import_Sandbox_Data.sh
   ```

4. Following grants are needed on BD Production Schema / Import Schema (using sysdba)

   ```
   GRANT read, write on directory DATA_PUMP_DIR to import_schema_name;

   GRANT import full database to import_schema_name;
   ```

5. Execute the import utility using the following command

   ```
   ./Import_Sandbox_Data.sh
   ```

   a. Provide Oracle schema details of the importing schema when prompted

   b. The Export schema user name / ID will also be captured as part of inputs.

7.16.2.2.1 Outputs

On successful execution, AIF_BEHAVIORAL_DATA_UNSUP will be populated for the model group.

7.16.2.2.2 Execution Logs

`IMP_AIF_DATA _UNSUP.log` will be created as part of the execution in case of any issues.

> **NOTE** DATA_PUMP_DIR for any oracle database server can be found out using the following query (using sysdba).
>
> ```
> select * from dba_directories where directory_name =
> 'DATA_PUMP_DIR'.
> ```

7.17 ECM Connector Batch

7.17.1 Supervised ML-ECM Connector Batch

Post Supervised ML Scoring Batch, execute ML-ECM connector batch from ECM UI (AIF-ECM connector batch)

- **RRF Run Name**: Oracle AIF Event Processing in ECM
- **RRF Run code**: Oracle_AIF_Event_Processing
- **RRF Run Parameters**: FIC MIS Date (should match the FIC MIS date of ML scoring batch)

For more information on how to navigate to RRF/Batch framework for the execution in the **Performing Batch Run** section in the OFS ECM Administration and Configuration Guide.

7.17.2 Typology Model-ECM Connector Batch

Post Typology scenario execution Batch, execute **Oracle_ML4AML_Scenario_Events** connector batch from ECM UI:

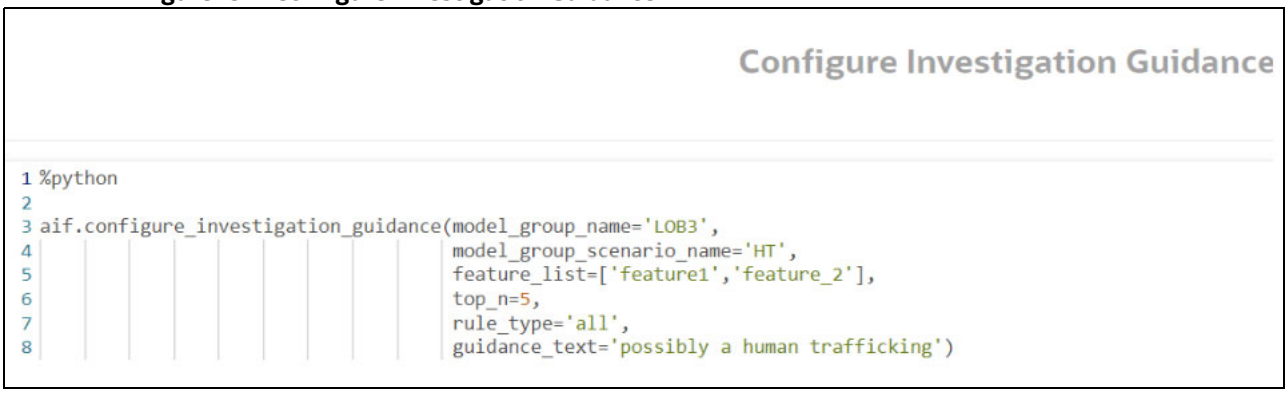
- **RRF Run Name**: Oracle ML4AML Scenario Event Processing in ECM
- **RRF Run code**: Oracle_ML4AML_Scenario_Events
- **RRF Run Parameters**: FIC MIS Date (should match the FIC MIS date of ML4AML typology scenario execution batch)

For more information on how to navigate to RRF/Batch framework for the execution in the **Performing Batch Run** section in the OFS ECM Administration and Configuration Guide.

7.18 Configure Investigation Guidance

Use `aif.configure_investigation_guidance()` API to load investigation guidance data in the `aif_investigation_guidance` table.

Figure 152: Configure Investigation Guidance



The `aif_investigation_guidance` table columns are as follows:

- V_MODEL_GROUP
- V_MODEL_GROUP_SCENARIO_NAME
- V_FEATURES
- TOP_N

- RULE_TYPE
- V_GUIDANCE_TEXT

The following parameters are the input value for the paragraph:

- **model_group_name**: Model group name for which you need to configure the investigation guidance.
- **model_group_scenario_name**: Model group scenario name for which you need to configure the investigation guidance.
- **feature_list**: The set of model features to be configured for investigation guidance.

  For example, ['feature1', 'feature2']
- **top_n**: The top N contributing features to be searched in the Model to consider for investigation guidance. The default value is **10**.
- **rule_type**: Consider feature(s) provided in the **feature_list** to be matched in model features. The default value is **any.**
  - **any**: Any one of the features in the **feature_list** will be matched with **top_n** contributing model features.
  - **all**: All of the features in the **feature_list** will be matched with **top_n** contributing model features.
- **guidance_text**: It provides the Investigation guidance for the following parameters:
  - Model group name
  - Model group scenario name
  - Feature list
  - Top N features

7.18.1 Output

The successful message is returned on successfully adding the top N features and Guided Text.

Returns error message if failed.

7.19 Data Model Support for AAI Applications

Oracle Data Model (ODM) data model support is added for the Unsupervised Customer Segmentation use case.

> **NOTE** This model should be uploaded as a **Logical** upload only (not as a **Physical** upload.

Perform the following:

1. Log in to Linux server as Compliance Studio (CS) user where CS is installed.
2. Navigate to `<COMPLIANCE_STUDIO_INSTALLED_PATH>/ml4aml/model/odm/ML4AML.ODM`

   The data model (`ML4AML.ODM`) is available as part of OFS Compliance Studio installation in the installed directory.
3. Copy `ML4AML.ODM` to AAI system or machine for uploading the model into AAI.

For more information on the ODM model upload, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.

7.20 Schema Grants for AML Event Scoring

To grant schema for AML Event scoring, follow these steps:

In Production Workspace

1. Provide the grant select of ECM related tables to the sandbox schema by using the following queries in the ECM atomic schema of the production database server:

   ```
   select 'GRANT SELECT ON '||TABLE_NAME ||' TO <sandbox_schema>;' from
   user_tables where table_name like 'FCC_%';
   ```

   For example: `select 'GRANT SELECT ON '||TABLE_NAME ||' TO EVENTSCORESANDBOX;' from user_tables where table_name like 'KDD_%';`

   ```
   select 'GRANT SELECT ON '||TABLE_NAME ||' TO <sandbox_schema>;' from
   user_tables where table_name like 'KDD_%';
   ```

   For example: `select 'GRANT SELECT ON '||TABLE_NAME ||' TO EVENTSCORESANDBOX;' from user_tables where table_name like 'KDD_%';`

2. Copy the output of the above executed queries and execute in the ECM atomic schema of the production database server.

In Sandbox Workspace

1. Export the ECM atomic schema dump from the production database server and import it to the sandbox database server.
2. Provide the grant select of ECM related tables to the sandbox schema by using the following queries in the ECM atomic schema of the sandbox database server:

   ```
   select 'GRANT SELECT ON '||TABLE_NAME ||' TO <sandbox_schema>;' from
   user_tables where table_name like 'FCC_%';
   ```

   For example: `select 'GRANT SELECT ON '||TABLE_NAME ||' TO EVENTSCORESANDBOX;' from user_tables where table_name like 'KDD_%';`

   ```
   select 'GRANT SELECT ON '||TABLE_NAME ||' TO <sandbox_schema>;' from
   user_tables where table_name like 'KDD_%';
   ```

   For example: `select 'GRANT SELECT ON '||TABLE_NAME ||' TO EVENTSCORESANDBOX;' from user_tables where table_name like 'KDD_%';`

3. Copy the output of the above executed queries and execute in the ECM atomic schema of the sandbox database server.

7.21 Fine Grain Data Access Control for Workspace

Institutions often need to restrict data access to users based on jurisdiction to comply with data residency or other privacy regulations. This functionality can be used to ensure that users will be able to access data only from those jurisdictions they are entitled to.

Prerequisites:

- Assuming existing / new Users are created using **AAI** or third-party **IDCS**.
- Security mapping between users to jurisdictions is done using AML BD application UI.
- User Mapped Jurisdiction and Threshold set Jurisdictions should match.
  - User Mapped Jurisdiction will take the priority if they do not match.
- User not mapped with any jurisdiction will not see/get all jurisdiction's data.
- Provide the following grant through SYS user where the workspace schema is created.

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

> **NOTE** If the accounts belonging to a customer do not belong to the same jurisdiction as the customer, but instead span multiple jurisdictions, the user executing the scenario should have access to all the relevant jurisdictions. If the user executing the scenario does not have access to the appropriate jurisdictions, then the scenario will not generate the expected number of alerts.

7.21.1 Sync up Security Mapper between BD Production and ASC BD Schema

> **NOTE** This step is optional and can be skipped if user management and security mapping for **ASC-BD** is self-managed.

- Generally, security mappings are done for BD production instances.
  - New user creations / user-security mapping happens in the BD Production.
- ASC BD instance is generally a **non-prod** BD, like BD UAT, BD Pre-Prod, etc...
- If user management and security mapping happens outside of the **ASC-BD** instance (say in BD Production), then the security mapper table needs to be synced up between **ASC-BD** and **BD-Prod.** Here is the approach for sync up users.
  - Create a new Data Store in the Compliance Studio pointing to BD Production Schema.
  - During ASC workspace creation, add BD Production Data Source and source following tables.
    - kdd_jrsdcn
    - kdd_review_owner
    - kdd_review_owner_jrsdcn

- Execute **Workspace data population batch** to sync up the security mapper with ASC-BD.

> **NOTE** This step must be repeated every time when users/security-mappings are created/modified.

7.21.2 Enable/Disable Fine Grain Data Access Control

To enable/disable fine grain data access control, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ml4aml/bin` directory.
3. Execute the following UNIX commands once against the ASC workspace.

```
./enableVPD.sh -w <ASC_Workspace_Target_Wallet_Alias>
```

> **NOTE** **ASC_Workspace_Target_Wallet_Alias** id is the placeholders to be replaced with actual values used to create ASC workspace.

4. Login to the configuration schema (Studio Schema) of the Compliance Studio.
   a. Run the following SQL to enable VPD.

```
MERGE INTO NEXTGENEMF_CONFIG T USING (SELECT 'IS_VPD_ENABLED' V_NAME
FROM DUAL)  S
ON (T.V_NAME = S.V_NAME)
WHEN MATCHED THEN UPDATE SET V_VALUE = 'Y',
V_DESC = 'Is VPD Enabled'
WHEN NOT MATCHED THEN INSERT (V_NAME, V_VALUE, V_DESC)
VALUES('IS_VPD_ENABLED', 'N', 'Is VPD Enabled')
```

   b. Run the following SQL to disable VPD.

```
MERGE INTO NEXTGENEMF_CONFIG T USING (SELECT 'IS_VPD_ENABLED' V_NAME
FROM DUAL)  S
ON (T.V_NAME = S.V_NAME)
WHEN MATCHED THEN UPDATE SET V_VALUE = 'N',
V_DESC = 'Is VPD Enabled'
WHEN NOT MATCHED THEN INSERT (V_NAME, V_VALUE, V_DESC)
VALUES('IS_VPD_ENABLED', 'Y', 'Is VPD Enabled')
```

8 Restart Services

Use this section to understand how to stop or start the Compliance Studio service if you have an issue with the services.

Topics:

8.1 Stop and Start the Compliance Studio Services

To stop the Compliance Studio installer, follow these steps:

1. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/bin` directory.
2. Run the following command:

   ```
   ./compliance-studio.sh --stop
   ```

To start the Compliance Studio services, follow these steps:

1. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/bin` directory.
2. Execute the following command in the console:

   ```
   ./compliance-studio.sh --start
   ```

8.2 Stop and Start the PGX Service

To stop the PGX service, follow these steps:

1. Navigate to the `<PGX_Installation_Path>/pgx/server/bin` directory.
2. Run the following command:

   `./pgx-server.sh --stop` or `./pgx-server.sh -k`

To start the PGX service, follow these steps:

1. Copy the `<Keystore file name>.jks` file from `<Compliance Studio Installation Path>/batchservice/conf` to the `<PGX Server path>/server/conf` directory.
2. Navigate to the `<PGX_Installation_Path>/pgx/server/bin` directory.
3. Run the following command:

   `./pgx-server.sh --start` or `./pgx-server.sh -s`

4. After the PGX service runs successfully, run the

   `./FCCM_Studio_ETL_BulkSimilarityEdgeGeneration.sh job` and `<FCCM_Studio path>/FCCM_Studio_ApplyGraphRedaction.sh` file.

> **NOTE** Ensure that the Global graph is loaded in the PGX Server.

9 Appendix

Topics:

9.1 Create Metadata Indexes using Logstash

To create metadata indexes using Logstash, perform the following:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/load-to-open-search/conf` directory.
2. Set the following parameter value as true in the `application.properties` file.

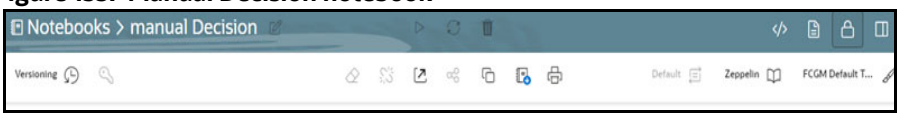
   ```
   index.logstash-conf.apply=true
   ```
3. Restart Compliance Studio services.
4. Create Indexes. Perform the steps specified in Create Index and Load the Data section.

9.2 Unlock the Notebook

1. Log in to the Compliance Studio application.
2. Navigate to the Compliance Studio server with the same URL by changing the port to 7008. `(http://hostname:7008 from http://hostname:7001/cs/home)`

3. Open the notebook. Unlock the notebook, and replace it with the new interpreter name in each paragraph.

Figure 153: Manual Decision notebook



4. Click **Write**  Paragraphs icon at the top-right corner to unlock the notebook.

9.3 Checking IP Address for User's Last Login

Navigate to the Compliance Studio schema in the database and run the following query:

```
select * from ds_user;
```

The output table will look like this:

Figure 154: Output Table

You can check the LAST_IP_ADDRESS column, which will contain the IP address from where the user has last logged in.

9.4 Roles, Functions and Permissions

9.4.1 Roles

A Role consists of one or more actions (functions/permissions). A Group can have single or multiple roles. For example, Admin, user, and guest. The Table 47 describes the Preconfigured Roles.

Table 47: Roles

Role Code	Role Name	Description
WKSPACC	Workspace Access	Workspace Access Role
WKSPAUTH	Workspace Authorize	Workspace Authorize Role
WKSPREAD	Workspace Read	Workspace Read Role
WKSPWRITE	Workspace Write	Workspace Write Role
FLDRACC	Folder Access	Folder Access Role
FLDRAUTH	Folder Authorize	Folder Authorize Role
FLDRREAD	Folder Read	Folder Read Role
FLDRWRITE	Folder Write	Folder Write Role
IDMGMTACC	Identity MGMT access	System admin access
IDMGMTADVN	Identity MGMT advanced	Identity management advanced
IDMGMTAUTH	Identity MGMT authorize	Identity management authorize

Table 47: Roles

Role Code	Role Name	Description
IDMGMTREAD	Identity MGMT read	Identity management read
IDMGMTWRIT	Identity MGMT write	Identity management write
FUNC_READ	Function Read Role	-
FUNC_WRITE	Function Write Role	-
FUNC_ADV	Function Advanced Role	-
ROLE_READ	Role Read Role	-
ROLE_WRITE	Role Write Role	-
ROLE_ADV	Role Advanced Role	-
ROLE_AUTH	Role Authorize Role	-
GRP_READ	Group Read Role	-
GRP_WRITE	Group Write Role	-
GRP_ADV	Group Advanced Role	-
GRP_AUTH	Group Authorize Role	-
USR_READ	User Read Role	-
USR_WRITE	User Write Role	-
USR_ADV	User Advanced Role	-
USR_AUTH	User Authorize Role	-
SRVC_READ	Service Read Role	-
APP_READ	Application Read Role	-
WRKSP_READ	Workspace Read Role	-
WRKSP_WRITE	Workspace Write Role	-
WRKSP_ADV	Workspace Advanced Role	-
FLDR_READ	Folder Read Role	-
FLDR_WRITE	Folder Write Role	-
FLDR_ADV	Folder Advanced Role	-
DTSRC_READ	DataStore Read Role	-
ADMIN_LINK	Admin Link Role	-
BATCH_READ	Batch Read Role	Batch read role in scheduler service
BATCH_WRITE	Batch Write Role	Batch write role in scheduler service
BATCH_ADV	Batch Advance Role	Batch advance role in scheduler service

Table 47: Roles

Role Code	Role Name	Description
BATCH_AUTH	Batch Authorization Role	Batch authorize role in scheduler service
BATCH_OPER	Bath Operation Role	Batch operation role in scheduler service
BATCH_MAINT	Batch Maintenance Role	Batch maintenance role in scheduler service
MDLACCESS	Model Access	User Group mapped will have access to Model Link and Summary
MDLREAD	Model Read	Model Read
MDLWRITE	Model Write	Model Write
MDLPHANTOM	Model Phantom	Model Phantom
MDLAUTH	Model Authorize	Model Authorize
MDLADV	Model Advanced	Model Advanced
MDLREVIEW	Model Review	Model Review
MDLDEPLOY	Model Deployment	Model Deployment
MDLADMIN	Model Admin	Model Admin
SBADMIN	Sandbox Admin	Sandbox Admin
DSREAD	DataStore Read	DataStore Read
DSWRITE	DataStore Write	DataStore Write
DSACCESS	DataStore Access	DataStore Access
DSADMIN	DSADMIN	Compliance Studio Admin Role
DSBATCH	DSBATCH	Batch Role
DSINTER	DSINTER	Compliance Studio Interpreter Configuration Role
DSUSER	DSUSER	Compliance Studio User Role
DSAPPROVER	DSAPPROVER	Manual Edges Approver role
DSREDACT	DSREDACT	Redaction role for Graph
MDLEXE	Model Execute	Model Execute
MDAPPROVER	MDAPPROVER	Approver
MDREQUESTER	MDREQUESTER	Requester

9.4.1.1 Default Roles Seeded in Notebook Server through permissions-int.yml file

Table 48 describes the Default Roles.

Table 48: Default Roles

Name	Description
DSADMIN	Admin Role (all permissions)
DSBATCH	Batch Role for running ETL and executing notebook using shell script
DSUSER	General Role (does not have access to modify Interpreter configurations or run batches)
DSINTER	Interpreter configurator Role
DSAPPROVER	A role for Approving Manual Edge
DSREDACT	Roles for applying redaction in Graph

9.4.2 Functions in Compliance Studio

Set of actions in the Compliance Studio. For example, limited_read, read, and write. A Role can have single or multiple functions. The Table 49 describes the Preconfigured Functions.

Table 49: Compliance Studio Functions

Function Code	Function Name	Description
WKSP_SUMM	Workspace Summary Access	The user mapped to this function can access the Workspace Summary Pages
WKSP_LNK_ACC	Workspace Link Access	The user mapped to this function can access the Workspace Links
WKSP_AUTH	Workspace Authorization	The user mapped to this function can authorize Workspace
WKSP_VIW	Workspace View	The user mapped to this function can view Workspace
WKSP_ADD	Workspace Add	The user mapped to this function can add Workspace
WKSP_CPY	Workspace Copy	The user mapped to this function can copy Workspace
WKSP_DEL	Workspace Delete	The user mapped to this function can delete Workspace
WKSP_EDIT	Workspace Edit	The user mapped to this function can edit Workspace
FLDR_LNK_ACC	Folder Link Access	The user mapped to this function can access the Folder Links

Table 49: Compliance Studio Functions

Function Code	Function Name	Description
FLDR_AUTH	Folder Authorization	The user mapped to this function can authorize Folder
FLDR_VIW	Folder View	The user mapped to this function can view the Folder
FLDR_CPY	Folder Copy	The user mapped to this function can copy Folder
FLDR_EDIT	Folder Edit	The user mapped to this function can edit the Folder
ADMINSCR	Administration Screen	The user mapped to this function can access the Administration Screen
FUNCMAINT	Function Maintenance Screen	The user mapped to this function can access the Function Maintenance Screen
FUNCROLE	Function Role Map Screen	The user mapped to this function can access the Function Role Map Screen
ROLEMAINT	Role Maintenance Screen	The user mapped to this function can access the Role Maintenance Screen
UGWKSPMAP	User Group Workspace Map Screen	The user mapped to this function can access the User Group Workspace Map Screen
UGFLROLMAP	User Group Folder Role Map Screen	The user mapped to this function can access the User Group Folder Role Map Screen
UGMAINT	User Group Maintenance Screen	The user mapped to this function can access the User Group Maintenance Screen
UGMAP	User Group User Map Screen	The user mapped to this function can access the User Group User Map Screen
UGROLMAP	User Group Role Map Screen	The user mapped to this function can access the User Group Role Map Screen
USRACTREP	User Activity Reports Screen	The user mapped to this function can access the User Activity Reports Screen
USRATTUP	User Attribute Upload Screen	The user mapped to this function can access the User Attribute Upload Screen
USRMAINT	User Maintenance Screen	The user mapped to this function can access the User Maintenance Screen

Table 49: Compliance Studio Functions

Function Code	Function Name	Description
USRATH	User Authorization Screen	The user mapped to this function can access the User Authorization Screen
FUNC_SUMM	Function Summary	-
FUNC_VIEW	Function View	-
FUNC_ADD	Function Add	-
FUNC_MOD	Function Modify	-
FUNC_DEL	Function Delete	-
FUNC_MAP	Function Map	-
FUNC_PURGE	Function Purge	-
ROLE_SUMM	Role Summary	-
ROLE_VIEW	Role View	-
ROLE_ADD	Role Add	-
ROLE_MOD	Role Modify	-
ROLE_DEL	Role Delete	-
ROLE_MAP	Role Map	-
ROLE_PURGE	Role Purge	-
ROLE_AUTH	Role Authorize	-
GRP_SUMM	Group Summary	-
GRP_VIEW	Group View	-
GRP_ADD	Group Add	-
GRP_MOD	Group Modify	-
GRP_DEL	Group Delete	-
GRP_MAP	Group Map	-
GRP_PURGE	Group Purge	-
GRP_AUTH	Group Authorize	-
USR_SUMM	User Summary	-
USR_VIEW	User View	-
USR_ADD	User Add	-
USR_MOD	User Modify	-
USR_DEL	User Delete	-
USR_MAP	User Map	-
USR_PURGE	User Purge	-

Table 49: Compliance Studio Functions

Function Code	Function Name	Description
USR_AUTH	User Authorize	-
SRVC_SUMM	Service Summary	-
SRVC_VIEW	Service View	-
APP_SUMM	Application Summary	-
APP_VIEW	Application View	-
WRKSP_SUMM	Workspace Summary	-
WRKSP_VIEW	Workspace View	-
WRKSP_ADD	Workspace Add	-
WRKSP_MOD	Workspace Modify	-
WRKSP_DEL	Workspace Delete	-
FLDR_SUMM	Folder Summary	-
FLDR_VIEW	Folder View	-
FLDR_ADD	Folder Add	-
FLDR_MOD	Folder Modify	-
FLDR_DEL	Folder Delete	-
DTSRC_SUMM	DataStore Summary	-
DTSRC_VIEW	DataStore View	-
ADMIN_LINK	Admin Link	-
BATCH_ADD	Batch Add Function	Batch add function in scheduler service
BATCH_DEL	Batch Delete Function	Batch delete function in scheduler service
BATCH_MOD	Batch Modify Function	Batch modify the function in scheduler service
BATCH_VIEW	Batch View Function	Batch view function in scheduler service
BATCH_SCH	Batch Schedule Function	Batch schedule function in scheduler service
BATCH_SUMM	Batch Summary Function	Batch summary function in scheduler service
BATCH_AUTH	Batch Authorize Function	Batch authorize function in scheduler service
BATCH_PURGE	Batch Purge Function	Batch purge function in scheduler service
BATCH_MON	Batch Monitor Function	Batch monitor function in scheduler service

Table 49: Compliance Studio Functions

Function Code	Function Name	Description
BATCH_EXEC	Batch Execute Function	Batch execution function in scheduler service
BATCH_COPY	Batch Copy Function	Batch Copy function in scheduler service
MDLCNFSUMM	Model Configuration Summary	This function gives access to Model Configuration Summary
MDLSUMM	Model Summary	This function gives access to the Model Summary
MDLVIEW	Model View	This function gives access to view Model
MDLTRACE	Model Trace	This function gives access to trace Model
MDLADD	Model Add	This function gives access to add Model
MDLCOPY	Model Copy	This function gives access to copy Model
MDLEDIT	Model Edit	This function gives access to edit Model
MDLDEL	Model Delete	This function gives access to delete Model
MDLAPPROVE	Model Approve	This function gives access to approve Model
MDLLOCK	Model Lock	This function gives access to the lock Model
MDLEXE	Model Execute	This function gives access to execute Model
MDLREVIEW	Model Review	This function gives access to review Model
MDLDEPL	Model Deploy	This function gives access to deploying Model
MDLPURGE	Model Purge	This function gives access to purge Model
SBADD	Sandbox Add	This function gives access to add Sandbox
DSADD	DataStore Add	The user mapped to this function can add DataStore
DSEDIT	DataStore Edit	The user mapped to this function can edit DataStore
DSDELETE	DataStore Delete	The user mapped to this function can delete DataStore

Table 49: Compliance Studio Functions

Function Code	Function Name	Description
DSVIEW	DataStore View	The user mapped to this function can view DataStore
DSSUMM	DataStore Access	The user mapped to this function can access the DataStore summary
MDAPPROVE	MDAPPROVE	The user mapped to this function can access the Match Rules, Merge Rules and Data Survival screen
MDREQUEST	MDREQUEST	The user mapped to this function can access the Manual Decisioning and Merge and Split Global Entities screen

9.4.3 Permissions in Notebook Server

Set of actions in the Notebook Server. For example, limited_read, read, and write. A Role can have a single or multiple permissions. The Table 50 describes the Preconfigured Permissions.

Table 50: Notebook Server Permissions

Name	Description
*	Do all of the following names
create_notebook	Create a notebook
delete_all	Delete all notebooks in the workspace view
export_all	Export all notebooks in the Workspace view
graph_create	Create a graph in the Graphs tab
import_notebook	Import a notebook
view_dashboard_tab	View the Tasks tab
view_permissions_tab	View the Permissions tab
view_interpreter_tab	View the Interpreters tab
view_credentials_tab	View the Credentials tab
create_credential	Create a credential
view_visualization_template_tab	View the Visualization Templates tab
visualization_template_create	Create a visualization template
graph_delete	Delete a graph
graph_share	Share a graph
graph_update	Update a graph
graph_view	View a graph
interpreter_create_variant	Create a new interpreter variant
interpreter_update_variant	Update a variant of an interpreter
interpreter_view	View an interpreter
interpreter_variant_execute	Execute an interpreter variant

Table 50: Notebook Server Permissions

interpreter_variant_delete	Delete an interpreter variant
interpreter_variant_view	View an interpreter variant
job_cancel	Cancel a job
job_view	View a job
add_relation	Add a relation to a notebook
Attach	(Deprecated) Attach a notebook
Clear	Clear all results in a notebook
Clone	Clone a notebook
Delete	Delete a notebook
Detach	(Deprecated) Detach a notebook
Export	Export a notebook
Iframe	Open a notebook in the iframe view
invalidate_session	Invalidate the session of a notebook
Layout	Change the layout of a notebook
paragraph_comment	Comment on paragraphs in a notebook
paragraph_create	Create a new paragraph in a notebook
paragraph_delete	Delete the paragraphs in a notebook
paragraph_execute	Execute the paragraphs in a notebook
paragraph_modify	Modify the paragraphs in a notebook
paragraph_move	Move the paragraphs in a notebook
paragraph_view	View the paragraphs in a notebook
remove_relation	Remove a relation from a notebook
Rename	Rename a notebook
run_all	Run all paragraphs in a notebook
schedule_notebook	Schedule a notebook
Share	Share a notebook
set_readonly	Set a notebook to read-only
Snapshot	Take a snapshot (immutable copy) of a notebook
Style	Change the style of a notebook
Template	Add a template to a notebook
toggle_show_code	Toggle the Show Code button in a notebook
toggle_show_result	Toggle the Show Result button in a notebook
Update	Update a notebook
View	View a notebook
view_code	View the code of the paragraphs of a notebook
view_result	View the result of the paragraphs in a notebook
view_sessions	View the sessions of a notebook
create_group	Create a group
create_permission_template	Create a permission template
create_role	Create a role
delete_group	Delete a group
delete_permission_template	Delete a permission template

Table 50: Notebook Server Permissions

delete_role	Delete a role
update_group	Update a group
update_permission_template	Update a permission template
update_role	Update a role
update_user	Update a user
view_group	View the Groups section in the Permissions screen
view_permission_template	View the Permission Templates section in the Permissions screen
view_role	View the Roles section in the Permissions screen
view_user	View the Users section in the Permissions screen
view_credential	View a credential and download its file in the credentials screen
use_credential	Use a credential to connect to a data store
delete_credential	Delete a credential from the credentials screen
visualization_template_view	View a visualization template
visualization_template_update	Update a visualization template
visualization_template_delete	Delete a visualization template
visualization_template_share	Share a visualization template
templates_view	View the templates Menu
review_approve (deprecated)	Users can approve the manual similarity edge
review_request (deprecated)	Users can request for approving manual similarity edge
Approve	Users can approve scenario notebook

9.4.4 Group - Role Mapping

Table 51 describes the Preconfigured Groups and the corresponding Roles.

Table 51: Role Mapping

Group Code	Group Name	Role Code	Role Name
DSREDACTGRP	DSREDACTGRP	DSREDACT	DSREDACT
DSUSRGRP	Datastudio User	DSADMIN	DSADMIN
IDNTYADMN	Identity Administrator group	ADMIN_LINK	Admin Link Role
		BATCH_ADV	Batch Advance Role
		BATCH_WRITE	Batch Write Role
		FUNC_ADV	Function Advanced Role
		GRP_ADV	Group Advanced Role
		ROLE_ADV	Role Advanced Role
		USR_ADV	User Advanced Role

Table 51: Role Mapping

Group Code	Group Name	Role Code	Role Name
IDNTYAUTH	Identity Authorizer group	ADMIN_LINK	Admin Link Role
		FUNC_READ	Function Read Role
		GRP_AUTH	Group Authorize Role
		GRP_READ	Group Read Role
		ROLE_AUTH	Role Authorize Role
		ROLE_READ	Role Read Role
		USR_AUTH	User Authorize Role
MDLAPPR	Modeling Approver	DSAPPROVER	DSAPPROVER
		DSINTER	DSINTER
		MDLACCESS	Model Access
		MDLAUTH	Model Authorize
		MDLDEPLOY	Model Deployment
		MDLREAD	Model Read
		WKSPACC	Workspace Access
		WKSPREAD	Workspace Read
MDLBATCHUSR	Modeling Batch User	DSBATCH	DSBATCH
MDLREV	Modeling Reviewer	DSUSER	DSUSER
		MDLACCESS	Model Access
		MDLREAD	Model Read
		MDLREVIEW	Model Review
		WKSPACC	Workspace Access
		WKSPREAD	Workspace Read

Table 51: Role Mapping

Group Code	Group Name	Role Code	Role Name
MDLUSR	Modeling User	BATCH_ADV	Batch Advance Role
		DSACCESS	DataStore Access
		DSREAD	DataStore Read
		DSUSER	DSUSER
		DSWRITE	DataStore Write
		MDLACCESS	Model Access
		MDLADV	Model Advanced
		MDLEXE	Model Execute
		MDLREAD	Model Read
		MDLWRITE	Model Write
		WKSPACC	Workspace Access
		WKSPREAD	Workspace Read
WKSPADMIN	Workspace Administrator	DSADMIN	DSADMIN
		IDMGMTADVN	Identity MGMT advanced
		WKSPACC	Workspace Access
		WKSPAUTH	Workspace Authorize
		WKSPREAD	Workspace Read
		WKSPWRITE	Workspace Write
GRPADMIN	Graph Administrator	GRPEXE	Graph Execute
		GRPREAD	Graph Read
		GRPSUMM	Graph Access
		GRPWRITE	Graph Write
GRPUSR	Graph User	GRPEXE	Graph Execute
		GRPREAD	Graph Read
		GRPSUMM	Graph Access
		GRPWRITE	Graph Write

9.4.5 Role - Function Mapping

Table 52 describes the pre-configured roles and the corresponding Functions.

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
ADMIN_LINK	Admin Link Role	ADMIN_LINK	Admin Link
APP_READ	Application Read Role	APP_SUMM	Application Summary
		APP_VIEW	Application View
BATCH_ADV	Batch Advance Role	BATCH_ADD	Batch Add Function
		BATCH_COPY	Batch Copy Function
		BATCH_DEL	Batch Delete Function
		BATCH_EXEC	Batch Execute Function
		BATCH_MOD	Batch Modify Function
		BATCH_PURGE	Batch Purge Function
		BATCH_SCH	Batch Schedule Function
		BATCH_SUMM	Batch Summary Function
		BATCH_VIEW	Batch View Function
		FUNC_SUMM	Function Summary
BATCH_AUTH	Batch Authorization Role	BATCH_AUTH	Batch Authorize Function
		BATCH_SUMM	Batch Summary Function
		BATCH_VIEW	Batch View Function
		FUNC_SUMM	Function Summary
BATCH_MAINT	Batch Maintenance Role	BATCH_MOD	Batch Modify Function
		BATCH_SUMM	Batch Summary Function
		BATCH_VIEW	Batch View Function
		FUNC_SUMM	Function Summary

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
BATCH_OPER	Bath Operation Role	BATCH_EXEC	Batch Execute Function
		BATCH_SCH	Batch Schedule Function
		BATCH_SUMM	Batch Summary Function
		BATCH_VIEW	Batch View Function
		FUNC_SUMM	Function Summary
BATCH_READ	Batch Read Role	BATCH_SUMM	Batch Summary Function
		BATCH_VIEW	Batch View Function
		FUNC_SUMM	Function Summary
BATCH_WRITE	Batch Write Role	BATCH_ADD	Batch Add Function
		BATCH_COPY	Batch Copy Function
		BATCH_MOD	Batch Modify Function
		BATCH_SUMM	Batch Summary Function
		BATCH_VIEW	Batch View Function
		FUNC_SUMM	Function Summary
DSACCESS	DataStore Access	DSSUMM	DataStore Access
DSAPPROVER	DSAPPROVER	MDAPPROVER	MDAPPROVER
DSREAD	DataStore Read	DSVIEW	DataStore View
DSUSER	DSUSER	MDREQUESTER	MDREQUESTER
DSWRITE	DataStore Write	DSADD	DataStore Add
		DSDELETE	DataStore Delete
		DSEDIT	DataStore Edit
DTSRC_READ	DataStore Read Role	DTSRC_SUMM	DataStore Summary
		DTSRC_VIEW	DataStore View
FLDR_ADV	Folder Advanced Role	FLDR_ADD	Folder Add
		FLDR_DEL	Folder Delete
		FLDR_MOD	Folder Modify
		FLDR_SUMM	Folder Summary
		FLDR_VIEW	Folder View

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
FLDR_READ	Folder Read Role	FLDR_SUMM	Folder Summary
		FLDR_VIEW	Folder View
FLDR_WRITE	Folder Write Role	FLDR_ADD	Folder Add
		FLDR_MOD	Folder Modify
		FLDR_SUMM	Folder Summary
		FLDR_VIEW	Folder View
FLDRACC	Folder Access	FLDR_LNK_ACC	Folder Link Access
FLDRAUTH	Folder Authorize	FLDR_AUTH	Folder Authorization
FLDRREAD	Folder Read	FLDR_VIW	Folder View
FLDRWRITE	Folder Write	FLDR_CPY	Folder Copy
		FLDR_EDIT	Folder Edit
FUNC_ADV	Function Advanced Role	FUNC_ADD	Function Add
		FUNC_DEL	Function Delete
		FUNC_MAP	Function Map
		FUNC_MOD	Function Modify
		FUNC_PURGE	Function Purge
		FUNC_SUMM	Function Summary
		FUNC_VIEW	Function View
FUNC_READ	Function Read Role	FUNC_SUMM	Function Summary
		FUNC_VIEW	Function View
FUNC_WRITE	Function Write Role	FUNC_ADD	Function Add
		FUNC_MOD	Function Modify
		FUNC_SUMM	Function Summary
		FUNC_VIEW	Function View
GRP_ADV	Group Advanced Role	GRP_ADD	Group Add
		GRP_DEL	Group Delete
		GRP_MAP	Group Map
		GRP_MOD	Group Modify
		GRP_PURGE	Group Purge
		GRP_SUMM	Group Summary
		GRP_VIEW	Group View

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
GRP_AUTH	Group Authorize Role	GRP_AUTH	Group Authorize
		GRP_SUMM	Group Summary
		GRP_VIEW	Group View
GRP_READ	Group Read Role	GRP_SUMM	Group Summary
		GRP_VIEW	Group View
GRP_WRITE	Group Write Role	GRP_ADD	Group Add
		GRP_MOD	Group Modify
		GRP_SUMM	Group Summary
		GRP_VIEW	Group View
IDMGMTACC	Identity MGMT access	ADMINSCR	Administration Screen
IDMGMTADVN	Identity MGMT advanced	ADMINSCR	Administration Screen
		FUNCMAINT	Function Maintenance Screen
		FUNCROLE	Function Role Map Screen
		ROLEMAINT	Role Maintenance Screen
		UGFLROLMAP	User Group Folder Role Map Screen
		UGMAINT	User Group Maintenance Screen
		UGMAP	User Group User Map Screen
		UGROLMAP	User Group Role Map Screen
		UGWKSPMAP	User Group Workspace Map Screen
		USRACTREP	User Activity Reports Screen
		USRATTUP	User Attribute Upload Screen
		USRMAINT	User Maintenance Screen
IDMGMTAUTH	Identity MGMT authorize	ADMINSCR	Administration Screen
		USRATH	User Authorization Screen
IDMGMTREAD	Identity MGMT read	ADMINSCR	Administration Screen

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
IDMGMTWRIT	Identity MGMT write	ADMINSCR	Administration Screen
		ROLEMAINT	Role Maintenance Screen
		UGFLROLMAP	User Group Folder Role Map Screen
		UGMAINT	User Group Maintenance Screen
		UGMAP	User Group User Map Screen
		UGROLMAP	User Group Role Map Screen
		UGWKSPMAP	User Group Workspace Map Screen
		USRACTREP	User Activity Reports Screen
		USRATTUP	User Attribute Upload Screen
		USRMAINT	User Maintenance Screen
MDLACCESS	Model Access	MDLCNFSUMM	Model Configuration Summary
		MDLSUMM	Model Summary
MDLADMIN	Model Admin	MDLPURGE	Model Purge
MDLADV	Model Advanced	MDLEXE	Model Execute
		MDLLOCK	Model Lock
MDLAUTH	Model Authorize	MDLAPPROVE	Model Approve
MDLDEPLOY	Model Deployment	MDLDEPL	Model Deploy
MDLREAD	Model Read	MDLTRACE	Model Trace
		MDLVIEW	Model View
MDLREVIEW	Model Review	MDLREVIEW	Model Review
MDLWRITE	Model Write	MDLADD	Model Add
		MDLCOPY	Model Copy
		MDLDEL	Model Delete
		MDLEDIT	Model Edit

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
ROLE_ADV	Role Advanced Role	ROLE_ADD	Role Add
		ROLE_DEL	Role Delete
		ROLE_MAP	Role Map
		ROLE_MOD	Role Modify
		ROLE_PURGE	Role Purge
		ROLE_SUMM	Role Summary
		ROLE_VIEW	Role View
ROLE_AUTH	Role Authorize Role	ROLE_AUTH	Role Authorize
		ROLE_SUMM	Role Summary
		ROLE_VIEW	Role View
ROLE_READ	Role Read Role	ROLE_SUMM	Role Summary
		ROLE_VIEW	Role View
ROLE_WRITE	Role Write Role	ROLE_ADD	Role Add
		ROLE_MOD	Role Modify
		ROLE_SUMM	Role Summary
		ROLE_VIEW	Role View
SBADMIN	Sandbox Admin	SBADD	Sandbox Add
SRVC_READ	Service Read Role	SRVC_SUMM	Service Summary
		SRVC_VIEW	Service View
USR_ADV	User Advanced Role	USR_ADD	User Add
		USR_DEL	User Delete
		USR_MAP	User Map
		USR_MOD	User Modify
		USR_PURGE	User Purge
		USR_SUMM	User Summary
		USR_VIEW	User View
USR_AUTH	User Authorize Role	USR_AUTH	User Authorize
		USR_SUMM	User Summary
		USR_VIEW	User View
USR_READ	User Read Role	USR_SUMM	User Summary
		USR_VIEW	User View

Table 52: Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
USR_WRITE	User Write Role	USR_ADD	User Add
		USR_MOD	User Modify
		USR_SUMM	User Summary
		USR_VIEW	User View
WKSPACC	Workspace Access	WKSP_LNK_ACC	Workspace Link Access
		WKSP_SUMM	Workspace Summary Access
WKSPAUTH	Workspace Authorize	WKSP_AUTH	Workspace Authorization
WKSPREAD	Workspace Read	WKSP_VIW	Workspace View
WKSPWRITE	Workspace Write	WKSP_ADD	Workspace Add
		WKSP_CPY	Workspace Copy
		WKSP_DEL	Workspace Delete
		WKSP_EDIT	Workspace Edit
WRKSP_ADV	Workspace Advanced Role	WRKSP_ADD	Workspace Add
		WRKSP_DEL	Workspace Delete
		WRKSP_MOD	Workspace Modify
		WRKSP_SUMM	Workspace Summary
		WRKSP_VIEW	Workspace View
WRKSP_READ	Workspace Read Role	WRKSP_SUMM	Workspace Summary
		WRKSP_VIEW	Workspace View
WRKSP_WRITE	Workspace Write Role	WRKSP_ADD	Workspace Add
		WRKSP_MOD	Workspace Modify
		WRKSP_SUMM	Workspace Summary
		WRKSP_VIEW	Workspace View

9.4.6 Role - Permission Mapping

Table 53 describes the Preconfigured Roles and the corresponding Permissions.

> **NOTE** The role **DSREDACTGRP** is used for applying redaction in the graph.

Table 53: Role - Permission Mapping

Permissions	Roles						
	DSADMIN	DSBATCH	DSINTER	DSUSER	DSAP-PRROVER	MDAP-PROVER	MDREQUESTOR
*	Yes						
create_notebook	Yes	Yes	Yes	Yes			
delete_all	Yes	Yes	Yes				
export_all	Yes	Yes	Yes				
graph_create	Yes	Yes	Yes	Yes			
import_notebook	Yes	Yes	Yes	Yes			
view_dashboard_tab	Yes	Yes	Yes	Yes			
view_permissions_tab	Yes		Yes				
view_interpreter_tab	Yes	Yes	Yes	Yes			
view_credentials_tab	Yes	Yes	Yes				
create_credential	Yes	Yes	Yes				
view_visualization_template_tab	Yes	Yes	Yes	Yes			
visualization_template_create	Yes	Yes	Yes	Yes			
graph_delete	Yes	Yes					
graph_share	Yes	Yes					
graph_update	Yes	Yes					
graph_view	Yes	Yes					
interpreter_create_variant	Yes		Yes				
interpreter_update_variant	Yes		Yes				
interpreter_view	Yes	Yes	Yes	Yes			
interpreter_variant_execute	Yes	Yes	Yes	Yes			
interpreter_variant_delete	Yes		Yes				
interpreter_variant_view	Yes	Yes	Yes	Yes			

Table 53: Role - Permission Mapping

job_cancel	Yes	Yes					
job_view	Yes	Yes	Yes	Yes			
add_relation	Yes	Yes	Yes	Yes			
Attach	Yes						
Clear	Yes	Yes	Yes	Yes			
Clone	Yes	Yes	Yes	Yes			
Delete	Yes	Yes	Yes	Yes			
Detach	Yes						
Export	Yes	Yes	Yes	Yes			
Iframe	Yes	Yes	Yes	Yes			
invalidate_session	Yes	Yes	Yes	Yes			
Layout	Yes	Yes	Yes	Yes			
paragraph_comment	Yes	Yes	Yes	Yes			
paragraph_create	Yes	Yes	Yes	Yes			
paragraph_delete	Yes	Yes	Yes	Yes			
paragraph_execute	Yes	Yes	Yes	Yes			
paragraph_modify	Yes	Yes	Yes	Yes			
paragraph_move	Yes	Yes	Yes	Yes			
paragraph_view	Yes	Yes	Yes	Yes			
remove_relation	Yes	Yes	Yes	Yes			
Rename	Yes	Yes	Yes	Yes			
run_all	Yes	Yes	Yes	Yes			
schedule_notebook	Yes	Yes					
Share	Yes	Yes	Yes	Yes			
set_readonly	Yes	Yes	Yes	Yes			
Snapshot	Yes	Yes	Yes	Yes			
Style	Yes	Yes	Yes	Yes			
Template	Yes	Yes	Yes	Yes			
toggle_show_code	Yes	Yes	Yes	Yes			
toggle_show_result	Yes	Yes	Yes	Yes			
Update	Yes	Yes	Yes	Yes			
View	Yes	Yes	Yes	Yes			
view_code	Yes	Yes	Yes	Yes			

Table 53: Role - Permission Mapping

view_result	Yes	Yes	Yes	Yes			
view_sessions	Yes	Yes	Yes	Yes			
create_group	Yes		Yes				
create_permission_template	Yes		Yes				
create_role	Yes		Yes				
delete_group	Yes		Yes				
delete_permission_template	Yes		Yes				
delete_role	Yes		Yes				
update_group	Yes		Yes				
update_permission_template	Yes		Yes				
update_role	Yes		Yes				
update_user	Yes		Yes				
view_group	Yes		Yes				
view_permission_template	Yes		Yes				
view_role	Yes		Yes				
view_user	Yes		Yes				
view_credential	Yes		Yes				
use_credential	Yes		Yes				
delete_credential	Yes		Yes				
visualization_template_view	Yes	Yes	Yes	Yes			
visualization_template_update	Yes	Yes	Yes	Yes			
visualization_template_delete	Yes	Yes	Yes	Yes			
visualization_template_share	Yes	Yes	Yes	Yes			
Approve	Yes	Yes					
review_request	Yes			Yes			
review_approve	Yes				Yes		
MDAPPROVE						Yes	
MDREQUEST							Yes

9.5 Setting Memory of Entity Resolution and Matching Services

To increase the memory of entity resolution and matching services, perform the following steps:

1. Log in to the server where Compliance Studio is installed.
2. Navigate to <COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin directory.
3. Open the compliance-studio.sh file, and edit the function start_services()
4. In entity resolution, update the memory in the JAVA_OPTS to a higher value according to your requirement.

   For example,

   ```
   export JAVA_OPTS="-Xms12g -Xmx24g"
   ```

   Code-block:

   ```
   entity-resolution
     export JAVA_OPTS="-Xms4g -Xmx8g"
       export LD_LIBRARY_PATH="$COMPLIANCE_STUDIO_INSTALLATION_PATH/
   deployed/python-packages/saneVirtualEnv/lib/python3.6/site-packages/
   jep:$COMPLIANCE_STUDIO_INSTALLATION_PATH/deployed/python-packages/
   saneVirtualEnv/lib/":$LD_LIBRARY_PATH
       export PATH_ORG=$PATH
       export PATH=$DEPLOY_APP_HOME/python-packages/saneVirtualEnv/
   bin:$PATH
       export TNS_ADMIN=$TNS_ADMIN_PATH
       sh "$DEPLOY_APP_HOME"/entity-resolution/bin/entity-resolution
   >"$LOGS_FOLDER"/entity-resolution.log &
       unset JAVA_OPTS
       export PATH=$PATH_ORG
       ;;
   ```

5. In the matching service, update the memory in the **JAVA_OPTS** to a higher value according to your requirement.

   For example,

   ```
   export JAVA_OPTS="-Xms12g -Xmx24g"
   ```

   Code-block:

   ```
   matching-service
       export JAVA_OPTS="-Xms6g -Xmx12g"
       export LD_LIBRARY_PATH="$COMPLIANCE_STUDIO_INSTALLATION_PATH/
   deployed/python-packages/saneVirtualEnv/lib/python3.6/site-packages/
   jep:$COMPLIANCE_STUDIO_INSTALLATION_PATH/deployed/python-packages/
   saneVirtualEnv/lib/":$LD_LIBRARY_PATH
       if ("$OPEN_SEARCH_HTTPS_ENABLED"); then
   ```

```
      export JAVA_OPTS="$JAVA_OPTS -
Djavax.net.ssl.trustStore=$DEPLOY_APP_HOME/matching-service/conf/
$OPEN_SEARCH_TRUSTSTORE_FILE_NAME
      -
Djavax.net.ssl.trustStorePassword=$OPEN_SEARCH_TRUSTSTORE_PASSWORD"
    fi
    export PATH_ORG=$PATH
    export PATH=$DEPLOY_APP_HOME/python-packages/saneVirtualEnv/
bin:$PATH
    export TNS_ADMIN=$TNS_ADMIN_PATH
    sh "$DEPLOY_APP_HOME"/matching-service/bin/matching-service
>"$LOGS_FOLDER"/matching-service.log &
    unset JAVA_OPTS
    export PATH=$PATH_ORG
    ;;
```

9.6 Cleanup Steps When the Create Index and Load Data Job Terminated Manually

To perform cleanup for Create Index and Load Data job, follow the step:

1. Execute the following command:

   ```
   nohup ./ER_Cleanup.sh "<Cleanup_type>" "<FIC_MIS_DATE>" "<Current_Run-
   skey>" "<Execution_mode>" "<ERSchemaId>" "<Batch_group>" "<Pipelineid>"
   &
   ```

   For example, 8126 version: `nohup ./ER_Cleanup.sh "CLEANUP-JOB1-INSTANCE" "20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8126" &`

9.7 Cleanup Steps When the Bulk Similarity Job Terminated Manually

To perform cleanup for Bulk Similarity job, follow the step:

1. Execute the following command:

   ```
   nohup ./ER_Cleanup.sh "<Cleanup_type>" "<FIC_MIS_DATE>" "<Current_Run-
   skey>" "<Execution_mode>" "<ERSchemaId>" "<Batch_group>" "<Pipelineid>"
   &
   ```

   For example, 8126 version: `nohup ./ER_Cleanup.sh "CLEANUP-JOB2-INSTANCE" "20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8126" &`

9.8 Cleanup Steps When the Data Survival Job Terminated Manually

To perform cleanup for Data Survival job, follow the step:

1. Execute the following command:

```
nohup ./ER_Cleanup.sh "<Cleanup_type>" "<FIC_MIS_DATE>" "<Current_Run-
skey>" "<Execution_mode>" "<ERSchemaId>" "<Batch_group>" "<Pipelineid>"
&
```

For example, 8126 version: `nohup ./ER_Cleanup.sh "CLEANUP-JOB3-INSTANCE" "20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8126" &`

9.9 Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually

To perform cleanup for Load Data in the FCC_ER_OUTPUT job, follow the step:

1. Execute the following command:

```
nohup ./ER_Cleanup.sh "<Cleanup_type>" "<FIC_MIS_DATE>" "<Current_Run-
skey>" "<Execution_mode>" "<ERSchemaId>" "<Batch_group>" "<Pipelineid>"
&
```

For example, 8126 version: `nohup ./ER_Cleanup.sh "CLEANUP-JOB4-INSTANCE" "20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8126" &`

9.10 Resetting Entity Resolution Back to Day 0

ATTENTION

- This section is applicable only when you wipe out ER-related tables and indexes. This will bring the Entity Resolution back to **Day0**.
- You can clean up the ER Schema after upgrading from **v8.1.2.0.0** to **v8.1.2.0.1** or restart ER with different rules.
- If FCC_BATCH_RUN is empty, you have to reset the ER to Day 0 and then runskey should be 0.

To perform cleanup for full reset to day 0, follow the step:

1. Execute the following command:

```
nohup ./ER_Cleanup.sh "<Cleanup_type>" "<FIC_MIS_DATE>" "<Current_Run-
skey>" "<Execution_mode>" "<ERSchemaId>" "<Batch_group>" "<Pipelineid>"
&
```

For example, 8126 version: `nohup ./ER_Cleanup.sh "RESET-TO-DAY0" "20151210" "182" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8126" &`

9.10.1 Compliance Studio Schema Changes

To truncate batch run tables, perform the following:

1. Log in to Compliance Studio Schema.
2. Check the **FCC_BATCH_RUN** table in the Compliance Studio schema and if there are any records exist, run the following command to truncate the table before executing the ER jobs:

```
truncate table FCC_BATCH_RUN;
```

9.10.2 OpenSearch Changes

To clean up ER staging indexes, perform the following:

1. Log in to the server where Compliance Studio is installed.
2. Run the following **curl** command:

   ```
   curl -XDELETE http://hostname:port/load-to-open-search/idx/deleteIndex/
   <Index name>
   ```

   For example,

   ```
   curl -XDELETE http://testserver.oracle.com:7053/load-to-open-search/idx/
   deleteIndex/stg_party_812
   ```

3. Repeat **Step 2** if multiple ER indexes, run with respective staging index names.

9.11 Utility Scripts

9.11.1 Data Slicing Utility Script

The Data Slicing Utility is a SQL script to perform data slicing (slicing the data into different chunks or data units) according to the user input (FIC_MIS_DATE). It helps faster turn-around time for individual batches as the load is moderately low.

FIC_MIS_DATE is the execution identifier for Entity Resolution, and it is easy to distribute records into different FIC_MIS_DATE values.

You can perform the data slicing for a high volume of data, which takes a long time and more resource based on your database performance.

> **NOTE** This utility is used for slicing the data in the following input tables of the out-of-the-box rules for Entity Resolution:
>
> - STG_PARTY_MASTER_PRE
> - STG_PARTY_DETAILS_PRE
> - STG_PARTY_EMAIL_MAP_PRE
> - STG_PARTY_PHONE_MAP_PRE
> - STG_CUSTOMER_IDENTIFCTN_DOC_PRE
> - STG_PARTY_ADDRESS_MAP_PRE
> - STG_ADDRESS_MASTER_PRE

The utility distributes the data into logical units based on the criteria (user input), resulting in multiple data chunks.

- It accepts comma-separated **FIC_MIS_DATE** as user input.

  For example. 20150101,20150102,20150103

- It distributes the records across the **FIC_MIS_DATE** equally. The last slice should contain additional records if there are any.

> **ATTENTION** It is recommended that you must split the data into slices of a maximum of 10 million records.
>
> Here is a scenario of data slicing:
>
> - Input data volume: **50 million**
> - Size of slice on which job has to execute: **10 million**
> - Total number of slices: **5** (different comma-separated **FIC_MIS_DATE)**

After the utility completes the distribution, you can perform the ER batch execution as follows:

1. Provide the chunk as **Day 0** load corresponding to the respective **FIC_MIS_DATE**.
2. Provide subsequent chunks such as **Day 1**, **Day 2**, etc. These chunks are treated as delta loads (delta having only new records).

To execute the utility script, perform the following:

1. Obtain the script from path `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/ficdb/Utilities/DataSlicingUtility/DataSlicingUtility.sql`
2. Log in to the ER Schema. The schema (input tables of Entity Resolution) is available.
3. Copy the script to the machine where you need to execute the script.
4. Run the following command in SQL prompt:

   ```
   @<Fully Qualified path of Utility Script>/DataSlicingUtility.sql
   ```

5. Enter the values according to the following prompt:

   ```
   Enter value for fic_mis_date:
   ```

   You need to enter comma-separated **FIC_MIS_DATE** in **YYYYMMDD** format.

   For example, 20150101,20150102,20150103

6. Press **Enter**.
   - On successful execution, the utility scripts exits with a success message "FIC_MIS_DATEs have applied for all <list of fic_mis_dates> slices"

     For example,

     ```
     SQL> @<path of the script>/DataSlicingUtility.sql

     Enter value for fic_mis_date:
     20150107,20150108,20150109,20150110,20150115

     old  24: FIC_MIS_DATES:='&FIC_MIS_DATE';

     new  24:
     FIC_MIS_DATES:='20150107,20150108,20150109,20150110,20150115';

     PL/SQL procedure successfully completed.
     ```

   - On failure, displays the appropriate error message.
7. You can validate the results of successful execution:
   - For each input table, check the count of records against **FIC_MIS_DATE**.

Run the following commands to check the count in each input table. Perform the same for all input tables:

```
SELECT DISTINCT FIC_MIS_DATE, COUNT(*) FROM <INPUT TABLE NAME> GROUP BY FIC_MIS_DATE;
```

For example,

```
SELECT DISTINCT FIC_MIS_DATE, COUNT(*) FROM STG_PARTY_MASTER_PRE GROUP BY FIC_MIS_DATE;
```

- Ensure that complete information for a particular party is included in the same slice.

a. For example, for any **V_PARTY_ID**, there should be the same **FIC_MIS_DATE** tagged in each input table.

9.12 Load Data into ICIJ Tables

After installing the Compliance Studio, you need to run the script. For more details, **Importing OOB Graph Definition and related Metadata** section in the OFS Compliance Studio Installation Guide.

The data pipeline does not currently support loading data directly from CSV files.

The following source tables are created during the Post Installation procedure.

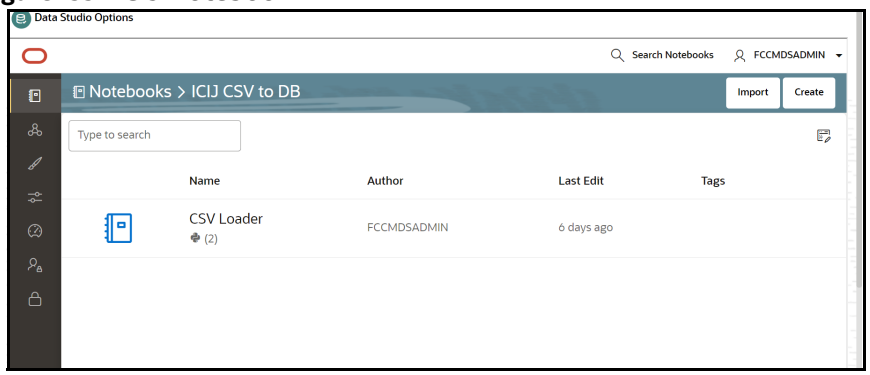
- ICIJ_NODES_ENTITY
- ICIJ_NODES_INTERMEDIARY
- ICIJ_NODES_OFFICER
- ICIJ_NODES_OTHERS
- ICIJ_NODES_ADDRESS
- ICIJ_RELATIONSHIP

To create ICIJ tables, perform the following:

1. Download zip file from the ICIJ's website and copy the downloaded files to the local server.
2. Log in to the Compliance Studio application.
3. Navigate to the Compliance Studio server with the same URL by changing the port to 7008. (`http://<hostname>:7008` from `http://<hostname>:7001/cs/`)

   The ICIJ Notebook is part of a built-in notebook, as shown below.

Figure 155: ICIJ Notebook



4. Open the Notebook, **ICIJ CSV to DB/CSV Loader.**

5. Click **Export Notebook** to download the notebook.

   The notebook is saved in the local machine.
6. Navigate to the **Modeling** drop-down list and select **Pipelines**.
7. Click **Add** and select **Objective** from the list to display the **Objective Details** dialog box.
8. Enter details in the Objective **Name** and **Description** fields in the Add **Objective** dialog box.
9. Click **Save**.

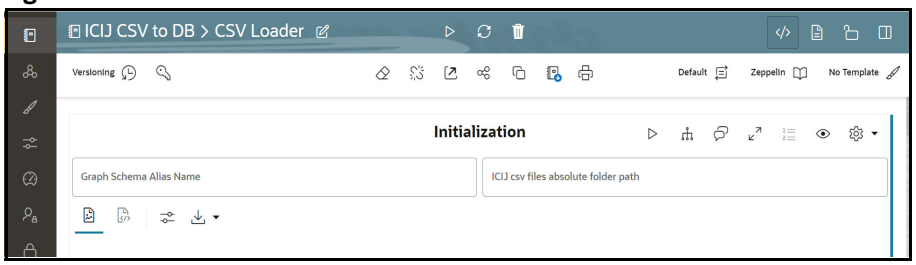
   For more information on objective, see the **Creating Objective (Folders)** section in the OFS Compliance Studio User Guide.
10. Click **Add** and select **Draft** from the list to display the **Add Draft** dialog box.

    Create New Model is the default setting in the **Model Details** dialog box.

    NOTE The draft should be created inside the objective folder.
11. Drag the toggle switch to select **Import Dump**.
12. Drag and drop the file into the **Import Dump File** field or click in the box to open the file selector dialog and select a file.
13. Click **Import**.
14. Enter the details for the **Draft Name** and **Description.**
15. Enter a tag in the **Tags** field.
16. Click **Import**. A new model is created by importing the model data dump.

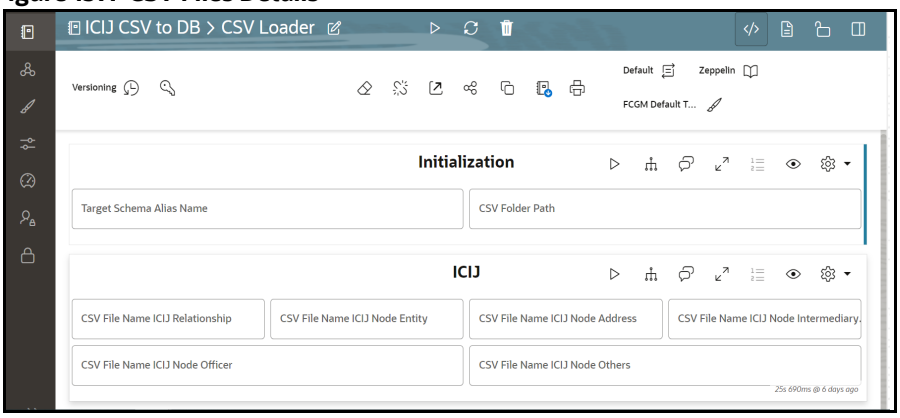
    For more information on importing workspace models, see the **Import a Workspace Model Data into a New Model** section in the OFS Compliance Studio User Guide.
17. Ensure that the SQL loader (`sqlldr`) is running in the Compliance Studio.
18. Enter the **Target Schema Alias Name** and the **ICIJ CSV Folder Path** and click ▷ to run the paragraph.

Figure 156: Initialization Field Details



19. Fill the names of CSV files in the required fields in each ICIJ source type. Ensure the name of the file is added with the **.csv** extension.

Figure 157: CSV Files Details



20. Click  to run the paragraphs for ICIJ source. You can simultaneously enter all the filenames and run the paragraph for all source files.

    On successful execution, the data will be loaded into ICIJ tables.

> **NOTE** The Notebook is accessible only by the Administrators.

9.13 Prescript Condition

The **Persist** of the Data pipeline of the corresponding node/edge should be defined with the following prescript:

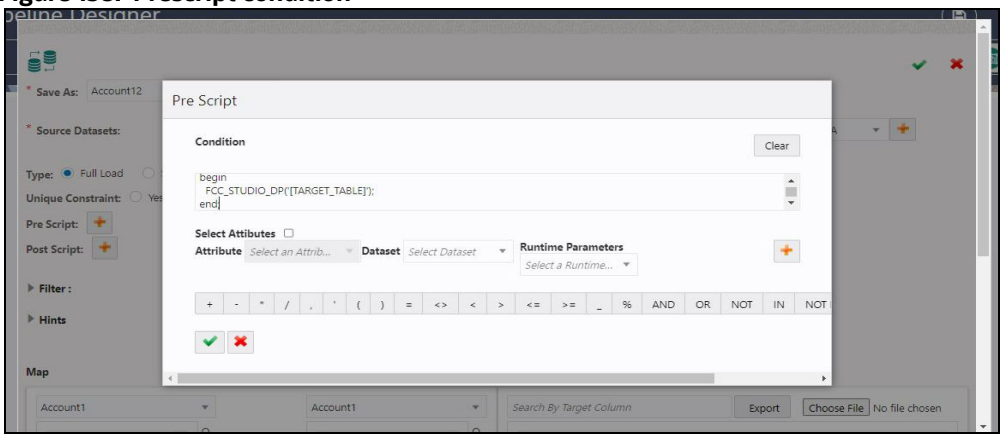
```
begin

FCC_STUDIO_DP('[TARGET_TABLE]');

end;
```

The following figure illustrates the Persist to add the Prescript condition.

Figure 158: Prescript condition



For more details on the Data pipeline, see **Managing Data Pipeline** section in the OFS Compliance Studio User Guide.

9.14 Resetting Graph Pipeline Back to Day 0

To reset the graph pipeline to Day0 batch, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/GraphPipeline-Cleanup-Scripts` directory.
2. Perform the steps provided in the `README.md` file.
3. Execute the following command:

```
BEGIN
    FOR rec IN (
        SELECT
            index_name
        FROM
            user_indexes
        WHERE
            status = 'UNUSABLE'    ) LOOP
        EXECUTE IMMEDIATE 'ALTER INDEX ' || rec.index_name || '
REBUILD';
        dbms_output.put_line(rec.index_name || ' index rebuilt');
    END LOOP;
END;
```

4. Restart PGX server.

9.15 Disable the User in Compliance Studio after SSO Login

To revoke the mapped CS Groups for a particular user in the Compliance Studio, follow these steps:

In SAML IDCS, Admin has to remove the Groups for a particular user.

1. Login to IDCS as **Admin**.
2. Navigate to **Users** tab and select the **User**.
3. Navigate to **Groups** tab and select the groups to be revoked.
4. Click **Revoke** Button.
5. Click **Save** to modify the changes.

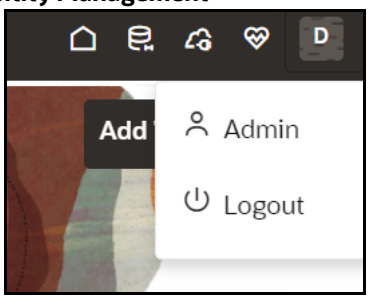
In Compliance Studio,

1. Login to Compliance Studio as **Admin User**.

> **NOTE** Admin users should have access to Identity Management.

2. Navigate to **Identity Management** and click **Users**.

Figure 159: Identity Management



3. Select the same user of the Groups that are removed from the IDCS.
4. Navigate to **Mapped Groups** tab and select the Groups to be revoked.
5. Click **Unmap**.
6. Login as another **Admin User** who can authorize the above changes.

> **NOTE** Any other user with admin access can authorize.

7. Navigate to Identity Management as **Authorizing User.**
8. Click **Users** and select the same user of the Groups that are removed from the IDCS.
9. Navigate to **Mapped Groups** tab and move the toggle switch to the right to enable **Authorization View**.
10. Select all the groups and click **Authorize** button.
11. Restart the Compliance Studio.

9.16 Migrating the Data from ElasticSearch to OpenSearch

Prerequisites:

- OpenSearch should be installed successfully and that service should be up and running.
- Wallet should be configured with Entity Resolution details.

  To configure OpenSearch, see **Configure the OpenSearch Component** section in the OFS Compliance Studio Installation Guide.
- Execute the following command for health check API of the OpenSearch:

  ```
  curl -X GET '<OPENSEARCH_CLUSTER_HOST>:<PORT_NUMBER>/_cat/health'
  ```

  Or

  ```
  curl -X GET '<OPENSEARCH_CLUSTER_HOST>:<PORT_NUMBER>/_cat/health?v'
  ```

  Sample output:

  ```
  1675934006 09:13:26 <OPENSEARCH_CLUSTER_NAME> green 1 1 true 0 0 0 0 0 0
  - 100.0%
  ```
- To verify the health check API in the browser, navigate to the following URL:

```
https://<OPENSEARCH_CLUSTER_HOST>:<PORT_NUMBER>/_cat/health?v
```

> **NOTE** If https is not configured then use the following URL:
>
> ```
> http://<OPENSEARCH_CLUSTER_HOST>:<PORT_NUMBER>/_cat/
> health?v
> ```

To migrate data from ElasticSearch to OpenSearch, see OpenSearch documentation.

Migrating data for 'csa_stg_party_812' from ElasticSearch to OpenSearch, follow these steps:

1. Use the following curl command to load index 'csa_stg_party_812':

> **NOTE** The following parameters to be configured as follows:
>
> - <SCHEMA-NAME> to be replaced with ER schema configured in the wallet.
> - <load_to_opensearch_service_port_number> to be replaced with default value 7053.
> - <FQDN_Compliance_Studio> to be replaced with fully qualified domain name of the Compliance Studio.

```
curl -X POST 'http://
<FQDN_Compliance_Studio>:<load_to_opensearch_service_port_number>/load-
to-open-search/idx/createIndex' \
-H 'Content-Type: application/json' \
-d '{
  "schemaName": "<SCHEMA-NAME>",
  "tableName": "FCC_ER_FULL",
  "filterCondition": "1=1",
  "indexName": "stg_party_812",
  "indexAlias": "csa_812_alias",
  "indexLogicalName": "csa_stg_party_812",
  "indexBusinessName": "csa_stg_party_812",
  "indexKeyAttribute": "original_id",
  "loadType": "FullLoad",
  "shards": 1,
  "replicas": 3,
   "attributes": [
    {
      "name": "address",
      "type": "text",
      "similarity": "boolean",
```

```
      "analyzerType": "address",
      "fields": []
    },
    {
      "name": "business_domain",
      "type": "text",
      "similarity": "boolean",
      "analyzerType": "Organization",
      "fields": []
    },
    {
      "name": "city",
      "type": "text",
      "similarity": "boolean",
      "analyzerType": "address",
      "fields": []
    },
    {
      "name": "country",
      "type": "text",
      "similarity": "boolean",
      "analyzerType": "address",
      "fields": []
    },
    {
      "name": "given_name",
      "type": "text",
      "similarity": "boolean",
      "analyzerType": "namestop",
      "fields": []
    },
  {
      "name": "middle_name",
```

```
  "type": "text",
  "similarity": "boolean",
  "analyzerType": "namestop",
  "fields": []
},
{
  "name": "family_name",
  "type": "text",
  "similarity": "boolean",
  "analyzerType": "namestop",
  "fields": []
},
{
  "name": "concat_name",
  "type": "text",
  "similarity": "boolean",
  "analyzerType": "namestop",
  "fields": []
},
  {
  "name": "alias",
  "type": "text",
  "similarity": "boolean",
  "analyzerType": "namestop",
  "fields": []
},
{
  "name": "state",
  "type": "text",
  "similarity": "boolean",
  "analyzerType": "address",
  "fields": []
}
```

```
  ],
  "customAnalyzer": [],
  "customFilter": [],
  "customCharFilter": [],
  "customTokenizer": [],
  "others": [
  "original_id",
"orgname",
"dob",
"source_name",
"start_date",
"jurisdiction",
"industry",
"naics_code",
"tax_id",
"doc_id",
"email",
"phone",
"postal_code",
"incorporation_date",
"entity_type"
],
  "replaceCharFields": [
  {
    "name": "address",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "city",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
```

```
{
    "name": "country",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "state",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "given_name",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "middle_name",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "family_name",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "concat_name",
    "charArray": [";", "~"],
    "replaceWith": [",", ";"]
},
{
    "name": "alias",
    "charArray": [";", "~"],
```

```
    "replaceWith": [",", ";"]
}
  ],
  "replaceEmptyFields": [],
  "translateFields":
["middle_name","family_name","concat_name","alias","given_name",
"address", "city", "country", "state"]
}'
```

After the successful execution, you will get the following response:

```
{"STATUS":"SUCCESS","MESSAGE":"Index created and loaded
successfully.","COUNT":<count of records loaded>}
```

2. Verify that the index is migrated from elastic search to OpenSearch by navigating the following URL:

   ```
   http://<OPENSEARCH_CLUSTER_HOST>:<PORT_NUMBER>/_cat/indices
   ```

   The sample output is as follows:

   open stg_party_812 E09Y31W_SBiZGlZjbX5zZA 1 3 346 4 521.4kb 521.4kb

9.17 Parameters for Entity Resolution Job execution

This section describes parameters for job execution and cleanup for Entity Resolution.

Table 54 lists parameter for job execution and cleanup for entity resolution.

Table 54: Parameter for Entity Resolution

Parameter	Description	ER Job Execution	Cleanup
Pipeline ID	ER Type has taken as Pipelined ID to execute. For example, CSA_8126.	Yes	Yes
ErSchemaID	The identifier of the schema on which Entity Resolution has to be run. This must be the same as specified in the `resources.xml` file.	Yes	Yes
ErSchemaName	Entity Resolution schema alias name.	Yes	No
MatchType	It processes the records based on the dataset, either Full Load or Delta Load.	Yes	No
LoadType	It can be either FullLoad or DeltaLoad. • **FullLoad**: Clear all the records from the history tables and match all the records based on the **fic_mis_date**. • **DeltaLoad**: Match the modified and new records with the current **fic_mis_date** against all the historical records.	Yes	No

Table 54: Parameter for Entity Resolution

Parameter	Description	ER Job Execution	Cleanup
FIC_MIS_DATE	The date on which the data is entered/loaded in the system in YYYYMMDD format.	Yes	Yes
FSDF VERSION	The version of FSDF for the underlying Stage tables.	Yes	No
Current_batch	The processing group for which batch needs to be run (Only one batch can run at a time for a processing group).	Yes	Yes
Source_batch	Future parameter. You can use the same value as the current batch for now.	Yes	No
Data_origin	Origin of data.	Yes	No
Execution_Mode	It executes the following modes that you want to perform the cleanup. • **Run**: This execution mode displays the list of queries that will be executed under the specified Cleanup_Type. • **Preview**: You can preview the list of queries that will be executed under the specified Cleanup_Type without executing them.	No	Yes
Current_runskey	This indicates the latest runskey on which particular job cleanup is to be performed. In case of resetting ER fully, this is the latest runskey in the FCC_BATCH_RUN run table and this table information is available in the studio schema.	No	Yes
Run_type	If Run_Type as **RUN**, the batch is triggered for the first time for the given FIC_MIS_DATE and Current_Batch. You can re-execute the failed job against the same FIC_MIS_DATE and Current_Batch using Run_Type as **RERUN**.	Yes	No

Table 54: Parameter for Entity Resolution

Parameter	Description	ER Job Execution	Cleanup
Cleanup_type	This indicates which specific ER job type the user wants to perform the cleanup operation. The cleanup types are: • **RESET-TO-DAY0**: This mode type helps to perform full cleanup and reset the ER schema to DAY 0 execution • **CLEANUP-JOB1-INSTANCE**: This mode type helps to perform cleanup when job1 is failed/ manually terminated • **CLEANUP-JOB2-INSTANCE**: This mode type helps to perform cleanup when job2 is failed/ manually terminated • **CLEANUP-JOB3-INSTANCE**: This mode type helps to perform cleanup when job3 is failed/ manually terminated • **CLEANUP-JOB4-INSTANCE**: This mode type helps to perform cleanup when job4 is failed/ manually terminated	No	Yes

9.18 Conda Environment in Notebook

Prior to 8126 environments used 3 different python interpreters, each with pre-defined python versions and libraries, in 8126 this has been replaced with a common python interpreter and multiple conda environments. Now when executing models users can select one of 3 predefined conda environments or can select their own. The recommended conda environments for each model are shown below. Assume we are going to complete this table.

> **NOTE** Users may need to wait 10 to 20 seconds to display the message "Invalidated the session and Initialized the connection" on the Pipeline UI to proceed with notebook execution.

Select the corresponding conda environment while executing model as described in the Table 55.

Table 55: Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Builtin Notebook	
Admin.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE:** There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Admin Notebook.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table 55: Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
AMLES Admin Notebook.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AMLES Data Load.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AMLES Update Event Labels.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AMLES Update Event Scores.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AMLES User Notebook.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AML Event Scoring.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AML Human Trafficking.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AML Scenario Generate Alerts.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AML Shell Scenario.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
ATL Analysis.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table 55: Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Auto-ML Output Tracking.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Auto-ML Output Viewing Using REST.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
BTL Analysis.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Customer Risk Scoring.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Customer Segmentation.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
ICIJ CSV to DB_CSV Loader.dsnb	default_8.1.2.6.0
ML_Address_Matching_Training_Admin.dsnb	sane_8.1.2.6.0
ML_Address_Matching_Training_ETL.dsnb	sane_8.1.2.6.0
ML_Name_Matching_Training_Admin.dsnb	sane_8.1.2.6.0
MLNamematchingTrainingAdminPublish.dsnb	sane_8.1.2.6.0
ML_Name_Matching_Training_ETL.dsnb	sane_8.1.2.6.0
ML_Name_Matching_Training_ETLPublish.dsnb	sane_8.1.2.6.0
Outcome Analysis.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
PreProd Analysis.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Sanctions Admin.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table 55: Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Sanctions EDQ Update.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Sanctions Event Scoring User Notebook.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Scenario Execution.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Annual Ongoing Model Validation.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Create Events.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Data Aggregation in Big Data.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Graph Analytics.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Historic Data Load.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Monthly Ongoing Data Quality Report.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Monthly Ongoing Model Validation.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table 55: Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Supervised ML Scoring Data Load.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML User Notebook.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Transaction Analysis.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Unsupervised ML Historic Data Load.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Unsupervised ML Scoring Data Load.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Unsupervised ML User Notebook.dsnb	Pre-configured with ml4aml_8.1.2.6.0 **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Entity Resolution Notebook	
ER DASHBOARD Data Analysis.dsnb	sane_8.1.2.6.0
ER DASHBOARD Match And Merge Analysis.dsnb	sane_8.1.2.6.0
Scenario Conversion Utility Notebook	
Scenario_Conversion_Utility.dsnb	default_8.1.2.6.0
Scenario_Conversion_Utility_Verification_NB.dsnb	default_8.1.2.6.0
SCU_Set_Calendar.dsnb	default_8.1.2.6.0

9.19 Python Libraries for Predefined Conda Environment

Compliance Studio comes with predefined Conda environments as follows:

- **default_8.1.2.6.0**
- **ml4aml_8.1.2.6.0**
- **sane_8.1.2.6.0**

Table 56 list libraries for default_8.1.2.6.0 conda python (**3.9.17**) environment.

Table 56: Default Conda Python Environment

Package	Version
asttokens	2.2.1
backcall	0.2.0
certifi	2022.12.7
cffi	1.15.1
charset-normalizer	2.0.12
click	8.1.3
cloudpickle	2.2.1
conda-pack	0.6.0
contourpy	1.0.6
cryptography	41.0.1
cx-Oracle	8.3.0
cycler	0.11.0
Cython	0.29.32
dask	2023.6.1
dataclasses	0.6
decorator	5.1.1
distributed	2023.6.1
ds-interpreter-client	23.4.2
evidently	0.1.50.dev0
executing	1.2.0
fonttools	4.38.0
fsspec	2022.3.0
greenlet	1.1.2
hivejdbc	0.2.3
idna	3.3
imbalanced-learn	0.8.1

Table 56: Default Conda Python Environment

Package	Version
importlib-metadata	6.7.0
ipython	8.14.0
jedi	0.18.2
Jinja2	3.1.2
joblib	1.2.0
JPype1	1.3.0
kafka-python	2.0.2
kiwisolver	1.4.4
locket	1.0.0
MarkupSafe	2.1.3
matplotlib	3.6.2
matplotlib-inline	0.1.6
mmg	8.1.2.5.0
modin	0.18.1
msgpack	1.0.5
nltk	3.6.7
numpy	1.24.0
oracle-pypgx-client	23.4.2
oracledb	1.2.2
packaging	21.3
pandas	1.5.3
parso	0.8.3
partd	1.4.0
patsy	0.5.2
pexpect	4.8.0
pickleshare	0.7.5
Pillow	9.3.0
pip	23.2.1
platformdirs	3.8.0
plotly	5.8.0
prompt-toolkit	3.0.38
protobuf	4.23.3

Table 56: Default Conda Python Environment

Package	Version
psutil	5.9.0
ptyprocess	0.7.0
pure-eval	0.2.2
py4j	0.10.9.5
pyarrow	6.0.1
pybars3	0.9.7
pycparser	2.21
pydantic	1.10.5
Pygments	2.15.1
pyjdbc	0.2.2
PyMeta3	0.5.1
pyparsing	2.4.7
python-dateutil	2.8.2
pytz	2022.6
PyYAML	5.4.1
regex	2022.10.31
requests	2.28.2
scikit-learn	1.2.2
scipy	1.10.1
seaborn	0.12.1
setuptools	68.0.0
six	1.16.0
sortedcontainers	2.4.0
SQLAlchemy	2.0.3
sqlparams	3.0.0
stack-data	0.6.2
statsmodels	0.13.5
tblib	2.0.0
tenacity	8.0.1
threadpoolctl	3.1.0
toolz	0.12.0
tornado	6.3.2

Table 56: Default Conda Python Environment

Package	Version
tqdm	4.65.0
traitlets	5.9.0
types-requests	2.31.0.1
types-urllib3	1.26.25.13
typing_extensions	4.4.0
urllib3	1.26.6
wcwidth	0.2.6
wheel	0.41.2
whylabs-client	0.5.2
whylogs	1.2.0
whylogs-sketching	3.4.1.dev3
xgboost	1.5.2
zict	3.0.0
zipp	3.15.0

Table 57 list libraries for ml4aml_8.1.2.6.0 conda python (**3.9.17**) environment.

Table 57: ml4aml Conda Environment

Package	Version
sqlalchemy	2.0.19
xgboost	1.7.6
seaborn	0.12.2
scikit-learn	1.2.2
SHAP	0.42.1
ELI5	0.13.0
PDPbox	0.3.0
Imbalanced learn	0.10.1
py4j	0.10.9.7
scikit-optimize	0.9.0
statsmodels	0.14.0
pyod	1.1.0
oracledb	1.2.2
numpy	1.24.4

Table 57: ml4aml Conda Environment

Package	Version
scipy	1.11.1
pandas	1.5.3
matplotlib	3.7.2
requests	2.31.0
minisom	2.3.1
Matplotlib-venn	0.11.9

> **NOTE** The **Pyspark** python package is not part of the default environment. To install pyspark python package in the environment, see the Install Pyspark for ml4aml conda python environment section.

Install Pyspark for ml4aml conda python environment

To use this feature, download the pyspark python package from the deployed spark distribution and install the package in the conda python environment of the Compliance Studio.

To install the pyspark python package, follow these steps:

1. Log in to the **UNIX** machine where Compliance Studio is installed.
2. Navigate to `<COMPLAINACE_STUDIO_INSTALLED_PATH>/deployed/python_packages/ml4aml/bin` directory.
3. If the machine is connected to the internet then install by executing the following command:

   ```
   ./python3 -m pip install pyspark
   ```

4. If the machine is not connected to the internet then download the available package from the deployed spark.
5. Copy the package to any location in the **UNIX** machine and install by executing the following commands:

   ```
   /python3 -m pip install pyspark --no-index --find-
   links $FULL_PATH_INCLUDING_PYSPARK_PACKAGE_NAME
   ```

Table 58 list libraries for sane_8.1.2.6.0 conda python (**3.9.17**) environment.

Table 58: Sane Conda Environment

Package	Version
catboost	1.2
certifi	2021.10.8
cffi	1.15.1
conda-pack	0.6.0
contourpy	1.1.0
cryptography	41.0.1
cx-Oracle	8.3.0

Table 58: Sane Conda Environment

Package	Version
cycler	0.11.0
deprecation	2.1.0
ds-interpreter-client	23.4.2
fonttools	4.40.0
globalparty	8.1.2.6.0rc8
graphviz	0.20.1
importlib-resources	5.12.0
jaro-winkler	2.0.3
jellyfish	0.11.2
kiwisolver	1.4.4
Levenshtein	0.21.1
matplotlib	3.7.1
mmg	8.1.2.5.0
numpy	1.22.4
oracle-pypgx-client	23.4.2
oracledb	1.3.2
packaging	21.3
pandas	1.5.3
Pillow	9.5.0
pip	23.2.1
plotly	5.15.0
py4j	0.10.9.5
pycparser	2.21
pyparsing	3.1.0
python-dateutil	2.8.2
python-Levenshtein	0.21.1
pytz	2021.3
pyxDamerauLevenshtein	1.7.1
rapidfuzz	3.1.1
retrying	1.3.4
sane-common	0.2.4
scipy	1.11.0

Table 58: Sane Conda Environment

Package	Version
setuptools	68.0.0
six	1.16.0
tenacity	8.2.2
textdistance	4.5.0
urllib3	1.26.16
wheel	0.41.2
zipp	3.15.0

9.20 Configure Custom Notebook in ECM

Notebooks can be embedded within ECM (Enterprise Case Management) to help enhance the investigation process. This section provides the details for how to configure this.

> **NOTE** If you are using Investigation Toolkit, see OFS Investigation Hub Installation Guide and OFS Investigation Hub Administration and Configuration Guide for configuration.

9.20.1 Prerequisites

- Install the ECM application. To install ECM, see OFS Enterprise Case Management Installation Guide.
- Configure PGX Interpreter for Graph functionality. To obtain PGX Interpreter, contact My Oracle Support (MOS).

9.20.2 Importing Notebook

Users can import or create their own notebooks into the Data Studio and integrate into ECM for investigation.

To import notebooks, follow these steps:

1. Login to the Data Studio application.

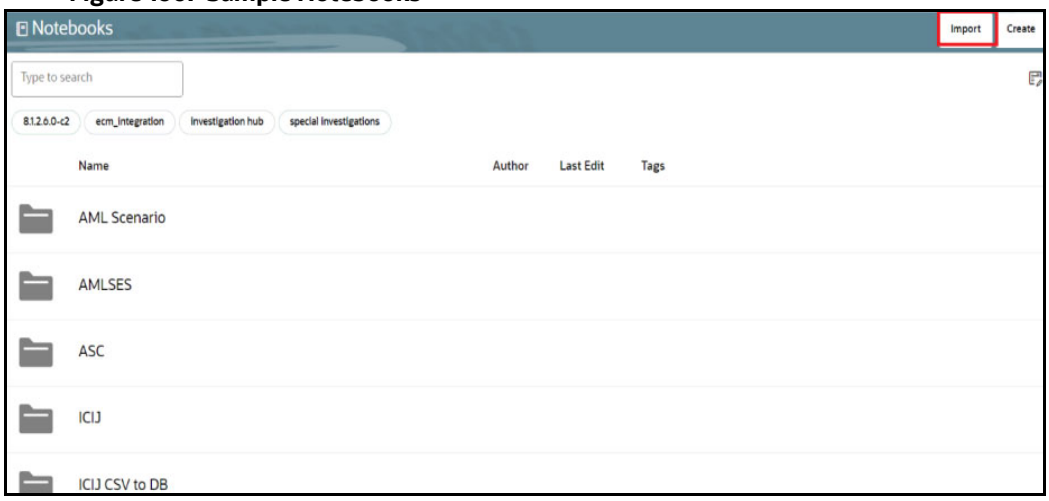
   `https://<Host_Name>:<Port_Number>/cs`

   Here <`Port_Number`> is **7008** for the Data Studio application.

   > **NOTE** If the user is logging in for the first time, then login to Compliance Studio first and then access the Data Studio.

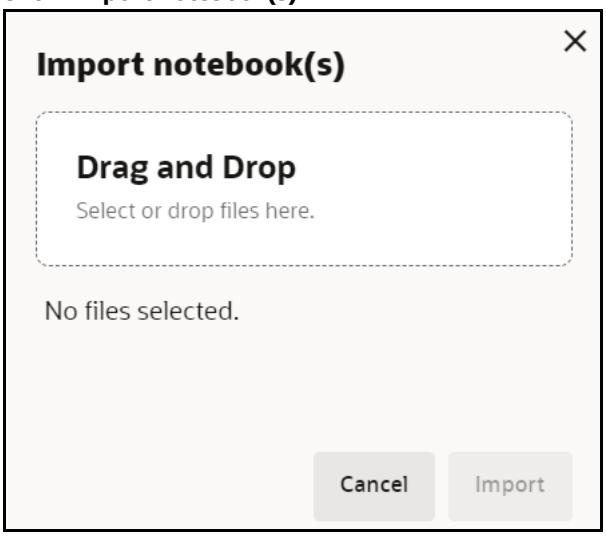
   Once logged in, the Notebooks page is displayed.

Figure 160: Sample Notebooks



2. Click **Import**. The Import notebook(s) pane is displayed.

Figure 161: Import notebook(s)



3. Click **Drag and Drop** and select your notebook from the local directory.
4. Click **Open**. The selected notebook is added to the Import notebook(s) pane.
5. Click **Import**. The notebook will be imported and available in the Notebooks page.
6. Click the **Notebook** and you can see the paragraphs to investigate.

> **NOTE** The notebook is loaded with FCGM Default Template and you can also use alternate template based on your requirement.

9.20.3 User Group Mapping

User must be mapped to this **DSUSRGRP** group for using the notebook. For more information, see the User Access and Permissioning Management section.

9.20.4 Integrating Notebook with ECM

The notebook is integrated with ECM to enable Case Investigators to investigate cases in the ECM.

9.20.4.1 Enable Notebook Tab in ECM Case Designer

The pre-configured ECM patch enables the notebook tab for **AMLSURV** case types. An admin user can add the tab for other case types by using the Case Designer component in the ECM.

For more information, see **Adding Optional Entities to the Case Type** section in the OFS ECM Administration And Configuration Guide.

> **NOTE** Add case type and notebook Id mappings in the FCC_CM_CTYPE_NB_MAPPING table.

9.20.4.2 User Role Precedence for Notebook

User role precedence in the FCC_CM_NB_ROLES table to decide which notebook to investigate when users have multiple roles where the mapped notebook ids are different.

To set the precedence among roles by Admin user, follow these steps:

1. Connect to ECM's Atomic Schema.
2. Edit records present in the **FCC_CM_NB_ROLES** table.
3. Enter the user role in the **V_USERROLE** column and the precedence in the **N_PRECEDENCE** column.

> **NOTE** Lower value of precedence has higher precedence.

9.20.4.3 Mapping User Roles and Case Type with Notebook

This section can be used to configure specific roles and case types. An admin user can map the notebook against a role and case type.

Map additional case types, roles, and respective notebook id in the table. You can see examples as listed in Table 59.

Table 59: Example

V_CASETYPE	V_USERROLE	V_NOTEBOOK_ID	V_CREATED_DATE	V_CREATED_BY	V_UPDATED_BY	V_UPDATED_DATE	V_NB_TOOLBAR	V_ADD_PARA	V_PARA_ACTIONS	V_PARA_CODE
CASE_TYPE_1	ROLE_1	notebook_id_1	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_1	ROLE_2	notebook_id_2	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_2	ROLE_1	notebook_id_1	02-02-2024	02-02-2024			N	N	Y	N

Table 59: Example

V_CASETYPE	V_USERROLE	V_NOTEBOOK_ID	V_CREATED_DATE	V_CREATED_BY	V_UPDATED_BY	V_UPDATED_DATE	V_NB_TOOLBAR	V_ADD_PARA	V_PARA_ACTIONS	V_PARA_CODE
CASE_TYPE_2	ROLE_2	notebook_id_3	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_2	ROLE_3	notebook_id_5	02-02-2024	02-02-2024			N	N	Y	N

NOTE Roll out an update by replacing the existing notebook ids with updated notebook ids.

9.20.4.4 Authenticate User to Access Notebook Tab in ECM

NOTE The user needs a self-signed certificate to authenticate the user for accessing notebook in ECM.

If the user is not using the self-signed certificate, follow these steps:

1. Copy the following files from `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/conf` to the server where ECM is installed.
   - `studio_server.p12`
   - `studio_server.jks`

   NOTE Make sure that the "studio_server.p12" and "studio_server.jks" certificates are compatible with Java 8. This is applicable only if the Compliance Studio server is in JDK 11 and the ECM application server is in Java 8. If there is a difference in Java versions, then both the files "studio_server.p12" and "studio_server.jks" need to be recreated in Compliance Studio server and replaced in all necessary locations. For more information about these certificates, see **Generate Self-signed Certificate** section in the OFS Compliance Studio Installation Guide.

2. Run the following command to create certificate files:

   ```
   openssl pkcs12 -in studio_server.p12 -nokeys -out server_cert.pem

   openssl pkcs12 -in studio_server.p12 -nodes -nocerts -out server_key.pem

   keytool -certreq -keystore studio_server.jks -alias studio_server -
   keyalg RSA -file client.csr

   openssl x509 -req -CA server_cert.pem -CAkey server_key.pem -in
   client.csr -out client_certificate.pem -days 365 -Cacreateserial
   ```

3. Modify the path and run the following command:

   ```
   keytool -import -file "/<ECM Installation Path>/client_certificate.pem"
   -alias studio_server -keystore "<JDK Installed Directory>/lib/security/
   cacerts" -storepass "changeit"
   ```

For example,

```
keytool -import -file "Testserver/client_certificate.pem" -alias
studio_server -keystore "jdk-11.0.10/lib/security/cacerts" -storepass
"changeit"
```

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.