Oracle® Financial Services Compliance Studio

Administration and Configuration Guide

Release 8.1.2.8.0
G24525-01
February 2025

Oracle Financial Services Compliance Studio Administration and Configuration Guide, Release 8.1.2.8.0

G24525-01

Contents

Preface

Audience xi
Related Resources xi
Abbreviations xi
Documentation Accessibility xii
Diversity and Inclusion xii
Conventions xii
Comments and Suggestions xii

1 About Compliance Studio Administration

1.1 Capabilities offered by Compliance Studio 1-1
1.2 Configurable Features 1-3
1.3 Administration Overview 1-3
1.4 Key Concepts 1-4

2 User Access and Permissioning Management

2.1 Mapping User Groups 2-1
2.1.1 User Groups 2-1
2.1.2 User Group - Role Mapping 2-3
2.1.3 Functions and Roles required to perform CRUD operations for Conda 2-4
2.2 Access Compliance Studio Using SAML Realm 2-5

3 Interpreter Configuration and Connectivity

3.1 Configure Interpreters 3-2
3.1.1 python Interpreter 3-8
3.1.1.1 Configure a python Interpreter 3-9
3.1.1.2 Change Version in the Python Interpreter 3-9
3.1.2 jdbc Interpreter 3-10
3.1.2.1 Configure a jdbc Interpreter 3-11
3.1.2.2 Link Wallet Credentials to jdbc Interpreter 3-16
3.1.3 md Interpreter 3-16

3.1.4 PGX Interpreter 3-17
3.1.5 Spark Interpreter 3-18
3.1.5.1 Spark Interpreter in Local Mode 3-18
3.1.5.2 Enable Additional Spark Interpreter 3-21
3.1.6 Pyspark Interpreter 3-21
3.1.6.1 Prerequisites 3-21
3.1.6.2 Configuration 3-22
3.1.6.3 Use Python Virtual Environments with PySpark 3-23
3.1.6.4 Configure Pyspark Interpreter 3-24
3.1.6.5 Enable Additional PySpark Interpreter 3-24
3.2 Create a Credential 3-25
3-27
3.3 Link Credentials 3-28
3.4 Create an Interpreter Group 3-30
3.5 Create an Interpreter Variant 3-30

4 Entity Resolution

4.1 Using Pre-configured Datasets and Rulesets 4-2
4.1.1 Pre-configured Rulesets for Matching, Merging, and Data Survival 4-2
4.1.2 Custom Rulesets for Matching 4-3
4.2 FCCM out-of-the-box Entity Resolution Pipeline on FSDF 4-4
4.2.1 Pre-configured Entity Resolution Pipelines 4-4
4.2.2 Prerequisites for out-of-the-box ER Pipelines 4-4
4.2.2.1 Creating Pre-Staging Tables in FSDF 4-4
4.2.3 Load Data into Pre-Staging Tables 4-5
4.2.4 Output Tables 4-7
4.2.5 Entity Resolution Mapping Information 4-8
4.2.6 Consolidated Information of the Resolved Entities 4-13
4.3 Prerequisites when MATCHING_MECHANISM is Set to Oracle Text (OT) 4-14
4.4 Executing the ER Jobs 4-14
4.4.1 Create Index and Load the Data 4-15
4.4.1.1 Additional Configurations 4-17
4.4.2 Perform Matching 4-20
4.4.2.1 Matching Output 4-21
4.4.2.2 Additional Configuration for Matching with Oracle Text (OT) 4-21
4.4.3 Data Survival 4-22
4.4.4 Load Data in FCC_ER_OUTPUT Table 4-26
4.4.5 Initial Run for High Volume Data 4-28
4.4.6 Status Codes 4-28
4.4.7 Using Wrapper Shell Script 4-28
4.5 Persisting the Data 4-30

4.5.1 Persisting the Data When F_PERSIST_GUID and F_MANUAL_APPROVAL Flags are Set to False Condition 4-30
4.5.2 Persisting the Data When F_PERSIST_GUID Flag is Set to True and F_MANUAL_APPROVAL Flag is Set to True/False Condition 4-34
4.6 Metadata Tables for Entity Resolution 4-39
4.6.1 Default Data in the tables 4-39
4.6.2 Customize Data in ER Tables 4-43
4.6.3 Populate Metadata for Data Survival in the Compliance Studio Schema 4-49
4.7 Removal of Entities from the Global Party (Deleted Party) 4-50
4.7.1 Impact on Manual Decisioning for Deleting Parties 4-51
4.8 Ability to Remove Split and Merge Action Manually 4-51
4.9 Expiry of Entity Child Records Mapping 4-52
4.9.1 Expiry of Entity Address Mapping 4-52
4.9.2 Expiry of Entity Phone Mapping 4-52
4.9.3 Expiry of Entity Email Mapping 4-52
4.9.4 Expiry of Entity Document Mapping 4-53
4.10 Statistics for ER Job Execution 4-53

5 Use Cases

5.1 Automated Scenario Calibration (ASC) 5-1
5.1.1 Creating ASC Workspace 5-1
5.1.2 Importing Workspace Metadata 5-4
5.1.3 Using Scenario Conversion Utility for ASC 5-5
5.1.3.1 Conversion Steps 5-6
5.1.4 Comparison of Events between BD and Conversion Utility 5-18
5.1.5 Using Delete Threshold Sets Notebook 5-20
5.1.6 Using Dynamic Datasets with AML Scenario Conversion 5-21
5.1.7 Advanced Concepts for ASC 5-23
5.2 Behavioral Model 5-23
5.2.1 Creating Sandbox Workspace 5-24
5.2.2 Populating Sandbox Workspace 5-24
5.2.3 Importing Workspace Metadata 5-24
5.2.4 Batch Framework for Behavioral Model 5-24
5.2.4.1 Behavioral Model Aggregate Base Features 5-25
5.2.4.2 Behavioral Model Scoring 5-28
5.2.4.3 Behavioral Model Annual Model Validation 5-34
5.2.4.4 Behavioral Model Monthly Model Validation 5-36
5.2.4.5 Obtain the SAR Information 5-38
5.2.5 Execute Batch 5-41
5.2.6 Monitor Batch 5-41
5.3 Sanctions Event Scoring 5-41
5.3.1 Creating Production Workspace 5-41

5.3.2 Prerequisites for Creating Sandbox Workspace 5-44
5.3.3 Creating Sandbox Workspace 5-44
5.3.4 Populating Sandbox Workspace 5-48
5.3.5 Importing Workspace Metadata 5-48
5.3.6 Batch Framework for Sanctions Event Scoring 5-49
5.3.7 Execute Batch 5-59
5.3.8 Monitor Batch 5-59
5.4 AML Event Scoring 5-60
5.4.1 Creating Sandbox Workspace 5-61
5.4.2 Populating Sandbox Workspace 5-61
5.4.3 Importing Workspace Metadata 5-61
5.4.4 Batch Framework for AML Event Scoring 5-62
5.4.4.1 AMLES Historic Event Load 5-62
5.4.4.2 AMLES Scoring 5-63
5.4.5 Execute Batch 5-64
5.4.6 Monitor Batch 5-65
5.5 Customer Segmentation and Anomaly Detection 5-65
5.5.1 Creating Sandbox Workspace 5-65
5.5.2 Populating Sandbox Workspace 5-65
5.5.3 Importing Workspace Metadata 5-65
5.5.4 Data Movement 5-66
5.5.5 Batch Framework for Customer Segmentation and Anamoly Detection 5-68
5.5.6 Execute Batch 5-68
5.5.7 Monitor Batch 5-68
5.6 Customer Risk Scoring 5-68
5.6.1 Creating Sandbox Workspace 5-69
5.6.2 Populating Sandbox Workspace 5-69
5.6.3 Importing Workspace Metadata 5-69
5.6.4 Obtaining SAR Labels for Customer Risk Scoring 5-69
5.6.5 Obtain SAR information for Production 5-72
5.6.5.1 Create a New Batch for Obtaining Investigated Entities 5-75
5.6.6 Data Movement 5-78
5.6.6.1 Export from Sandbox 5-79
5.6.6.2 Import into Production 5-80
5.6.7 Batch Framework for Customer Risk Scoring 5-81
5.6.7.1 Supervised Historic Data Load 5-81
5.6.7.2 Supervised Scoring 5-82
5.6.7.3 Annual Model Validation 5-85
5.6.7.4 Monthly Model Validation 5-86
5.6.8 Execute Batch 5-88
5.6.9 Monitor Batch 5-88
5.6.10 ECM Connector Batch 5-88

5.7 Shell Account Detection Scenario for AML 5-88
5.7.1 Creating Sandbox Workspace 5-89
5.7.2 Populating Sandbox Workspace 5-89
5.7.3 Importing Workspace Metadata 5-89
5.7.4 Batch Framework for Shell Account Detection Scenario for AML 5-89
5.7.4.1 AML Scenario Processing batch 5-89
5.7.5 ECM Connector Batch 5-91
5.8 Custom Scenario 5-91
5.8.1 Batch Framework for Custom Scenario 5-92
5.8.1.1 Aggregate Base Features for Custom Scenario 5-93
5.8.1.2 Event Generation for Custom Scenario 5-96
5.8.2 Execute Batch 5-98
5.8.3 Monitor Batch 5-98

6 Restart Services

6.1 Stop and Start the Compliance Studio Services 6-1
6.2 Stop and Start the PGX Service 6-1

A Appendix

A.1 Create Metadata Indexes using Logstash A-1
A.2 Unlock the Notebook A-1
A.3 Checking IP Address for User’s Last Login A-1
A.4 Roles, Functions and Permissions A-2
A.4.1 Roles A-2
A.4.2 Default Roles Seeded in Notebook Server through permissions-int.yml file A-3
A.4.3 Functions in Compliance Studio A-4
A.4.4 Permissions in Notebook Server A-8
A.4.5 Group - Role Mapping A-10
A.4.6 Role - Function Mapping A-12
A.4.7 Role - Permission Mapping A-16
A.5 Setting Memory of Entity Resolution and Matching Services A-20
A.6 Cleanup Steps When the Create Index and Load Data Job Terminated Manually A-21
A.7 Cleanup Steps When the Bulk Similarity Job Terminated Manually A-22
A.8 Cleanup Steps When the Data Survival Job Terminated Manually A-22
A.9 Cleanup Steps When the Load Data in FCC_ER_OUTPUT Job Terminated Manually A-22
A.10 Resetting Entity Resolution Back to Day 0 A-23
A.10.1 Compliance Studio Schema Changes A-23
A.11 Utility Scripts A-23
A.11.1 Data Slicing Utility Script A-23
A.12 Load Data into ICIJ Tables A-25

A.13 Prescript Condition A-28
A.14 Resetting Graph Pipeline Back to Day 0 A-28
A.15 Disable User in Compliance Studio after SSO Login A-29
A.16 Migrating the Data from ElasticSearch to OpenSearch A-30
A.17 Parameters for Entity Resolution Job execution A-34
A.18 Conda Environment in Notebook A-37
A.19 Python Libraries for Predefined Conda Environment A-41
A.20 Implementation of Connection Pooling in PGX Realm A-46
A.21 Configure Custom Notebook in ECM A-49
A.21.1 Prerequisites A-49
A.21.2 Importing Notebook A-49
A.21.3 User Group Mapping A-51
A.21.4 Integrating Notebook with ECM A-51
A.22 How-To A-53
A.22.1 How to Create Data Store A-53
A.22.2 How to Register Conda Environment in BD Production Workspace A-56
A.22.3 How to Create Sandbox Workspace A-57
A.22.4 How to Populate the Sandbox Workspace A-62
A.22.5 How to Execute Batch A-65
A.22.6 How to Monitor Batch A-66
A.22.7 How to Add User Defined Transformation (UDT) as Python Module A-67
A.22.8 How to get Studio Alert Tables into Workspace Schema A-68
A.23 Advanced Feature for ASC Use Case A-68
A.24 Incremental Workspace Refresh A-70
A.25 Data Model Support for AAI Applications A-71
A.26 Enable Additional Spark or PySpark interpreter A-71
A.26.1 Spark Interpreter User Impersonation A-76
A.26.2 Sample spark-default.conf Configuration File A-77
A.27 Enable Data Studio Options in Compliance Studio A-78
A.28 Rebuilding Indices in OpenSearch A-78

Document Control

The following table describes document control of this guide.

Table Document Control

Version Number	Revision Date	Change Log
8.1.2.8.3	January 2025	Added the Custom Scenario section. Added the EXTERNAL_ENTITY_ADDR table and Custom_Scenario_Scheduler_8.1.2.8.3 in the How to Create Sandbox Workspace section.
8.1.2.8.2	December 2024	Added the Sanctions_Scheduler_8.1.2.8.2 in Sanctions Event Scoring for both Sandbox and Production workspaces. Added these optional parameters (is_aai_batch and sanctions_batch_run_id) in SES Aggregate Events and SES Scoring batches for Sanctions Event Scoring.
8.1.2.8.1	October 2024	Added STRUCTURED_CASH_LIMIT_MIN and STRUCTURED_CASH_LIMIT_MAX parameters in the Table 5-4. Added the following sections: • Conversion Steps • Using Dynamic Datasets with AML Scenario Conversion • How to get Studio Alert Tables into Workspace Schema

Table (Cont.) Document Control

Version Number	Revision Date	Change Log
8.1.2.8.0	August 2024	Added the following sections: • Statistics for ER Job Execution • Expiry of Entity Child Records Mapping • Rebuilding Indices in OpenSearch • Enable Data Studio Options in Compliance Studio • Configuration for Create Index and Load the Data • Configuration for Data Survival • Sanctions Event Scoring Added N_CREATED_RUN_SKEY parameter in the Entity Resolution Mapping Information. Added parameters SINGLETON_TASK_PARALLEL_LEVEL and F_CAPTURE_COUNT_STAT in the Additional Configurations section. Added new pipeline (CSA_8128) and it is updated in all the applicable sections. Updated configuration steps in the Configure a jdbc Interpreter and Spark Interpreter in Local Mode sections.

Preface

This section provides information on the Oracle Financial Services (OFS) Compliance Studio Administration and Configuration Guide.

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

Related Resources

This section identifies additional resources to the OFS Compliance Studio. You can access additional documents from the Oracle Help Center.

Abbreviations

The following table lists the abbreviations used in this document.

Table 1 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

Table 1 (Cont.) Abbreviations

Abbreviation	Meaning
MMG	Model Management and Governance
SSO	Single Sign-On
SSH	Secure Shell
FSDF	Financial Services Data Foundation

Documentation Accessibility

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.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

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

Comments and Suggestions

Please give us feedback about Oracle Applications Help and guides! You can send an e-mail to: https://support.oracle.com/portal/.

1

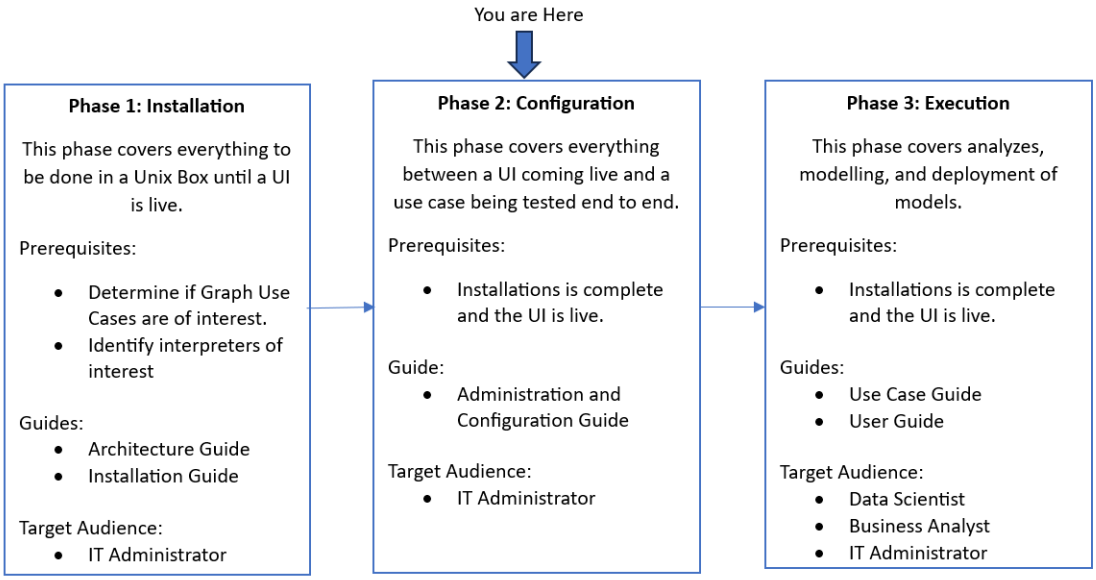
About Compliance Studio Administration

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.

When to Use this Guide

The following illustration demonstrates when this guide should be used.

Figure 1-1 When to Use this Guide



1.1 Capabilities offered by Compliance Studio

This section lists the Compliance studio capabilities:

- 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

1.2 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.

1.3 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).

1.4 Key Concepts

This section provides an overview of key concepts in the Compliance Studio:

- **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 dataprocessing- backend. For example, to use Python code in Zeppelin, you need a %python 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**: PySpark is the Python API for Apache Spark. It enables you to perform real-time, largescale data processing in a distributed environment using Python. 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 thirdparty 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.
- **Conda**: Miniconda3 is a minimal installer for Conda, a package management and environment management system. It is a smaller, lighter alternative to Anaconda, which is a more comprehensive distribution. Here are some key points about Miniconda3:
  - **Minimal Installer**: Miniconda3 includes only Conda, Python, and a small number of necessary packages. It allows users to create custom Python environments with only the packages they need.
  - **Package Management**: Conda, the package manager included with Miniconda3, can install, update, and manage software packages and their dependencies. It can handle multiple versions of Python and other packages.
  - **Environment Management**: Miniconda3 allows users to create isolated environments for different projects, ensuring that dependencies for one project do not interfere with those for another. This is particularly useful for managing different versions of Python or other software libraries.
  - **Customizable**: Because it starts with a minimal set of packages, users can customize their environment by installing only the packages they need using Conda. This can lead to a more efficient and lightweight setup compared to a full Anaconda installation. Miniconda3 is particularly useful for users who want more control over their environment and prefer to install only the necessary packages for their specific projects.

    > **Note:**
    >
    > Conda helps us to upgrade python stack across Compliance Studio version at the same time maintaining older Conda environments for backward compatibility.
    >
    > For example, model deployed with an older version of Conda/Compliance Studio can co-exist with a model developed in the higher/upgraded version of the Conda/Compliance Studio.

  - **Workspace**: Compliance Studio provides the ability to create and manage sandbox workspaces for the creation and testing of models in a discrete schema with a subset of production data before deployment to the production workspace, where the model will be run on FCCM application data directly. The application comes with two predefined schemas to be used in sandboxes for model development with different subsets of data from production. The workspace administration functionality and orchestration capability will manage the movement of data from production to the sandbox.
    The workspace provides granular user access control for various activities performed within the workspace which includes data access, Notebook access, Scheduler access, etc., These workspaces allow models to be tested in the sandbox before deployment into the production environment.

2

User Access and Permissioning Management

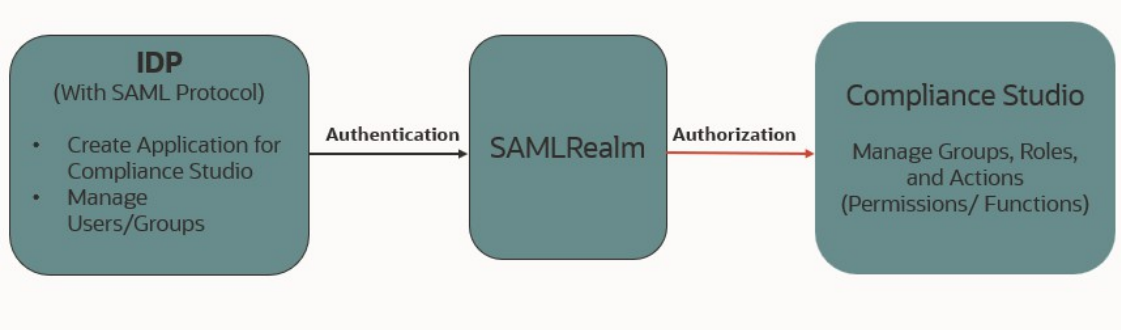
Compliance Studio uses a realm based on unique authentication and authorization for its users. Realm is a security policy domain defined for the application server. It is used to authenticate and authorize users of Compliance Studio.

SAML Realm is selected based on the Identity Provider (IDP) during the installation. 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.

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

Figure 2-1 Compliance Studio - Authentication and Authorization process



2.1 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.

2.1.1 User Groups

Table 2-1 User Groups

User Group	Description
IDNTYADMN	Identity Administrator group
IDNTYAUTH	Identity Authorizer group

Table 2-1 (Cont.) User Groups

User Group	Description
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.

2.1.2 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 2-2 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
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

Table 2-2 (Cont.) User Group to Role Mapping

Group Name	Role Name
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
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

2.1.3 Functions and Roles required to perform CRUD operations for Conda

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

Table 2-3 Functions and Roles

Function	Role	Groups Mapped	Access
CONDAENVSUMM	CONDAENVACCESS	• MDLUSR • MDLREV • MDLAPPR	Summary view
CONDAENVVIEW	CONDAENVREAD	• MDLUSR • MDLREV • MDLAPPR	Read

Table 2-3 (Cont.) Functions and Roles

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

2.2 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:

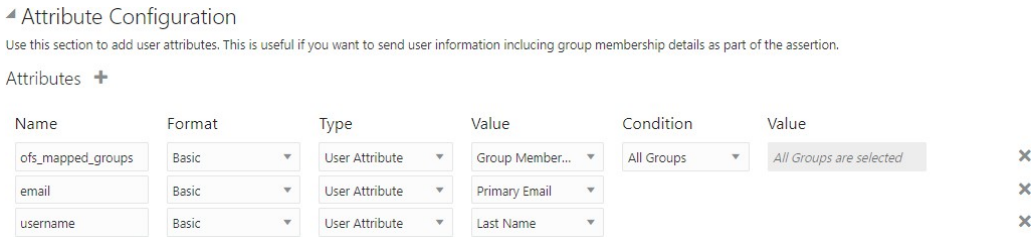
1. Create the following Group in the IDP system. For more information on creating groups in IDP, see the OFS Admin Console User Guide.
   - Create the new groups with the same name as the pre-configured groups. For more information, see the User Groups section.
2. Create a SAML application in IDP.
3.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



Update the SAML attribute configuration as tabulated in the following table.

Table 2-4 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.

3

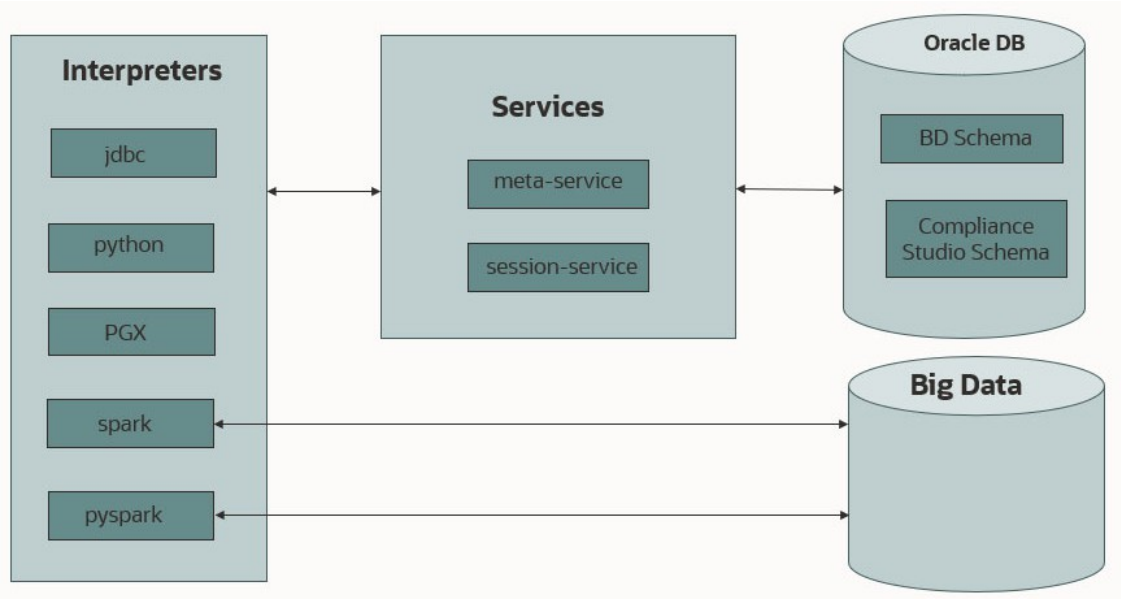
Interpreter Configuration and Connectivity

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 3-1 Examples of Interpreters



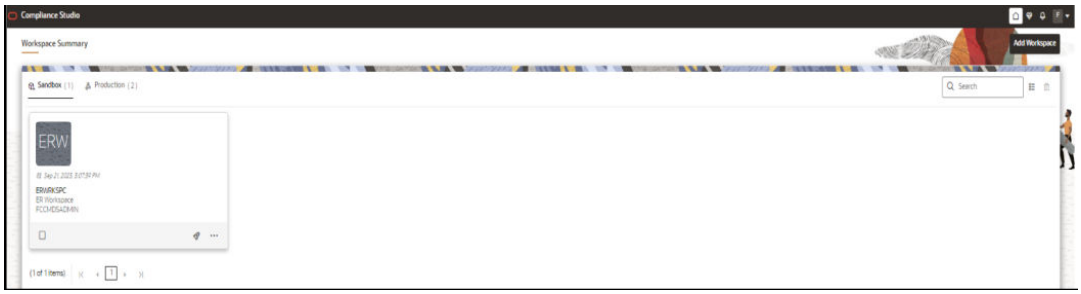
3.1 Configure Interpreters

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:

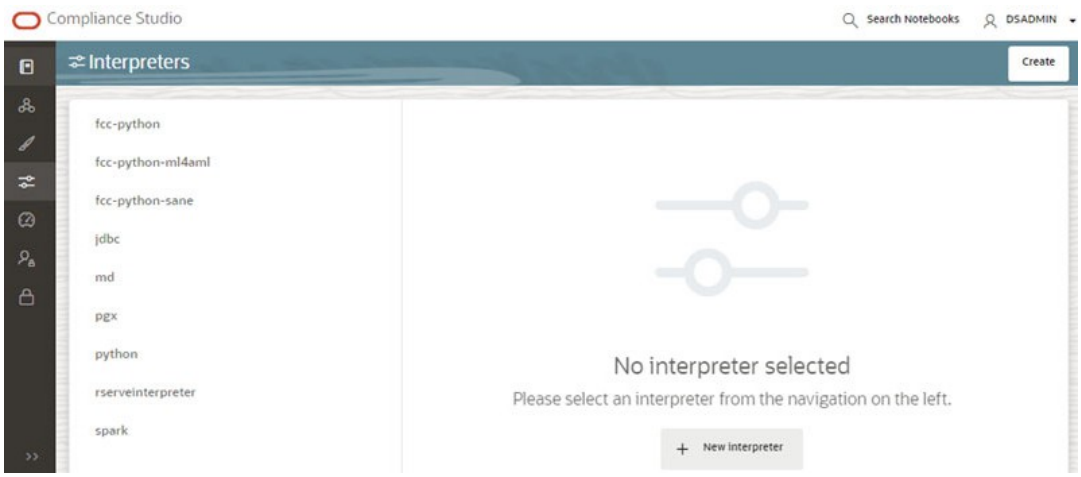
1. On the **Workspace Summary** page, select Launch workspace to display the **CS Production workspace** window.

Figure 3-2 Workspace Summary



2. Click 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 3-3 Interpreters' screen

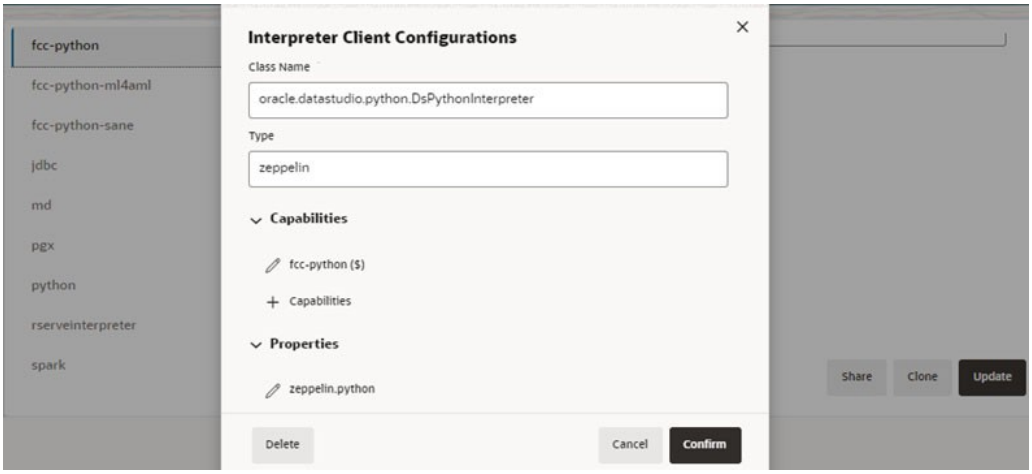


4. Modify the values in the fields as per requirement. For example, to modify a parameter's limit, connect to a different schema, PGX server, etc.

You can modify the values in the following UI options:

- Wizard

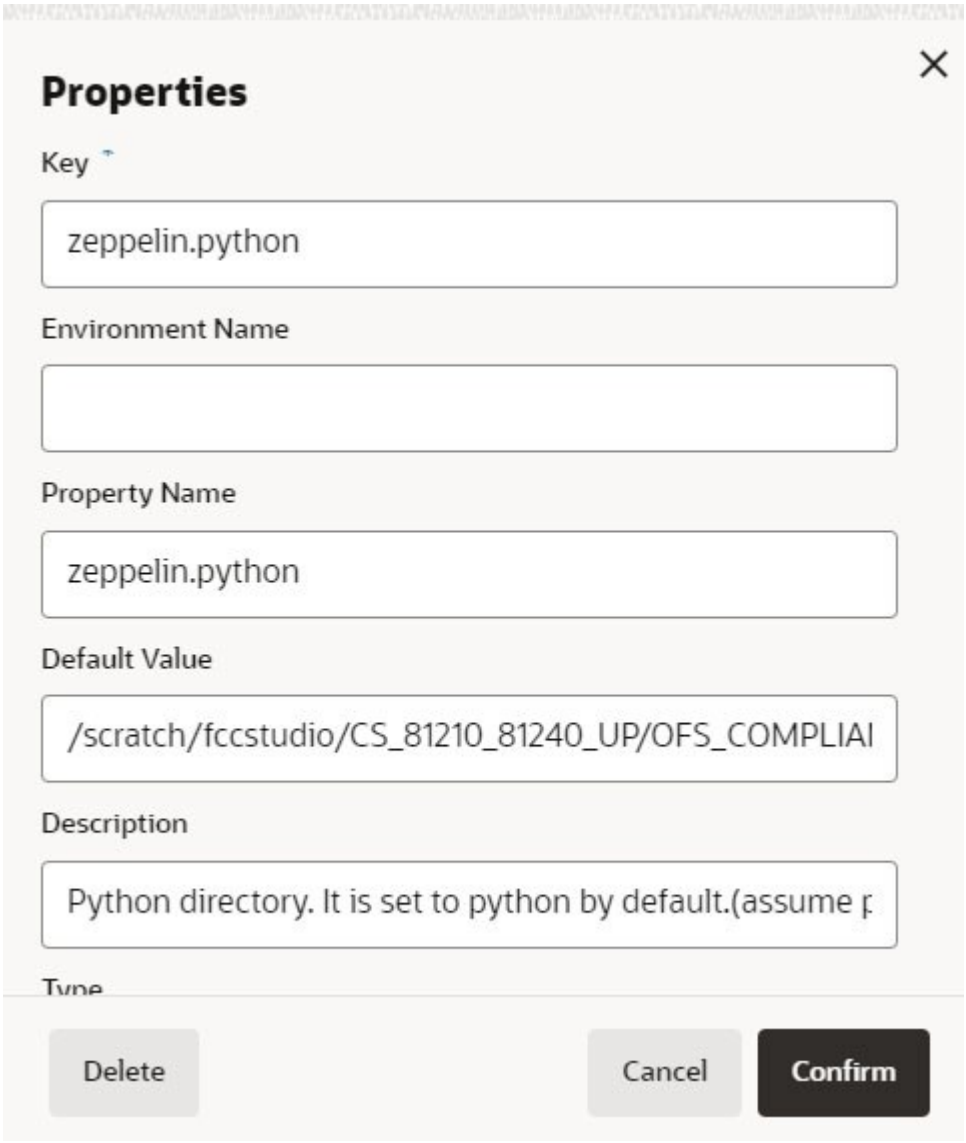
Figure 3-4 Wizard UI options



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 3-5 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.

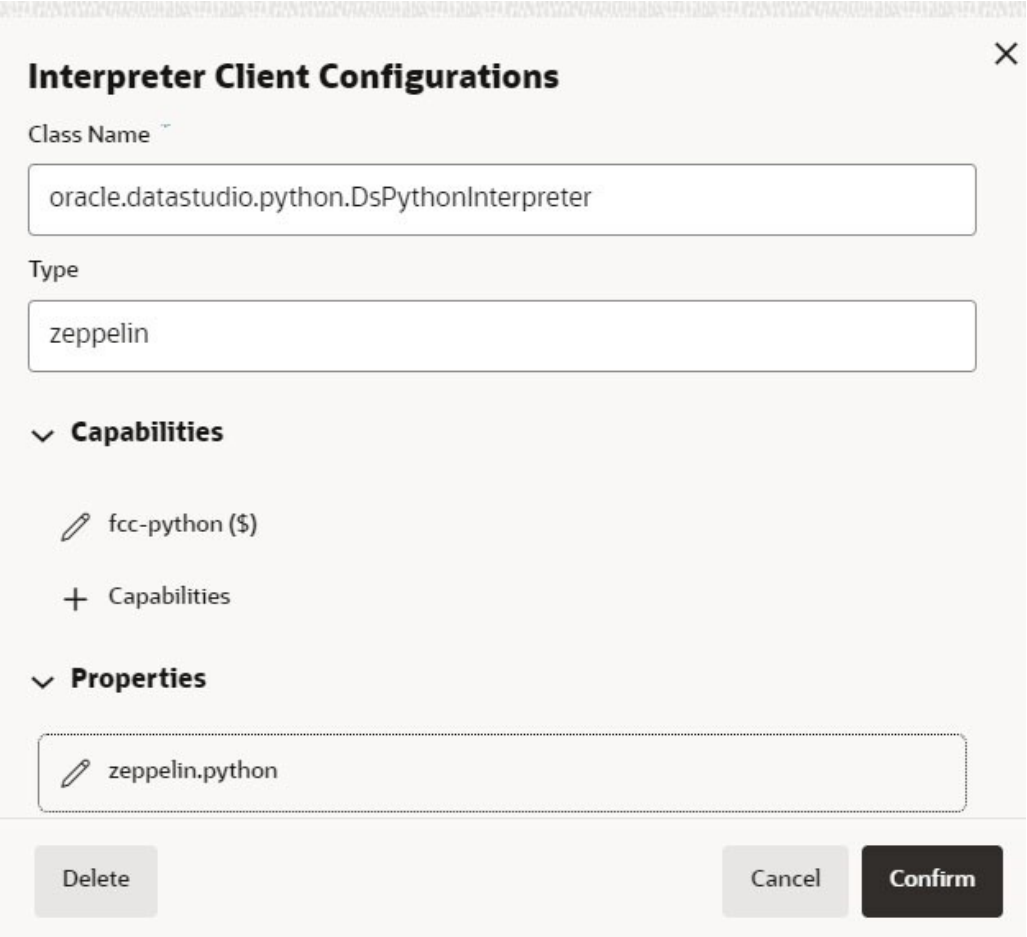
Credential Configurations

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 3-6 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.

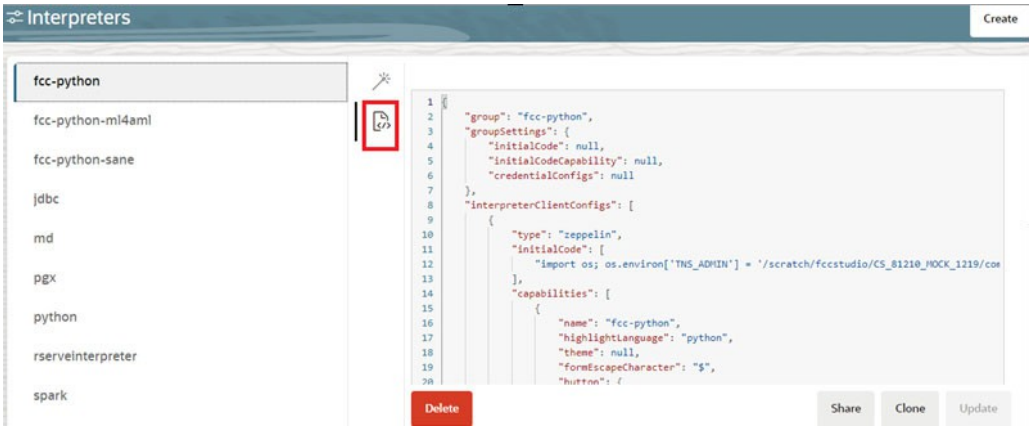
Credentials

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 3-7 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.

The following table lists the Ready-to-use interpreter in Compliance Studio.

Table 3-1 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_<CS version> • ml4aml_<CS version> • sane_<CS version> Before executing any python notebooks, you need to attach the conda environment using drop-down list.

Table 3-1 (Cont.) Ready-to-use interpreter

Interpreters	Description
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.
jdbc 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.
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.

Table 3-1 (Cont.) Ready-to-use interpreter

Interpreters	Description
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.
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.
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.

3.1.1 python Interpreter

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

- **default_<CS Version>**
- **ml4aml_<CS Version>**
- **sane_<CS Version>**

%python interpreter points to a different conda environment. The following table lists the predefined conda environment.

Table 3-2 Predefined Conda Environment

Conda Environment	Description
default_<CS version>	Default python interpreter.
ml4aml_<CS version>	Python interpreter for ML4AML use cases.

Table 3-2 (Cont.) Predefined Conda Environment

Conda Environment	Description
sane_<CS version>	Python interpreter for scoring Name and Address Matching.

> **Note:**
>
> Users can also configure the python libraries. For more information about python libraries, see the Python Libraries for Predefined Conda Environment section.

3.1.1.1 Configure a python Interpreter

To configure an python interpreter, follow these steps:

1. On the Interpreter page LHS menu, select python. The python interpreter pane is displayed.
2. 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.
3. Enter the following information in the python interpreter variant pane as described in the following table.

Table 3-3 Python Interpreter Settings

Field	Description
zeppelin.python	Enter the Python installed path. The value points to the default Python version set for the Interpreter.
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. **Note**: You can increase the limit upto 10240000 bytes. Increasing the default value from 102400 bytes to larger values is going to slow down the rendering of outputs in the UI.

3.1.1.2 Change 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 the 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/pythonpackages/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.

3.1.2 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/mmgstudio/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.

3.1.2.1 Configure a 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.

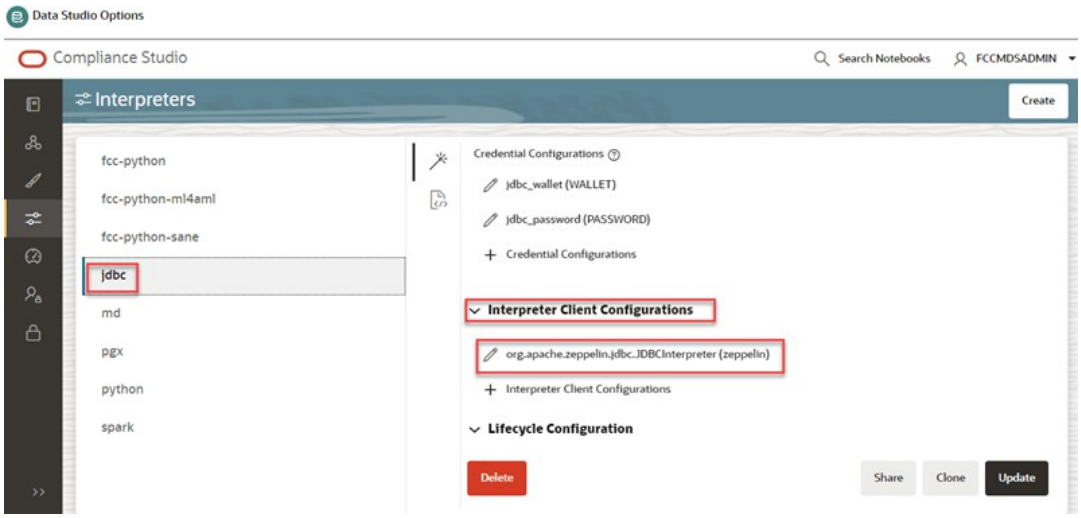
To configure a jdbc interpreter, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/bin` directory.
2. Open `startup.sh` file, navigate to line 29 and update jdbc value as **7011**.

   For example: . ./"$DIR"/datastudio --port 7008 --markdown 7009 --spark 7014 --python 7012 - -**jdbc 7011** --shell -1 --pgx 7022 --external
3. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/bin` directory.
4. Open the `config.sh` file and update DATASTUDIO_JDBC_INTERPRETER_PORT as **7011**.
5. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/bin` directory.
6. Open the `config.sh` file and update DATASTUDIO_JDBC_INTERPRETER_PORT as **7011**.
7. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/server/builtin/interpreters/jdbc.json` directory.
8. Navigate to line 154 and update port value as **7011**.

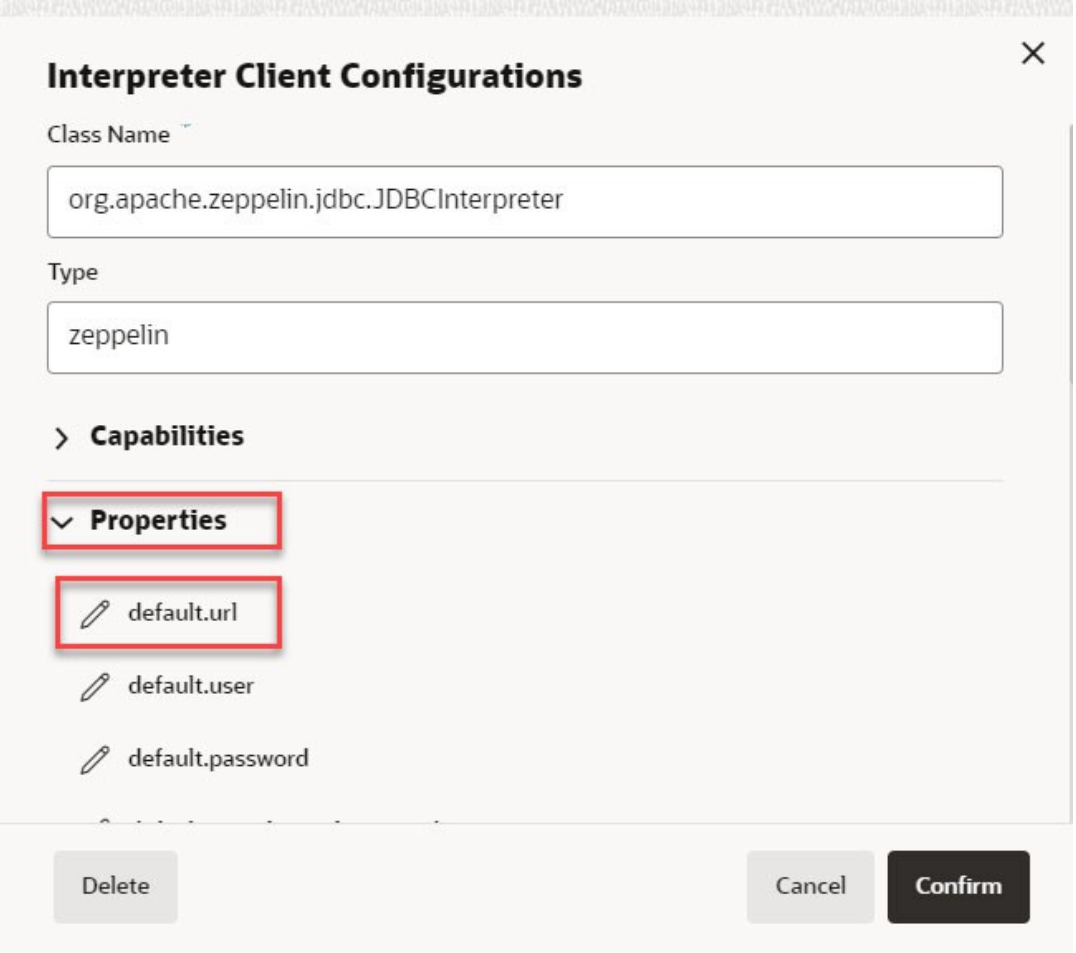
9. Restart Compliance Studio.
10. On the Interpreter page LHS menu, select jdbc. The jdbc interpreter pane is displayed.

Figure 3-8 jdbc Interpreter



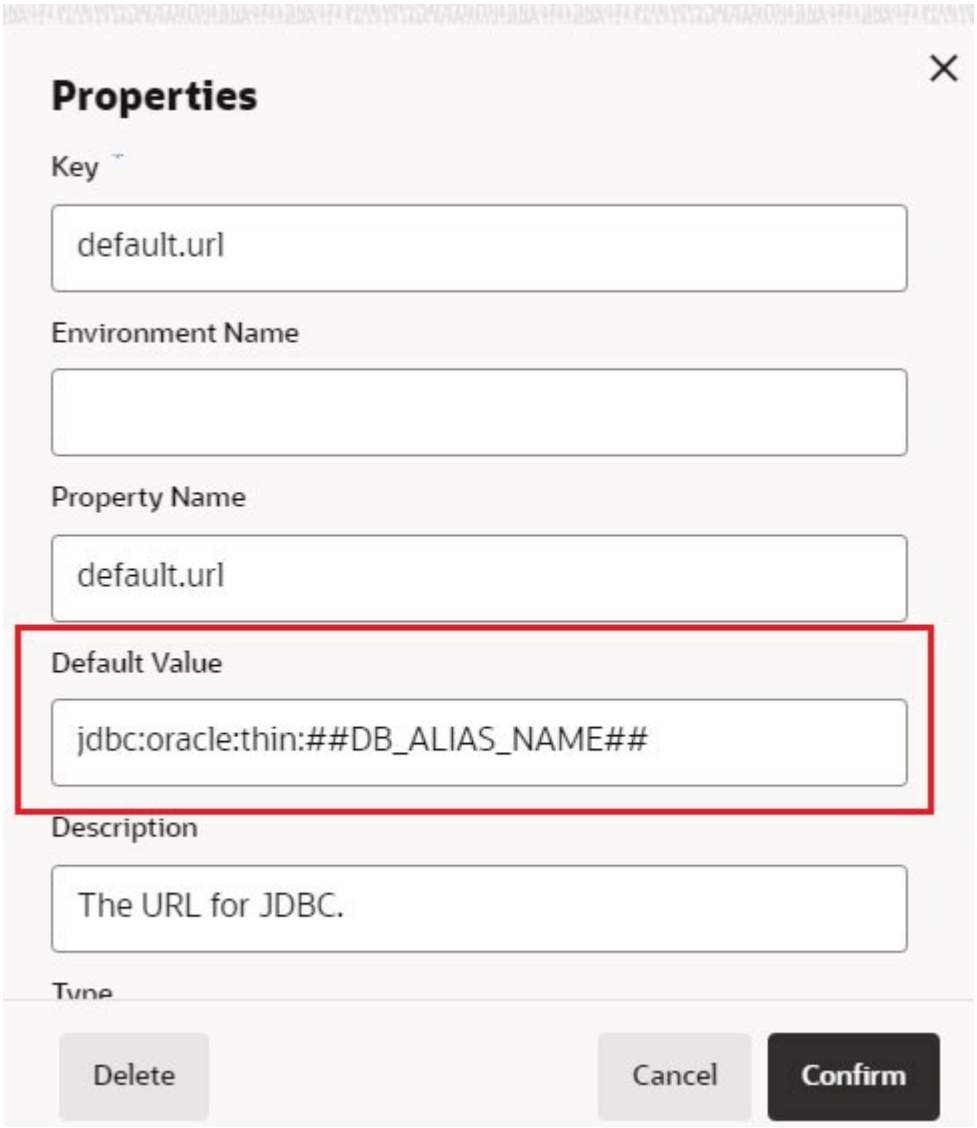
11. On Interpreter Settings page, expand **Interpreter Client Configurations** and click the Edit icon on the **<Class Name> (zeppelin)**. The Interpreter Client Configurations Window is displayed.

Figure 3-9 Interpreter Client Configurations



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

Figure 3-10 Properties



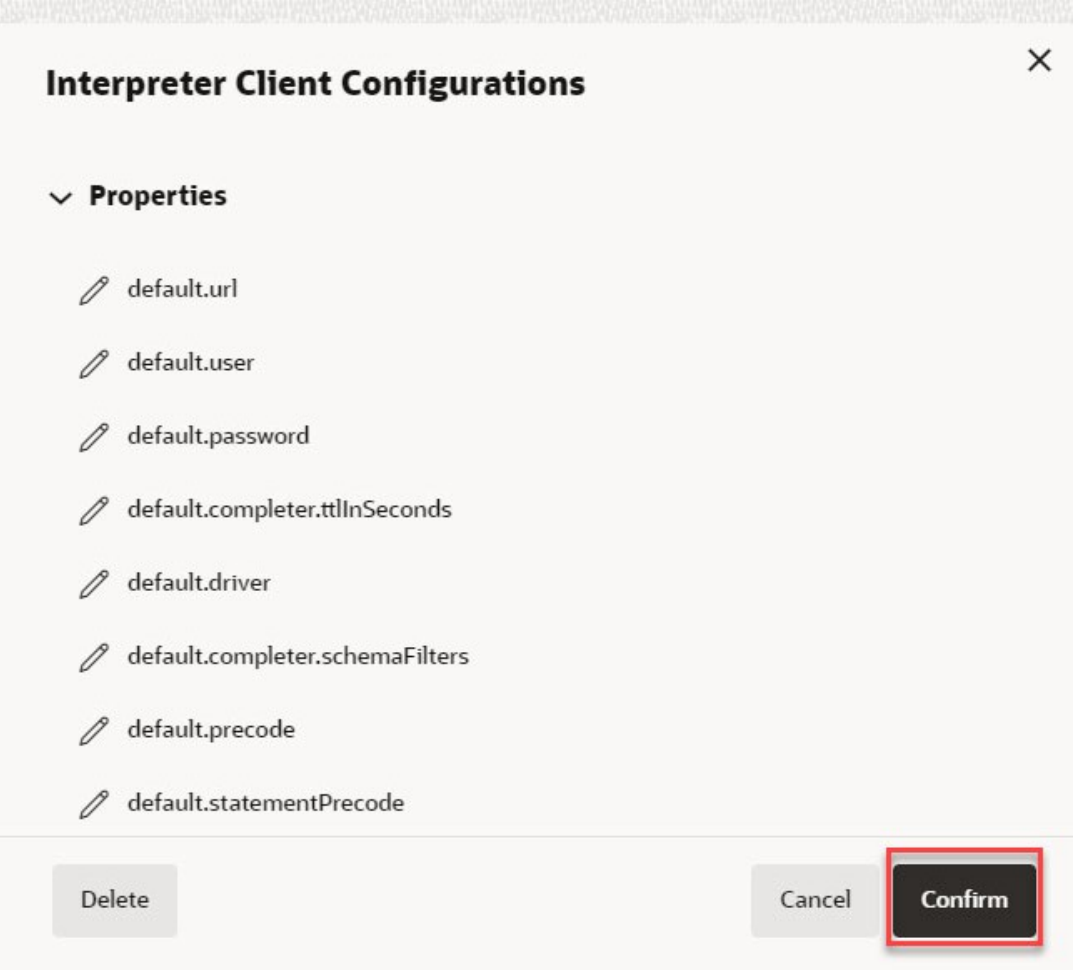
13. Enter the alias name in the **Default Value** field.

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

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

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

Figure 3-11 Interpreter Client Configurations



15. Click default.user property and it should be null in the **Default Value** field.
16. 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.

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

3.1.2.2 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. 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.

3.1.3 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 the Interpreter Settings page, expand **Interpreter Client Configurations** and click the 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.

3.1.4 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 the Interpreter Settings page, expand **Interpreter Client Configurations** and click the 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 following table.

Table 3-4 PGX interpreter

Field	Description
graphviz.formatter.class	Enter the class which implements the formatting of the visualization output. For example,`oracle.datastudio.graphviz.formatter.DataStudi oFormatter`
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.outpu t.limit	Enter the output message limit. Any message that exceeds the limit is truncated. For example, 102 or 400.

Table 3-4 (Cont.) PGX interpreter

Field	Description
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.useIPyth on	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 Version in the Python Interpreter section.

3.1.5 Spark Interpreter

This section explains about Spark Interpreter configurations.

3.1.5.1 Spark Interpreter in Local Mode

To start spark interpreter in the local mode, follow these steps:

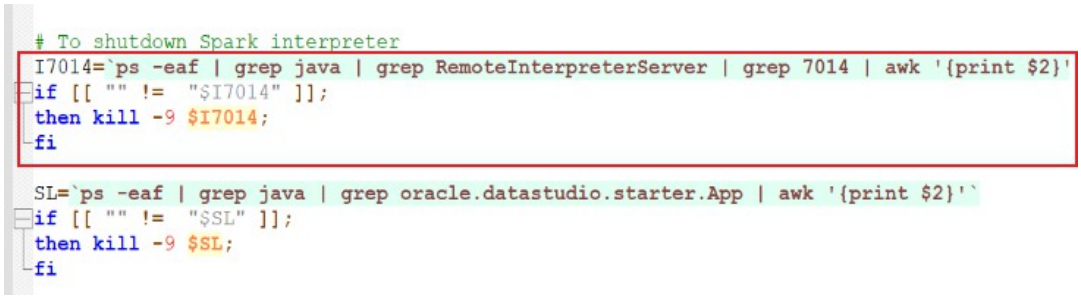
1. Download `spark-3.0.3-bin-hadoop2.7.tgz` from the website.
2. Unzip the spark hadoop cluster's zip file in the below mentioned locations:
   - `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/interpreter-server/spark-interpreter-<version>/extralibs`
   - `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-studio/interpreter-server/spark-interpreter-<version>/extralibs`
3. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/bin` directory.
4. Open the `startup.sh` file and add following line before the line containing "counter=1";`nohup "$DIR"/../interpreter-server/spark-interpreter-<version>/bin/spark-interpreter &>> <path_to_save_the_logs>/<log_file_name>.log &`

Figure 3-12 Snapshot of startup.sh file

5. Save and close the file.
6. Open the `shutdown.sh` file and add following line before the line containing “SL=”.

```
I7014=`ps -eaf | grep java | grep RemoteInterpreterServer | grep 7014 |
awk '{print $2}'`
if [[ "" != "$I7014" ]];
then kill -9 $I7014;
fi
```

Figure 3-13 Snapshot of shutdown.sh file



> **Note:**
>
> In the above step, the port number for the spark interpreter is assumed to be 7014, the default port that comes with the installer. If a different port is used, then change the configuration accordingly.

7. Save and close the file.
8. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/bin` directory.
9. Open the `startup.sh` file, navigate to line 29 and update spark value as **7014**.

   For example: . ./"$DIR"/datastudio --port 7008 --markdown 7009 --**spark 7014** --python 7012 --jdbc 7011 --shell -1 --pgx 7022 --external
10. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/bin` directory.

11. Open the `config.sh` file and update the following parameters:
    - MMG_SPARK_ENABLED=true
    - SPARK_HOME=`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/interpreter-server/spark-interpreter-<version>/extralibs/spark-<version>-bin-hadoop<version>`
    - HADOOP_HOME=##HADOOP_HOME##

      > **Note:**
      >
      > Retain the placeholder as it is.

    - SPARK_MASTER=local
    - SPARK_DEPLOY_MODE=

      > **Note:**
      >
      > Retain the SPARK_DEPLOY_MODE as **blank**.

    - DATASTUDIO_SPARK_INTERPRETER_PORT=7014
12. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/bin` directory.
13. Open the `config.sh` file and update the following parameters:
    - MMG_SPARK_ENABLED=true

      > **Note:**
      >
      > By default, it is set to false. You can configure the following parameters only when **MMG_SPARK_ENABLED** is set to **true**.

    - SPARK_HOME=`<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/interpreter-server/spark-interpreter-<version>/extralibs/spark-<version>-bin-hadoop<version>`
    - HADOOP_HOME= ##HADOOP_HOME##

      > **Note:**
      >
      > Retain the placeholder as it is.

    - SPARK_MASTER= local
    - SPARK_DEPLOY_MODE=

      > **Note:**
      >
      > Retain the SPARK_DEPLOY_MODE as blank.

    - DATASTUDIO_SPARK_INTERPRETER_PORT=7014

14. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/server/builtin/interpreters/spark.json` directory.
15. Navigate to line 169 and update port value as **7014**.
16. Update **default value** as **local** for spark.master and **blank** for spark.submit.deployMode.

    For example:

```
"spark.master": {
                            "envName": "MASTER",
                            "propertyName": "spark.master",
                            "defaultValue": "local",
                            "description": "Spark master uri. ex) spark://
masterhost:7077",
                            "type": "string"
},
                            "spark.submit.deployMode": {
                                "envName": null,
                                "propertyName":"spark.submit.deployMode",
"defaultValue": "",
                            "description": "The deploy mode of Spark driver
program, either 'client' or 'cluster'",
                            "type": "string"
                 },
```

17. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
18. Restart Compliance Studio using the following command.

```
./compliance-studio.sh -restart
```

19. Verify if the spark-interpreter has started using the following command:

```
netstat -nltp | grep 7014
```

3.1.5.2 Enable Additional Spark Interpreter

Interpreter variant does not apply to spark interpreters. Hence, you must enable an additional set of interpreters.

To enable an additional spark interpreter, see Enable Additional Spark or PySpark interpreter section in the Appendix.

3.1.6 Pyspark Interpreter

Compliance Studio uses PySpark 2.4.0. Before you begin the configurations, check the prerequisites depending on your operation mode.

3.1.6.1 Prerequisites

The PySpark interpreter has the same prerequisites as that as the Spark Interpreter. For more information, see Spark Interpreter. Also, all Spark components must be configured to use the same Python version.

3.1.6.2 Configuration

The PySpark interpreter can be configured through the Spark interpreter, with the only exception being the Python version used. By default, the Python version is set to 3 that can be changed either in the interpreter JSON files before the startup or from the Interpreters page of the Compliance Studio application UI during runtime by changing the following properties:
To change the value of the `spark.pyspark.python` property before installing the Compliance Studio, follow these steps:

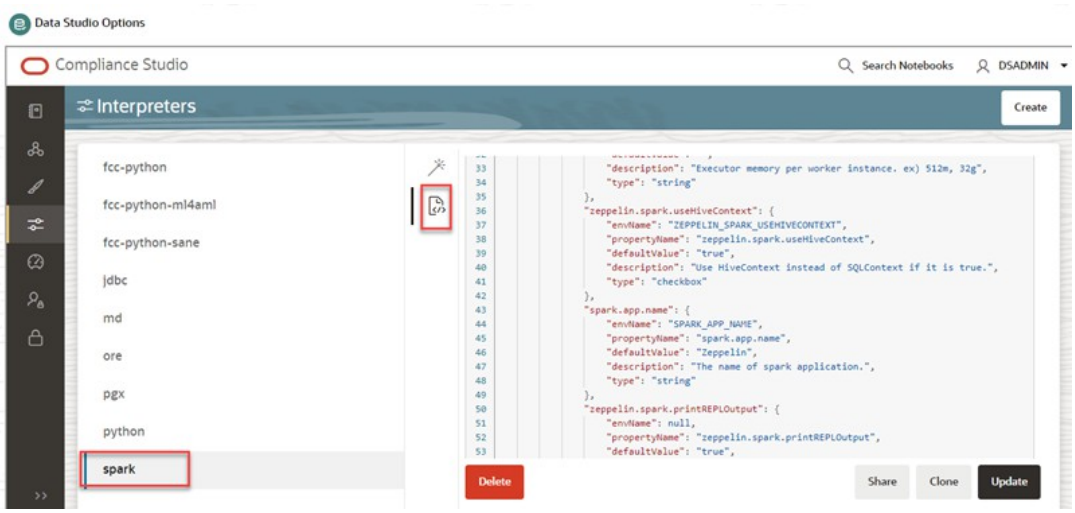
1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmgstudio/server/builtin/interpreters/spark.json` directory.
2. Update the value in `spark.pyspark.python` property of the spark.json file.

To change the value of the `spark.pyspark.python` property after installing the Compliance Studio, follow these steps:

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

   By default, the Interpreters page lists all the available interpreters on the LHS.

Figure 3-14 Spark Interpreter



5. Click **spark** interpreter on the LHS and then click **Plain Configuration** on the RHS.
6. Update the value in the spark.pyspark.python property and click **Update**.

   In the **Spark Interpreter Settings** page of the Compliance Studio application UI (or `spark.json` file), change the value of the `spark.pyspark.python` property to the Python executable path that is to be used by the Spark executors.

   In the **PySpark Interpreter Settings** page of the Compliance Studio application UI (or `pyspark.json` file), change the value of the `zeppelin.pyspark.python` property to the Python executable path that is to be used by the Spark driver.

3.1.6.3 Use Python Virtual Environments with PySpark

To ensure that the two Python versions match, in case your components run on different machines, you must use the Python virtual environments with PySpark.

Create a Virtual Environment with Conda

> **Note:**
>
> You can also use **virtualenv** to create your virtual environment instead of **conda**.

To create a virtual environment with Conda, follow these steps:

1. Ensure that you have conda and conda-Pack installed.

   > **Note:**
   >
   > To check if conda is installed, then execute the following command: `"conda --version"`

2. Create your virtual environment using the following command:

   ```
   conda create -y -n <environment-name> python=<python-version>
   <additional-packages>
   ```

   > **Note:**
   >
   > The <environment-name> can be chosen freely and subsequently has to be substituted in further commands.

3. Activate your virtual environment using the following command:

   ```
   conda activate <environment-name>
   ```

4. Execute the following to obtain the path to your virtual environment:

   ```
   which python
   ```

   The obtained result is referred to as `<environment-abs-path>`.

5. Compress your virtual environment using the following command:

   ```
   conda pack -n <environment-name> -o <environment-abs-path>/
   <environmentname>.
   tar.gz
   ```

Update Interpreter Properties

The interpreter properties can either be configured in the interpreter JSON files or from the Interpreters page of the Compliance Studio application UI after starting the Compliance Studio application.

- In the Spark Interpreter Settings page of the Compliance Studio application UI (or `spark.json`), change the following:
  - Change the value of the `spark.yarn.dist.archives` property to `<environment-abspath>/< environment-name>.tar.gz#<environment-name>`
  - Change the value of the `spark.pyspark.python property to ./<environmentname>/bin/python`
- In the **PySpark Interpreter Settings** page of the Compliance Studio application UI (or `pyspark.json`), change the value of the `zeppelin.pyspark.python` parameter to `<environment-abs-path>/bin/python`.

3.1.6.4 Configure 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 postinstallation 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 the Interpreter Settings page, expand **Interpreter Client Configurations** and click the 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 following table

Table 3-5 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

3.1.6.5 Enable Additional PySpark Interpreter

Interpreter variant does not apply to pyspark interpreters. Hence, you must enable an additional set of interpreters.

To enable an additional pyspark interpreter, see Enable Additional Spark or PySpark interpreter section in the Appendix.

3.2 Create a Credential

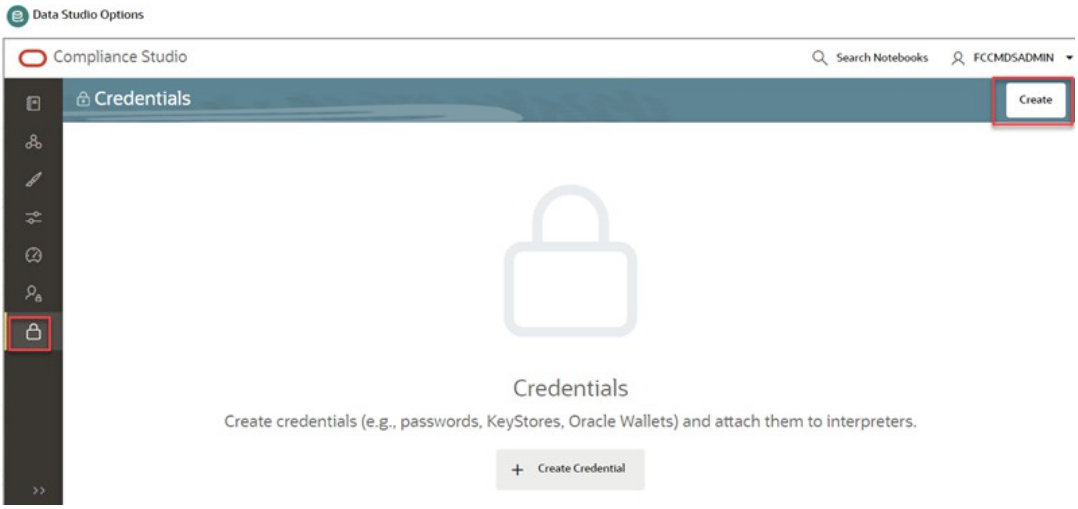
New credentials are created when database details are changed or updated. For example, change in Transparent Network Substrate (TNS) due to hostname change or compulsory periodic update of schema passwords.

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.

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

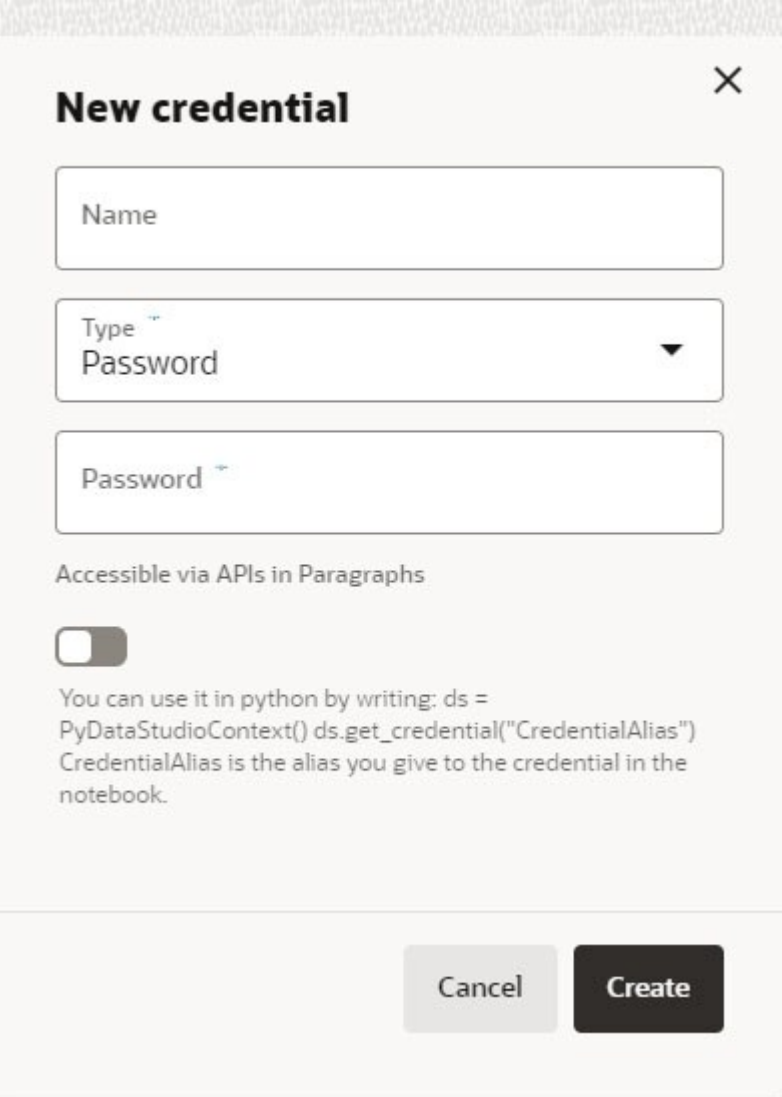
1. On the Compliance Studio workspace LHS Menu, click **Credentials**. The Credentials page is displayed.

Figure 3-15 Credentials Page



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

Figure 3-16 New Credential for Password



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

Table 3-6 Create Credential dialog

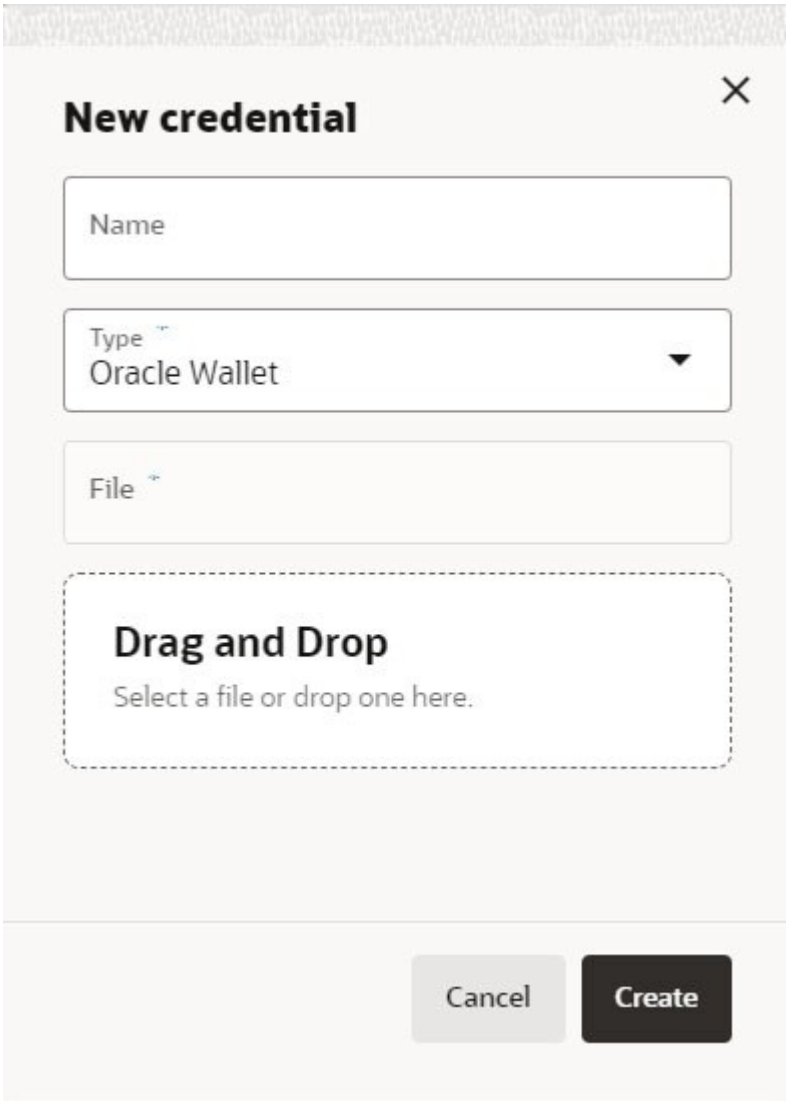
Field	Description
Name	Enter the name for the password credential.
Type	From the drop-down list, select the Password type.
Password	Enter the wallet password for the password credential.
Accessible via APIs in Paragraphs	Move this toggle switch to right to enable this option.

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 3-17 New Credential for Wallet



2. Enter the following information in the New credential dialog as tabulated in the following table.

Table 3-7 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 3-7 (Cont.) Create Credential dialog box

Field	Description
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.

3.3 Link Credentials

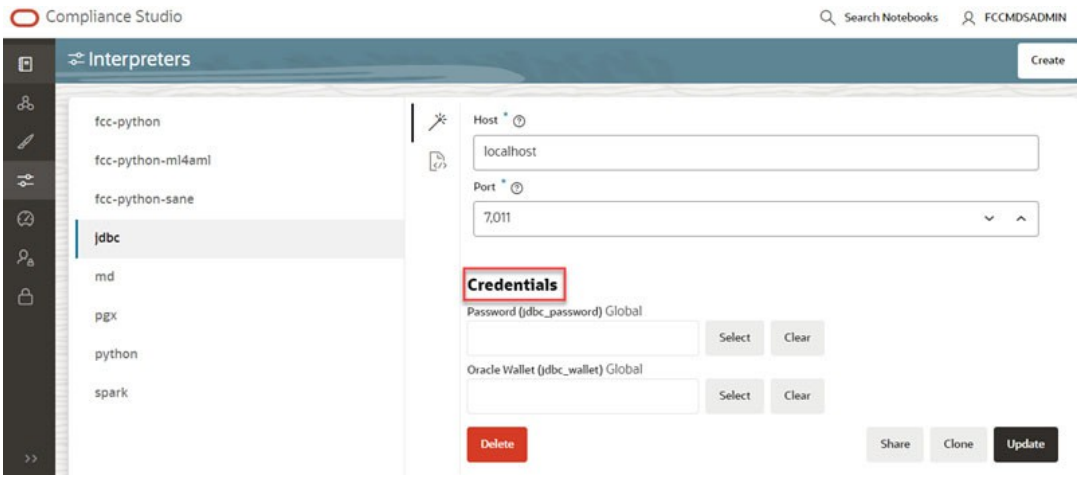
Compliance Studio provides secure and safe credential management. Examples for credentials are passwords, Oracle Wallets, or KeyStores. To link credentials (a wallet and a password) to the jdbc interpreter variant to enable secure data access. This linking enables the jdbc interpreter to securely connect 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:

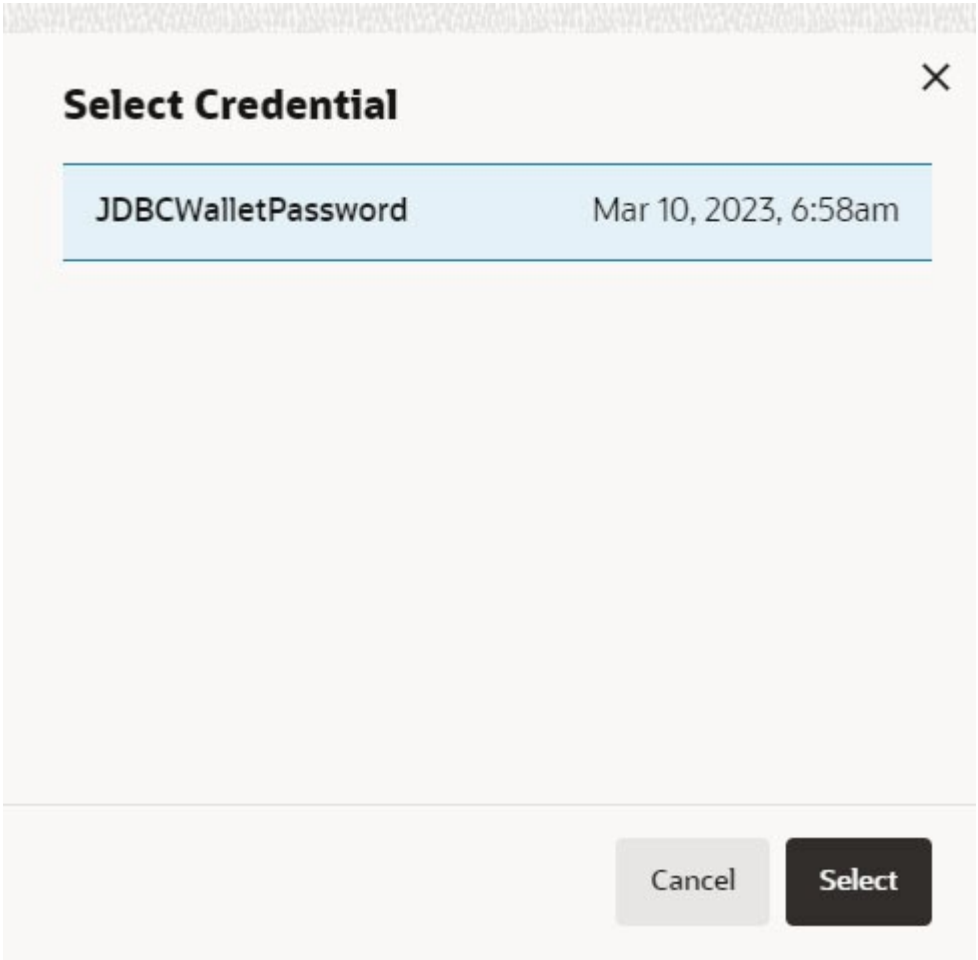
1. On the Interpreters page, select the required interpreters. For example, jdbc.
2. Navigate to the Credentials section.

Figure 3-18 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 3-19 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.

3.4 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:

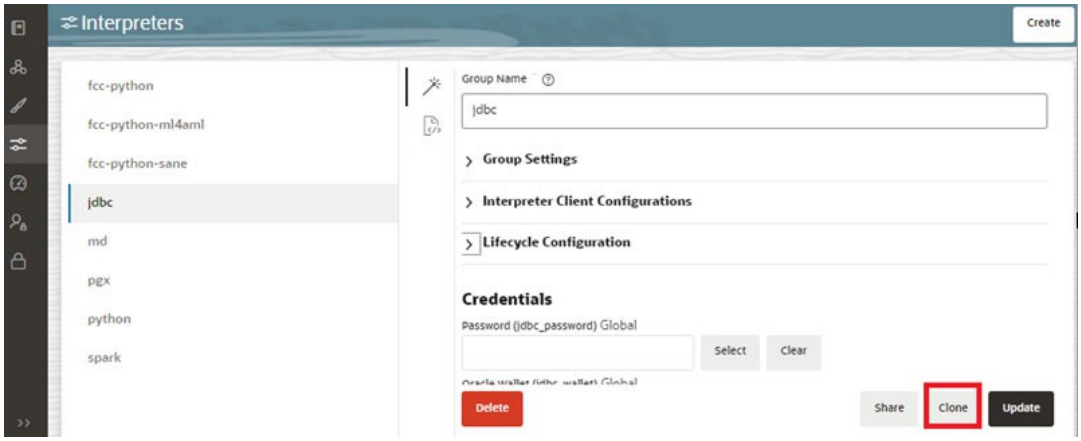
1. On the Interpreters page, click the required interpreters from the LHS list. For example, jdbc interpreter.
2. The default interpreter group is displayed on the RHS.
3. On the default interpreter, click **Clone** button to create a new group. The Create Interpreter Group dialog box is displayed.
4. 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.

3.5 Create an Interpreter Variant

1. Log in to the Compliance Studio application.
2. Launch the **CS Production** Workspace.
3. Click 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 3-20 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.

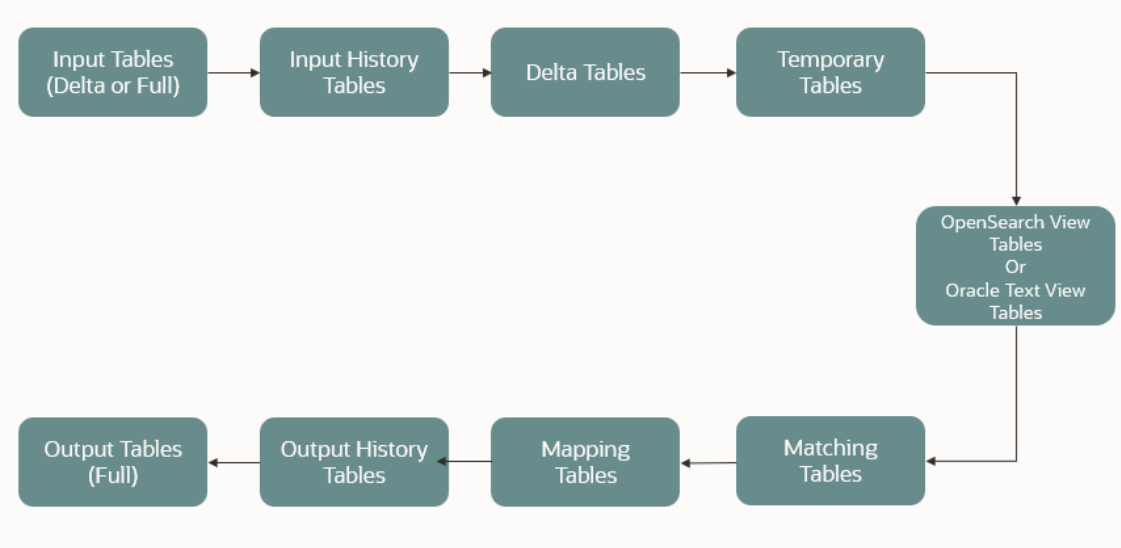
4

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 4-1 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**
  Scoring on all pairs of records is not performant, so the Entity Resolution process first finds candidates with similar attributes and only scores on those pairs of records. Candidate Selection can either be run using Oracle OpenSearch or in the database using Oracle Text (OT).
- **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.

- **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.
- **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.
- **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.

> **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.

4.1 Using Pre-configured Datasets and Rulesets

The section explains about using the Pre-configured Datasets and Rulesets.

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

The application provides pre-configured rulesets/rules for Matching, Merging, and Data Survival for Entity Resolution pipeline (**CSA_8128**).

> **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 outof- the-box ER pipeline.
> - Pre-configured rulesets are supported only when the matching mechanism is selected as “OT”. Users can use “OS” as a matching mechanism for any custom ruleset.

4.1.2 Custom Rulesets for Matching

> **Note:**
>
> Custom rulesets for matching are not supported when MATCHING_MECHANISM is set to OT. Contact My Oracle Support (MOS) to create custom rulesets with Oracle Text.

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 4-2 Sample Snapshot for Custom Rulesets of Matching

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

This section explains about the FCCM out-of-the-box Entity Resolution Pipeline on FSDF.

4.2.1 Pre-configured Entity Resolution Pipelines

The application is pre-configured to support the Entity Resolution pipeline (CSA_8128).

> **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 Metadata Tables for Entity Resolution 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.

4.2.2 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.

4.2.2.1 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_81280.ODM` file is applicable only for Behavior Detection 8.1.2.8.0 version and CSA_8128 pipeline.

To upload the data model, follow these steps:

1. Copy `ER_81280.ODM` from `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/entityresolution/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 OFS Analytical Applications Infrastructure User Guide.

4.2.3 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.
>
> – 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.

4.2.4 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, 8126, 8128 and 8128.

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

4.2.5 Entity Resolution Mapping Information

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

The following table describes column details in the FCC_ER_MAPPING.

Table 4-1 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	Information about **V_ACTION** column, see the following section.
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.

Table 4-1 (Cont.) FCC_ER_MAPPING Details

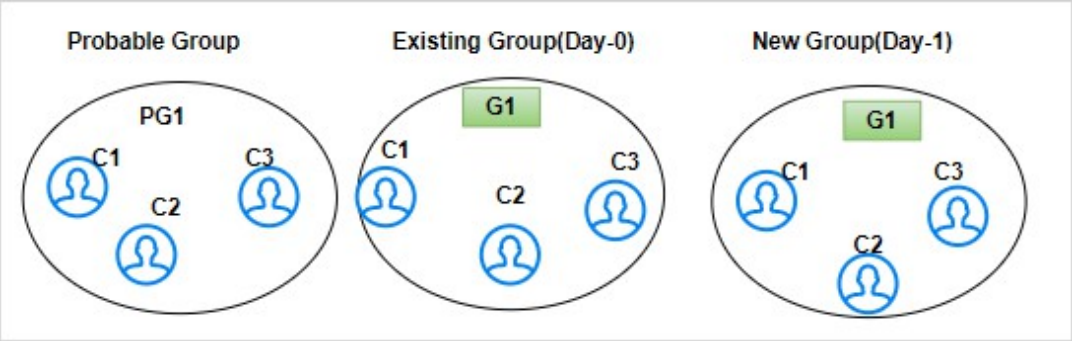
Column Name	Description
F_OVERRIDE_FLAG	This flag controls whether to override the manual decision or not irrespective of the V_MD_FLAG value. By default it should be null.
V_MD_FLAG	It stores the state of the records upon which manual actions are taken **from Manual Decision UI** and **Merge and Split 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**.
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.
N_CREATED_RUN_SKEY	It contains runskey on which the global party was created. This column would be null if the global party was created as a result of manual action taken from the Merge and Split Global Entities UI.

The following section describes V_ACTION column in the FCC_ER_MAPPING.

V_ACTION Details for 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.

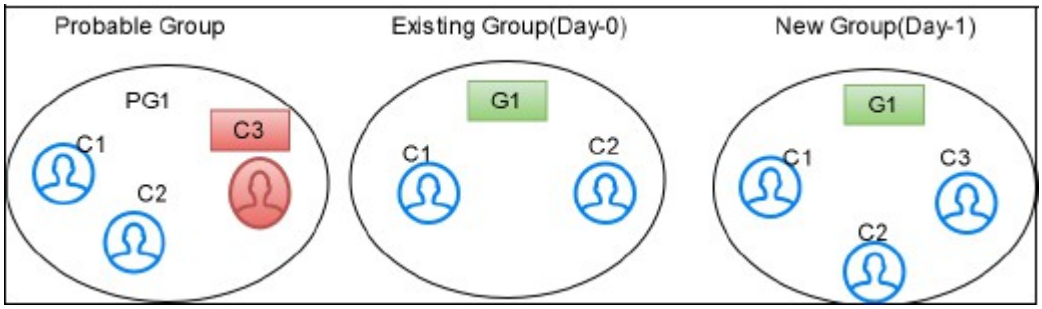
Figure 4-3 New Global Party



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.

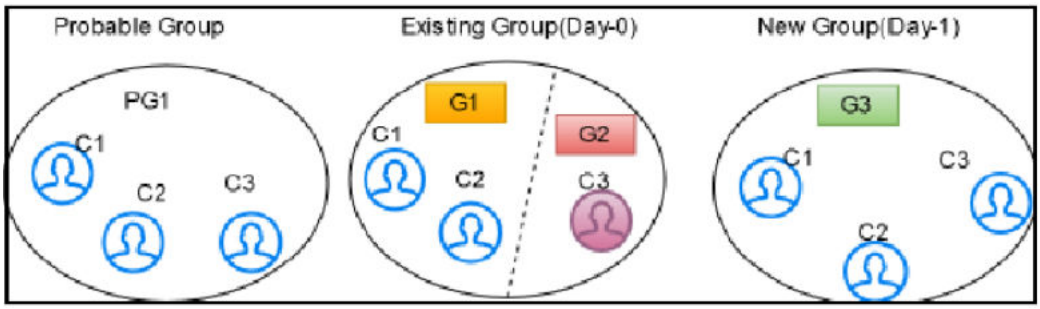
- **add**: If a new entity is available and matches the existing group, then it is defined as **add** in the V_ACTION column for a newly added entity. If a new entity matches the existing group, it will be added to the existing group and assigned the same global id.

Figure 4-4 Add



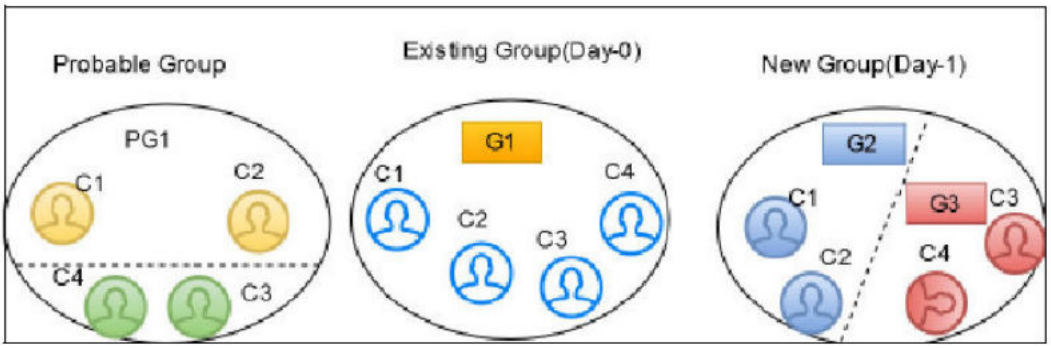
For example, G1 has C1 and C2 entities. After the Day 1 batch execution, if C3 entity matches with C1 or C2 then C3 will be added to the existing group G1 with the same global id.

- **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.



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.

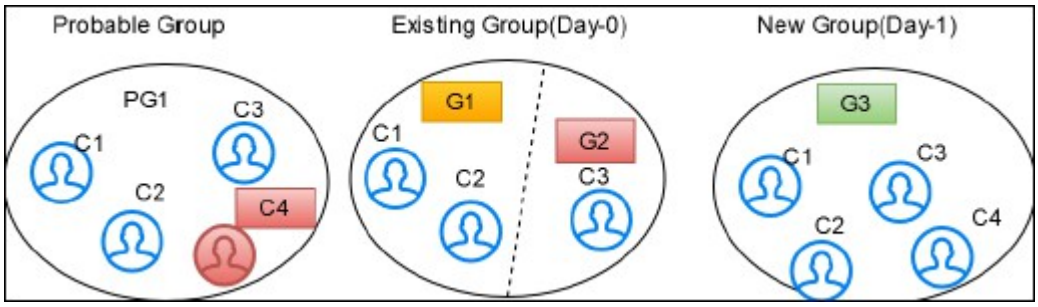
- **split**: If there is a data change in the existing group entity which does not matches with other entities of an existing group; then it is defined as **split** in the V_ACTION column for the split entities. The changed entities can be split into a new group and a new global id is assigned to each.



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.

- **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.

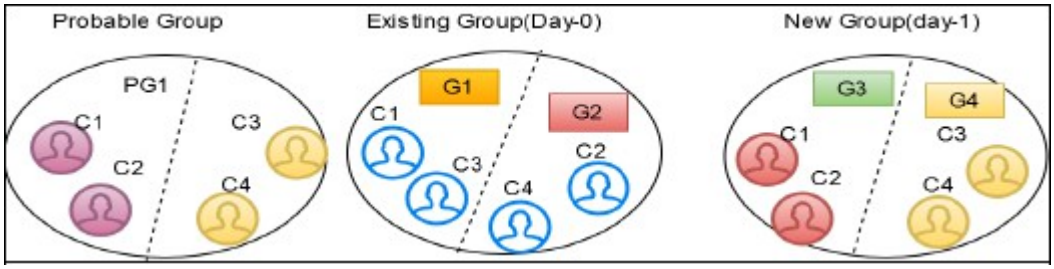
Figure 4-5 Merge and Add



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 4-6 Split and Merge

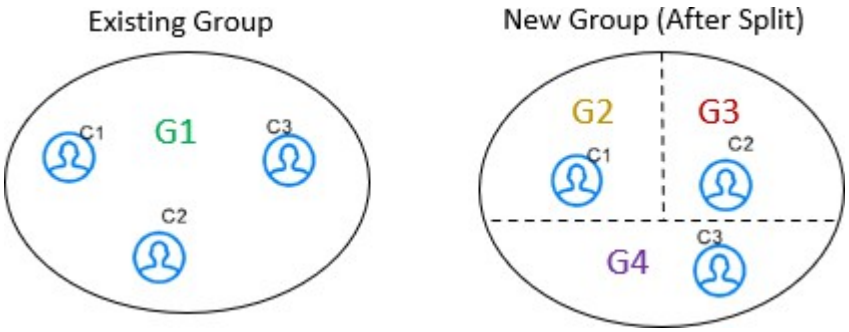


For example, G1 has C1 and C3 entities and G2 has C2 and C4 entities. After the Day 1 batch execution, if C1 matches with C2 and C3 matches with C4 then C2 and C4 will be split separately and merged with C1 and C3 respectively. G3 has C1 and C2 entities and G4 has C3 and C4 entities with a new global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

V_ACTION Details for Manual Action

- **split**: You can split the entities into different groups with new global ids assigned to each.

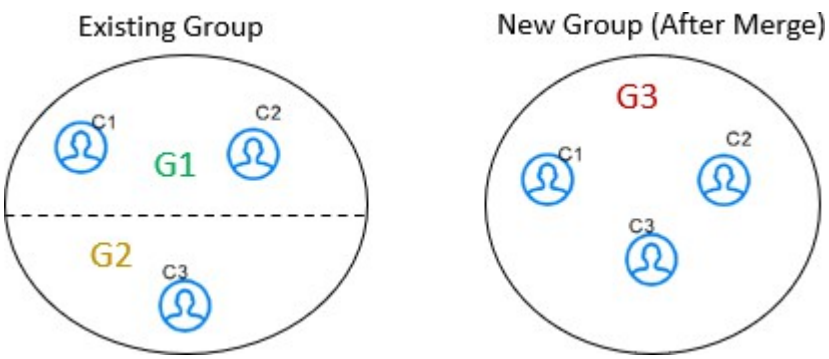
Figure 4-7 Split



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.

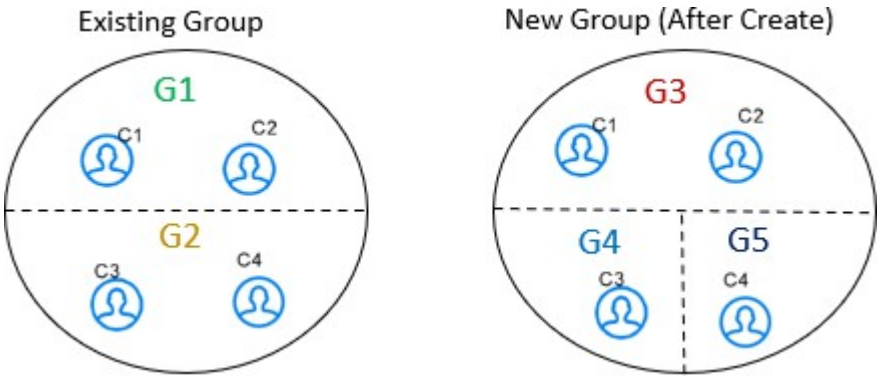
Figure 4-8 Merge



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.

- **create**: You can create a new entity from the existing group with a new global id is assigned.

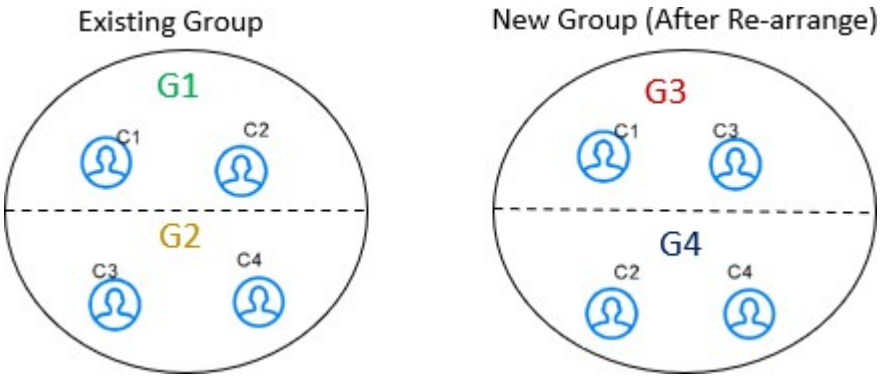
Figure 4-9 Create



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.

- **rearrange**: You can rearrange the entities from another group with a new global id is assigned.

Figure 4-10 Re-arrange



For example, G1 has C1 and C2 entities, G2 has C3 and C4 entities. After rearrange, 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 rearrange and G1 and G2 will be deactivated.

4.2.6 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.

4.3 Prerequisites when MATCHING_MECHANISM is Set to Oracle Text (OT)

Before executing the ER jobs when matching mechanism is selected as “OT”, follow these steps:

1. Enable the required rulesets.
2. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/candidateselection/utility/bin` directory.
3. Initialize the ER schema by executing the following command:

   ```
   ./CreateMViewAndIndex.sh <DATA_SCHEMA_ALIAS> <PIPELINE_ID>
   ```

   For example: `./CreateMViewAndIndex.sh ER_SCHEMA_ALIAS CSA_8128`
4. Execute the following command:

   ```
   ./CreateDBThesaurus.sh <DATA_SCHEMA_ALIAS> <PATH TO STORE PRE-PROCESSED FILES GENERATED BY UTILITY> <MODE>
   ```

   For example: `./CreateDBThesaurus.sh ER_SCHEMA_ALIAS /user/thesaurusFiles CREATE`

   The script has two options:

   - **Create**: This option helps to generate the pre-seeded thesaurus in the database.
   - **Reset**: This option helps the user to update the pre-existing thesaurus. If there is any change in the data, the user can run the script with a reset flag, and the thesaurus will be updated.

> **Note:**
>
> Only one thesaurus can be created in one Database server with the specified thesaurus name.

4.4 Executing the ER Jobs

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

For more information, see the Entity Resolution Use Case section in the OFS Compliance Studio Installation Guide.

> **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.

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.

4.4.1 Create Index and Load the Data

> **Note:**
>
> When the **MATCHING_MECHANISM** parameter is OS, ensure that you have configured the **Logstash** parameter as **true** (`index.logstashconf.apply`) in the load-to-open-search `application.properties` to load data from the Database.

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.

Configuration for Create Index and Load the Data

Full View Table (FCC_ER_FULL) Initrans: A high number of parallel processes require a table to have a higher INITRANS value. The maximum number of parallel processes during a MERGE operation on the FCC_ER_FULL can be configured using SINGLETON_TASK_PARALLEL_LEVEL parameter.

To configure SINGLETON_TASK_PARALLEL_LEVEL parameter, see the Additional Configurations section.

Initrans of the FCC_ER_FULL can be configured by changing the metadata. To update the metadata, follow these steps:

1. Update the metadata under **V_MAKE_TABLE_QUERIES** column in the **FCC_STUDIO_ER_QUERIES** table in Studio Schema for the active ER pipeline. For example, CSA_812.
2. Select V_MAKE_TABLE_QUERIES from the fcc_studio_er_queries where DF_NAME= '<ACTIVE ER DF_NAME>' and V_PIPELINE_ID = '<ACTIVE ER PIPELINE ID>';
   For example:

   Select V_MAKE_TABLE_QUERIES from fcc_studio_er_queries where DF_NAME= 'Customer812' and V_PIPELINE_ID = 'CSA_812';
3. Search for "N_CUSTOM_INITRANS NUMBER" and only set the custom value if required. For example, N_CUSTOM_INITRANS NUMBER := 50;
4. Commit the changes.

Steps

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/bin` directory.

2. Run the following command:

```
nohup ./ER_Create_And_Load_Data_Into_Index.sh "<PIPELINE_ID>"
"<ER_SCHEMA_WALLET_ALIAS>" "<LOAD_TYPE>" "<FIC_MIS_DATE>"
"<FSDF_VERSION>" "<BATCH_GROUP>" "<SOURCE_BATCH>" "<DATA_ORIGIN>"
"<RUN_TYPE>" &
```

> **Note:**
>
> - <BATCH_GROUP> refers to the FCC_PROCESSING_GROUP table 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 command for CSA_8128 pipeline.

FSDF 8128 version: `nohup ./ER_Create_And_Load_Data_Into_Index.sh "CSA_8128" "ER_SCHEMA_PP_ALIAS" "FullLoad" "20151210" "8128" "CSA_812" "CSA_812" "US" "RUN" &`

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

4.4.1.1 Additional Configurations

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

> **Note:**
>
> 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.
> – Tables with regular expression **H$STG_%_PRE_DELETED** would be excluded from this MAX_HISTORY_PARTITIONS limit.

- **DB_PARALLEL_LEVEL**: It configures a degree of parallelism for data survival (Job 3). By default, it is set to **8**, and you can modify this value to change the level of parallelism.
- **BULK_APPLY_DS_FOR_SINGLETON_PARTIES**: It configures whether data survival (Job 3) should be applied or not for singleton parties. By default, it is set to “N.” In this case, the data survival will not be applied to the singleton parties.
  If the value is set to “Y,” then data survival will be applied to the singleton parties.
- **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.
- **F_CAPTURE_COUNT_STAT**: This flag indicates count statistics to be captured during the entity resolution job execution. If it is set to true, count statistics are logged in the FCC_ER_JOB_VOL_STATS table of the ER/FSDF schema. By default, this value is set to true.

  > **Note:**
  >
  > This parameter is applicable for all the entity resolution jobs.

- **SINGLETON_TASK_PARALLEL_LEVEL**: The parameter indicates the maximum number of parallel processes spawned during DBMS PARALLEL EXECUTE. By default, the value is 8. This configuration is populated with default value when the value is not available during the Create Index and Load the Data (Job 1) immediate run. This is a onetime configuration that has to be handled for all runs which includes bulk volume runs.
- **CAN_SEL_BUCKET_SIZE**: The maximum bucket size in the source data. By default, it is 2000.

3. Save the changes.

Profiler Table

The table, E**R_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.

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.

4.4.2 Perform Matching

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.

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` directory.
2. Run the following command:

```
nohup ./ER_Run_Bulk_Similarity_Job.sh "<PIPELINE_ID>"
"<ER_SCHEMA_WALLET_ALIAS>" "<MATCH_TYPE>" "<BATCH_GROUP>" "<RUN_TYPE>" &
```

> **Note:**
>
> <BATCH_GROUP> refers to FCC_PROCESSING_GROUP table in the Compliance Studio schema.

For example, you can use the following command for CSA_8128 pipeline.

FSDF 8128 version: `nohup ./ER_Run_Bulk_Similarity_Job.sh "CSA_8128" "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.

4.4.2.1 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

4.4.2.2 Additional Configuration for Matching with Oracle Text (OT)

> **Note:**
>
> This section is applicable when MATCHING_MECHANISM is set to OT.

The source data is divided into buckets (N_BUCKET_ID) for performing candidate selection. Candidate selection is matched using a DBMS job on each bucket, and multiple buckets can be processed in parallel.

The following parameters are configured with respective pipeline id in the can_sel_ot_config table of the studio schema. For example, CSA_8128.

- **CAN_SEL_MAX_JOBS**: Maximum number of buckets that can run anytime during candidate selection. By default, the value is 35.

- **QUERY_LOG_LEVEL**: The logging level for Oracle text SQL queries for each source data. The acceptable values are:
  - **INFO**: Info level shows only failed matching queries in the CAN_SEL_OT_QUERY_LOG table. By default, it is set to INFO.
  - **DEBUG**: Debug shows all the source data SQL queries.
- **CAN_SEL_BATCH_SIZE**: The maximum bucket size in the source data. By default, it is 2000.

> **Note:**
>
> It is applicable only for graph pipelines. For Entity Resolution, see CAN_SEL_BUCKET_SIZE parameter in the Additional Configurations section.

- **BUCKET_MAX_EXEC_TIME**: The maximum time in seconds for candidate selection is executed on each bucket. By default, it is 7200.

> **Note:**
>
> - For processing a larger volume of data, increase the execution time.
> - If any buckets get timed out, the process gets terminated automatically, and the user needs to re-run the matching job.

- **PARALLEL_LEVEL**: Database parallel hint used to query data, index, materialized view creation, and materialized view refresh. By default, it is 8.
- **APPLY_TRANSLITERATION**: This flag represents the transliteration for candidate selection. By default, it is set to N.

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.

4.4.3 Data Survival

> **Note:**
>
> Ensure that only one pre-configured 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.

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.

Configuration for Data Survival

FCC_ER_QUERIES_PROPERTIES: This table is to configure hints for SQL queries and hints for data survival (Job 3) rollback query is configured. This table is created in the Studio Schema on the Compliance Studio startup.

Steps

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

```
nohup ./ER_Run_Data_Survival_Engine.sh "<PIPELINE_ID>"
"<ER_SCHEMA_WALLET_ALIAS>" "<MATCH_TYPE>" "<BATCH_GROUP>" "<RUN_TYPE>" &
```

> **Note:**
>
> <BATCH_GROUP> refers to the FCC_PROCESSING_GROUP table in the Compliance Studio schema.

For example, you can use the following command for CSA_8128 pipeline.

FSDF 8128 version:

```
nohup ./ER_Run_Data_Survival_Engine.sh "CSA_8128"
"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>`

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.

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

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.

The following table describes column/flag deatils in the FCC_ER_GUID_PERSIST_CONFIG.

Table 4-2 FCC_ER_GUID_PERSIST_CONFIG

Column Name/Flag	Description
V_ACTION	It represents the different actions that can be performed on the Global Party ID. The possible actions are: • create • rearrange • 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 in an ER batch. The valid values are Y and N. The GUID is persisted if the flag is set to Y for the particular action. **Note**: This flag is not applicable for create and rearrange actions as these are manual actions.
F_MANUAL_APPROVAL	This flag represents manual approval is required when GUID undergoes change in an ER batch. 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.
F_PERSIST_MANUAL_ACTI ON	This flag represents whether the Global Party ID can be persisted or not through manual action. The valid values are Y and N. The GUID is persisted if the flag is set to Y for the particular action. If the flag is set to N, then new Global Party ID will be created. **Note**: This flag is applicable only for Merge, Split, Create and Rearrange actions.

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 4-11 fcc_er_guid_persist_config table

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

4.4.4 Load Data in FCC_ER_OUTPUT Table

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.

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` directory.
2. Run the following command:

```
nohup ./ER_Run_Full_Data_Output.sh "<PIPELINE_ID>"
"<ER_SCHEMA_WALLET_ALIAS>" "<FIC_MIS_DATE>" "<BATCH_GROUP>"
"<LOAD_TYPE>" "<RUN_TYPE>" &
```

> **Note:**
>
> <BATCH_GROUP> refers to the FCC_PROCESSING_GROUP table in the Compliance Studio schema.

FSDF 8128 version: `nohup ./ER_Run_Full_Data_Output.sh "CSA_8128" "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.

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.

4.4.5 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.

4.4.6 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 4-3 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

4.4.7 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`)

Steps

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

```
nohup ./Wrapper_Run_ER.sh "<PIPELINE_ID>" "<ER_SCHEMA_WALLET_ALIAS>"
"<LOAD_TYPE>" "<FIC_MIS_DATE>" "<FSDF_VERSION>" "<CURRENT_BATCH>"
"<SOURCE_BATCH>" "<DATA_ORIGIN>" "<ER_SCHEMA_NAME>" "<RUN_TYPE>" &
```

> **Note:**
>
> - <CURRENT_BATCH> refers to the FCC_PROCESSING_GROUP table 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 command for CSA_8128 version:

```
nohup ./Wrapper_Run_ER.sh "CSA_8128" "ER_SCHEMA_PP_ALIAS" "FullLoad"
"20151210" "8128" "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

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.

4.5 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.

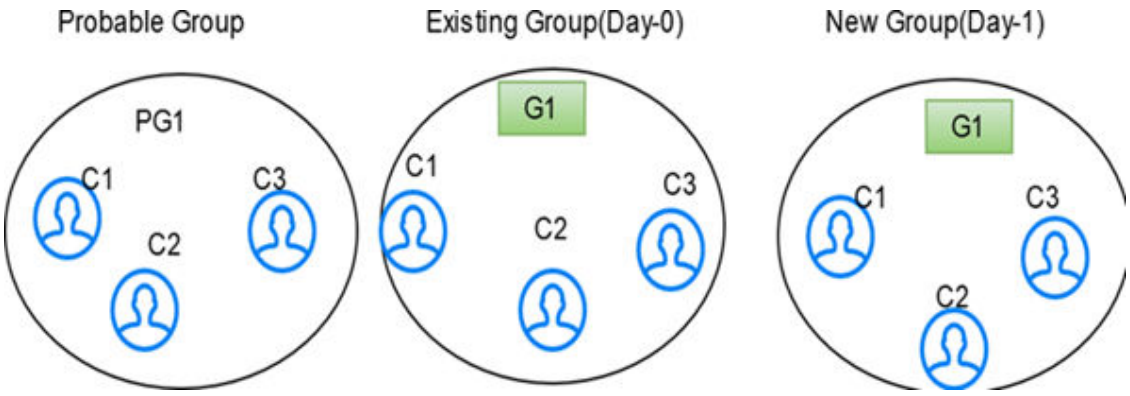
4.5.1 Persisting the Data When F_PERSIST_GUID and F_MANUAL_APPROVAL Flags are Set to False Condition

> **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.

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.

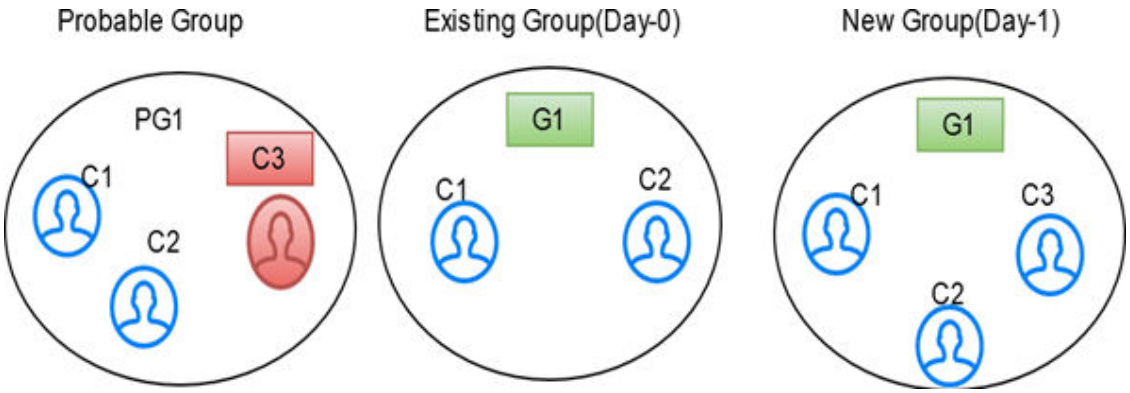
Figure 4-12 No change



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 4-13 Add

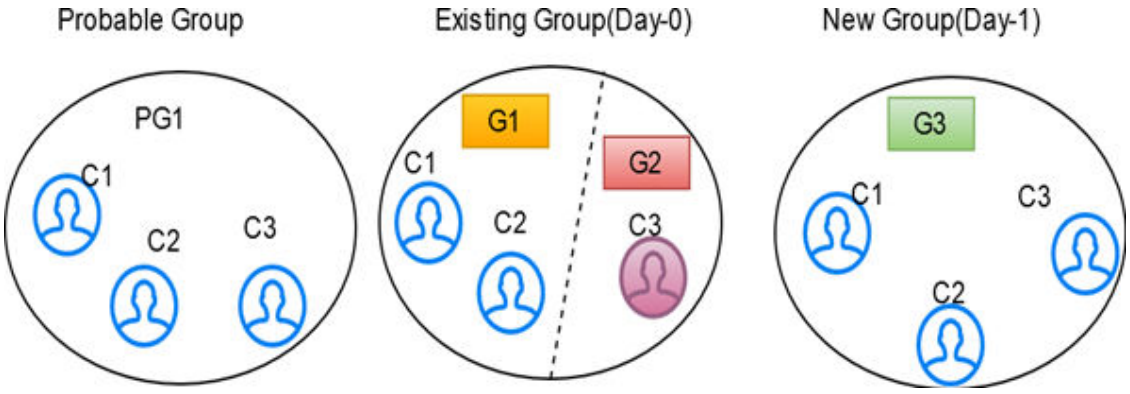


Merge
Existing group elements are a subset of probable group elements, and the number of elements is the same in both groups. Elements in the existing Group have different global ids assigned.

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 4-14 Merge

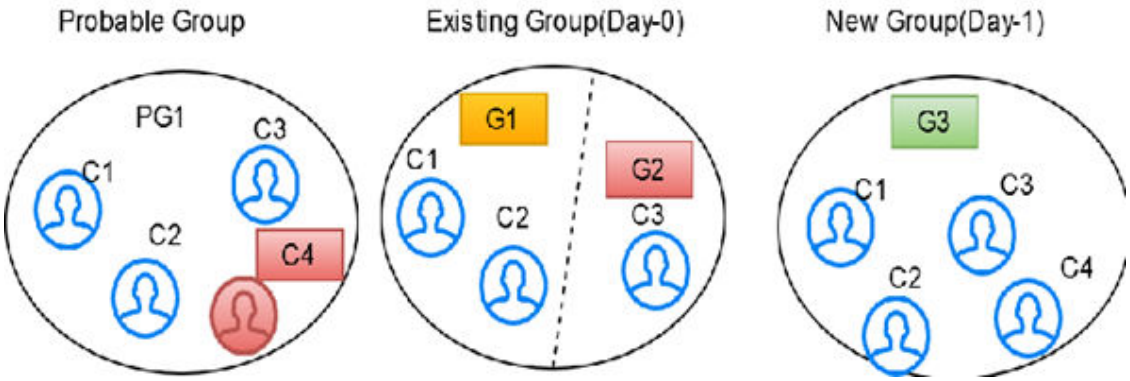


Merge and Add
Existing group elements are a subset of probable group elements, and the number of elements in the probable Group is more than the existing Group. Extra elements in the probable Group don't have any global id assigned yet, and standard elements have different global IDs

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.

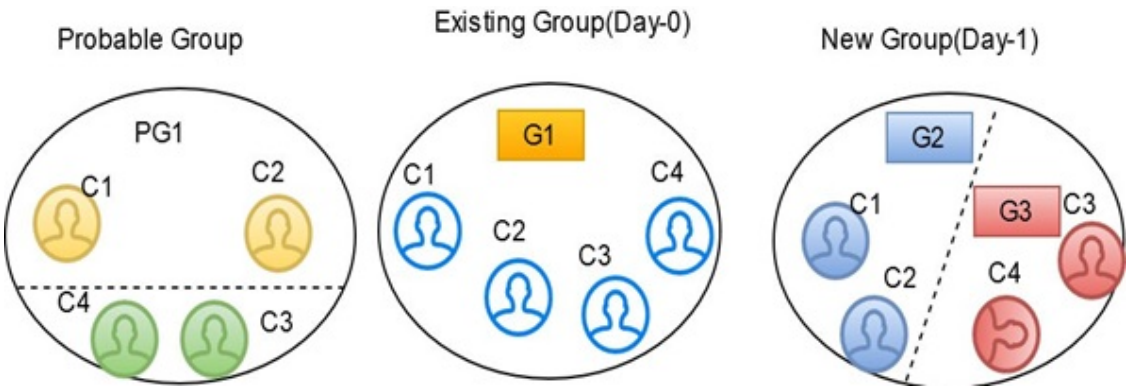
Figure 4-15 Merge and Add



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 4-16 Split

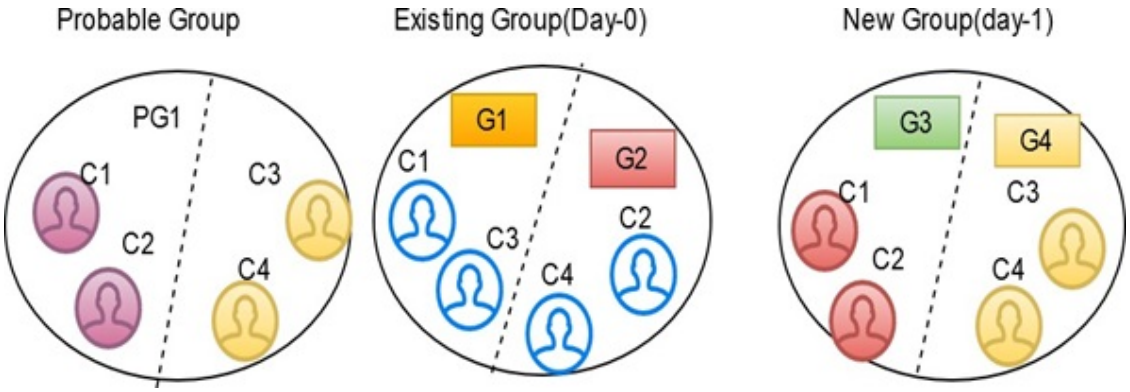


Split and Merge
After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements

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 4-17 Split and Merge

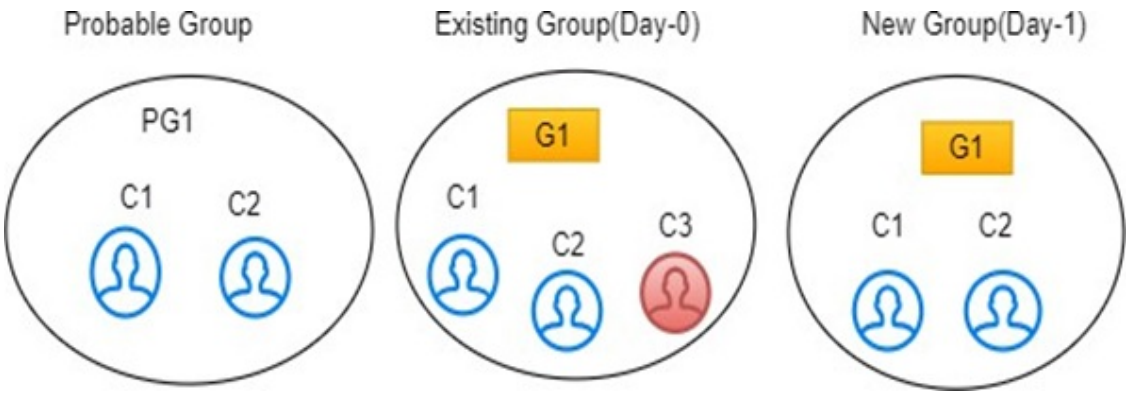


Delete

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 4-18 Delete



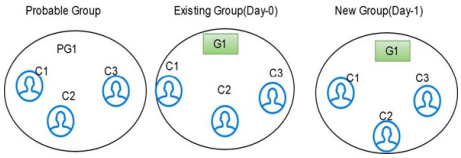
4.5.2 Persisting the Data When F_PERSIST_GUID Flag is Set to True and F_MANUAL_APPROVAL Flag is Set to True/False Condition

> **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).

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.

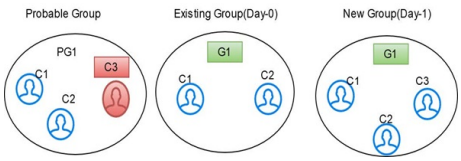
Figure 4-19 No change



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 ma

Figure 4-20 Add



Merge

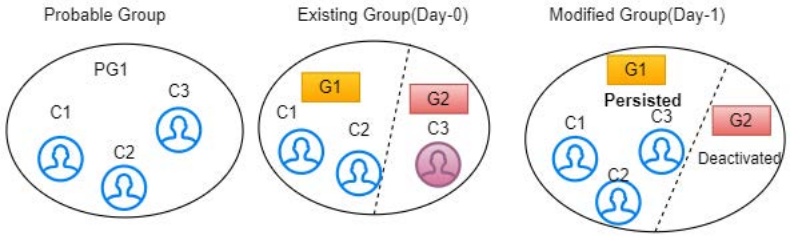
Existing group elements are a subset of probable group elements, and the number of elements is the same in both groups. Elements in the existing Group have different global ids assigned. Elements are merged into a single group, and the existing global id is persisted.

> **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 P**ersisting 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.

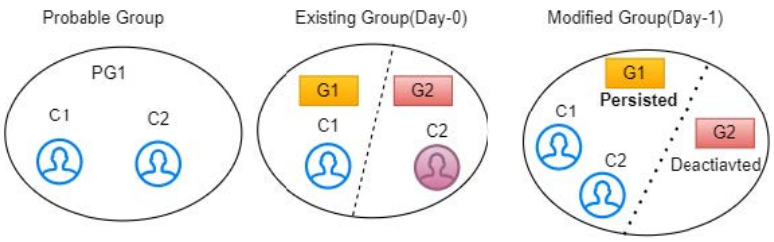
Figure 4-21 Merge Action for most Number of Entities



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 4-22 Merge Action for Lowest Global ID



Merge and Add

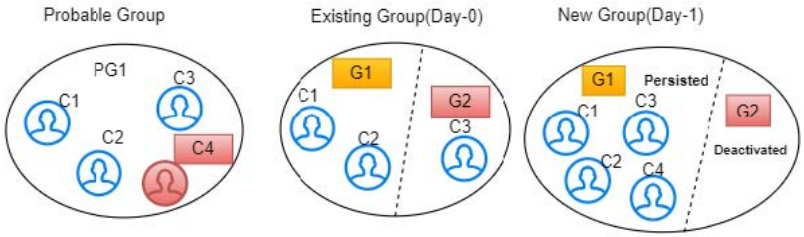
Existing group elements are a subset of probable group elements, and the number of elements in the probable Group is more than the existing Group. Extra elements in the probable Group do not have any global id assigned yet, and standard elements have different global IDs assigned already. Common elements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

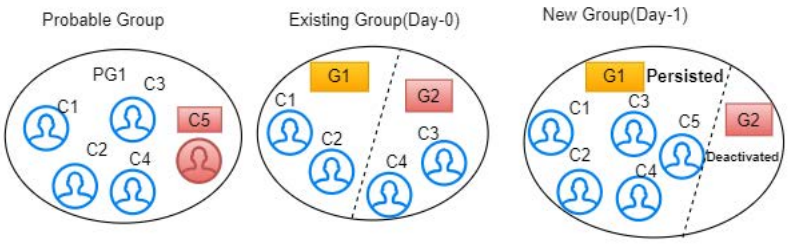
Figure 4-23 Merge and Add Action for most Number of Entities



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 4-24 Merge and Add Action for Lowest Global ID



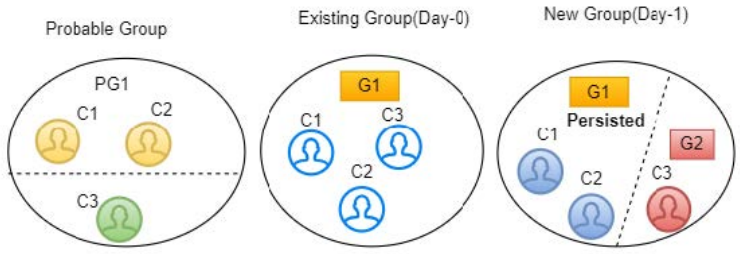
Split
After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements in both probable and existing groups is the same. A single global id is assigned to all elements in the existing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

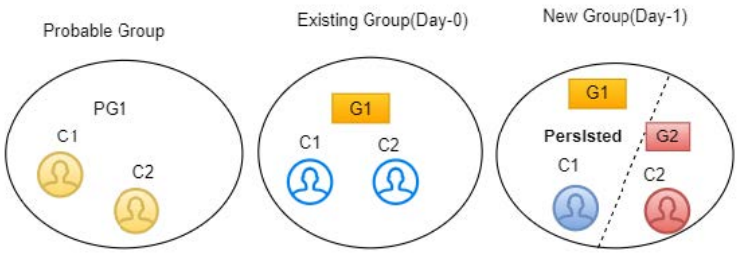
Figure 4-25 Split Action for most Number of Entities



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 (regardless which entity was changed) 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 4-26 Split Action for Lowest Global ID



Split and Merge

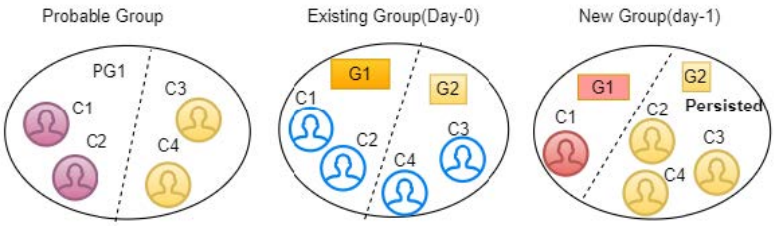
After applying merging rules criteria, if multiple groups are created for elements of a probable group, these elements are also a subset of existing group elements. The number of elements in both probable and existing groups is the same, and different global ids are assigned to elements in the existing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

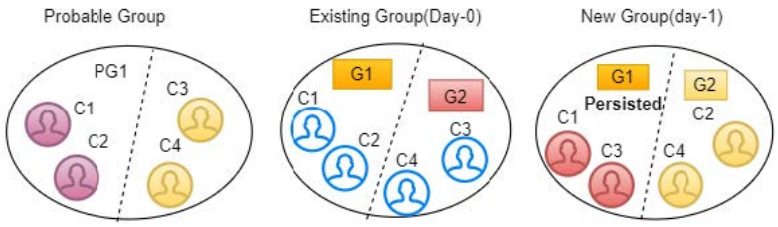
Figure 4-27 Split and Merge Action for more Number of Entities



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.

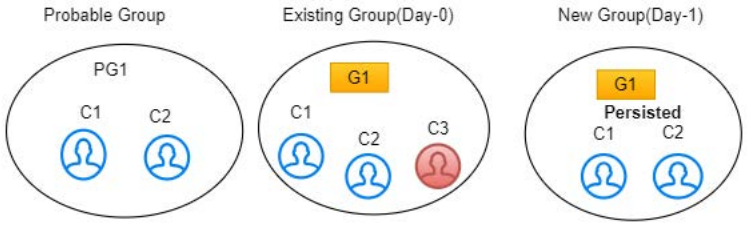
Figure 4-28 Split and Merge Action for Least Global ID



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 4-29 Delete



4.6 Metadata Tables for Entity Resolution

Metadata tables manage operation for the Entity Resolution jobs.

4.6.1 Default Data in the tables

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_T**L: 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.
  If users want to add custom Data Survival method, follow these steps:

  1. Open the Compliance Studio schema.

2. In the **FCC_DATASURV_TYPE** table, add a new row and update the following:
   a. **N_TYPE_ID**: Provide the numerical value based on the existing sequence order. For example, this FCC_DATASURV_TYPE table having 10 ids already then you should provide N_TYPE_ID as 11 for the new custom method.
   b. **V_TYPE_NAME**: Provide the name for the custom method. This name will be displayed in the Data Survival's Type drop-down list in the Compliance Studio UI.
   c. **V_TYPE_CD**: Provide logical name for the custom method.

      > **Note:**
      >
      > For the logical name, special characters are not allowed except underscore (_) and should not contain any spaces.

   d. **F_IS_CUSTOM_TYPE**: Set the value as "T".
   e. **N_SEQ_ID:** Provide the numerical value based on the existing sequence order. For example, this FCC_DATASURV_TYPE table having 10 sequences already and you should provide N_SEQ_ID as 11 for the new custom method.
3. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/entity-resolution/ extensions/data-survival` directory.
4. Open the `UserDefinedMethods.py` file and add the custom function inside the class UserDefinedMethods.
   For example, if you are adding custom function as **gender_criteria** then update as follows:

```
class UserDefinedMethods:
           def method(self, entity_list, func_name,
entity_constraint=None,
entity_index=0):
          func = getattr(self, func_name)
          return func(entity_list, entity_constraint, entity_index)

          def gender_criteria(self, entity_list, entity_constraint=None,
entity_index=0):
          if all(elem == 'A' for elem in entity_list):
             return 'A'
          elif all(elem == 'B' for elem in entity_list):
          log.info("Inside B {}".format(entity_list))
          return 'B'
    else:
     return 'U'
```

> **Note:**
>
> - The custom method should be added in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/entityresolution/extensions/data-survival/ UserDefinedMethods.py` file as well. In case of reinstallation; the custom method is preserved.
> - The method name for the custom data survival type in the python file should be the same as the value given for V_TYPE_CD column in the FCC_DATASURV_TYPE table.

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 4-30 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 4-31 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.

4.6.2 Customize Data in ER Tables

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.

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

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.

4.6.3 Populate Metadata for Data Survival in the 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 4-4 Metadata

v_metadata_type	v_column_cd	n_precedence
Occupation	teacher	2
Geo-location	US	3

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"
}
]
```

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"
}
]
```

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

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

4.7 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 the deleted parties id.

> **Note:**
>
> If one or more entities are deleted from the global party ID, the V_ACTION column retains its value for the remaining entities.
>
> For example, consider G1, which has C1, C2, and C3 entities that were merged earlier. After the Day 1 batch execution, if entity C3 is deleted, the V_ACTION column for entities C1 and C2 will still show as "Merge".

4.7.1 Impact on Manual Decisioning for 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

4.8 Ability to Remove Split and Merge Action 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.

4.9 Expiry of Entity Child Records Mapping

Expiry of child records mapping is the process where relationship between a parent record (Global Party ID) and its associated child records (Address, Phone, Email, and document) is no longer considered as valid after certain period of time. The expiry can be due to several reasons, such as data aging, changes in parent record, or updates in the underlying data that requires re-evaluation of the entity mappings.

4.9.1 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 or equal to fic_mis_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 be loaded into the 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).

4.9.2 Expiry of Entity Phone Mapping

> **Note:**
>
> This functionality is available only on the new pipeline (CSA_8128). To uptake new feature/enhancement on new pipeline from an older pipeline, contact My Oracle Support (MOS).

If phone mapped to the parties is to be removed from the system, then set the D_RECORD_END_DATE attribute as a date lesser than or equal to fic_mis_date in the STG_PARTY_PHONE_MAP_PRE table. This will remove the phone mapping as part of the Entity Resolution batch run from the STG_PARTY_PHONE_MAP table.

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

4.9.3 Expiry of Entity Email Mapping

> **Note:**
>
> This functionality is available only on the new pipeline (CSA_8128). To uptake new feature/enhancement on new pipeline from an older pipeline, contact My Oracle Support (MOS).

If an email mapped to the parties is to be removed from the system, then set the D_RECORD_END_DATE attribute as a date lesser than or equal to fic_mis_date in the

STG_PARTY_EMAIL_MAP_PRE table. This will remove an email mapping as part of the Entity Resolution batch run from the STG_PARTY_EMAIL_MAP table.

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

4.9.4 Expiry of Entity Document Mapping

> **Note:**
>
> This functionality is available only on the new pipeline (CSA_8128). To uptake new feature/enhancement on new pipeline from an older pipeline, contact My Oracle Support (MOS).

If document mapped to the parties is to be removed from the system, then set the D_RECORD_END_DATE attribute as a date lesser than or equal to fic_mis_date in the STG_CUSTOMER_IDENTIFCTN_DOC_PRE table. This will remove document mapping as part of the Entity Resolution batch run from the STG_CUSTOMER_IDENTIFCTN_DOC table.

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

4.10 Statistics for ER Job Execution

Users can capture execution time and data volume statistics for Entity Resolution jobs to be able to monitor performance and tracking support job. The start time, end time and total time is logged for each individual ER Job as well as each logical step within an ER Job. The statistics are logged into the respective job level and step level in the following tables every time when job execution is triggered with valid input parameters.

Volume statistics can be disabled by setting F_CAPTURE_COUNT_STAT= N. By default, this value is set to true in the FCC_ER_CONFIG table. For more information, see the Additional Configurations section.

The following tables capture time and volume statistics for all the ER Job execution.

- **FCC_ER_JOB_STATS**: This table is present in the ER Schema and it contains start timestamp, end timestamp, total time taken and status for each ER job execution corresponding to fic_mis_date, runskey, job id and sequence id. N_JOB_ID column is an identifier for the ER Jobs. For example, if it is 1 in the N_JOB_ID column, then it indicates for ER Job 1 (Create Index and Load the Data).
- **FCC_ER_JOB_STEP_STATS**: This table is present in the ER Schema and it contains start timestamp , end timestamp, total time taken and status for each step in the ER job execution corresponding to fic_mis_date, runskey , job id and sequence id.
- **FCC_ER_JOB_VOL_STATS**: This table is present in the ER Schema and it contains log for data volume statistics pertaining to an ER job.
- **FCC_ER_JOB_STATS_QUERIES**: This table is present in the Studio Schema and it contains the pre-seeded metadata queries that are executed to capture volume statistics during the ER job execution and the pre-seeded queries are marked with job id. The volume queries having N_STEP_ID value that are executed within the particular job step while others are executed on successful execution of the ER Job. The queries are marked

against the key name (V_KEY) as an identifier for the volume query. The output of these volume queries is captured in FCC_ER_JOB_VOL_STATS against the fic_mis_date, runskey and sequence id.

- **FCC_ER_JOB_STEP_DESCRIPTION**: This table is present in Studio Schema and it contains description of each of step id mentioned in the ER Statistics tables (FCC_ER_JOB_STATS, FCC_ER_JOB_STEP_STATS).

5
Use Cases

The following use cases are supported in the Compliance Studio application:

- **Automated Scenario Calibration (ASC)**
- **Behavioral Model**
- **Sanctions Event Scoring**
- **AML Event Scoring**
- **Customer Segmentation and Anomaly Detection**
- **Customer Risk Scoring**
- **Shell Account Detection Scenario for AML**
- **Custom Scenario**

5.1 Automated Scenario Calibration (ASC)

This section describes about Automated Scenario Calibration (ASC).

Prerequisites
Before creating the ASC workspace, the user should follow these steps:

1. The target schema used for the ASC workspace should be a valid BD atomic schema like BD **preprod**, BD **UAT**, BD **Dev**, etc., because we use BD packages and functionality to reproduce alerts as in BD.
2. Create the Tablespace
3. Assign grants to the ASC BD Schema
4. Create a new data store for ASC BD schema

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

To create the data store, see How to Create Data Store section.

For more information on creating tablespace and assign grants to sandbox (ASC BD) schema, see the OFS Compliance Studio Installation Guide.

5.1.1 Creating ASC Workspace

On the **Workspace Summary** page, click **Add Workspace**. The Create Workspace window is displayed with the following process:

1. Basic Details
2. Workspace Schema

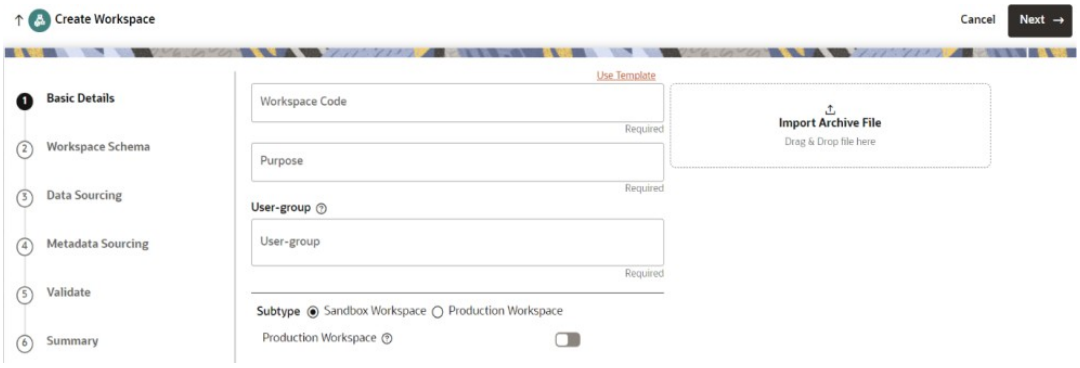
3. Data Sourcing
4. Metadata Sourcing
5. Validate
6. Summary

Basic Details

To create a basic details of the workspace, follow these steps:

1. Enter the **Workspace Code** and **Purpose** of the workspace.
2. Select the **User-group** from the drop-down list.
3. Select the subtype as **Sandbox Workspace**.
   By default, the **Production Workspace** is disabled.
4. Click **Next**.

Figure 5-1 Basic Details



Workspace Schema

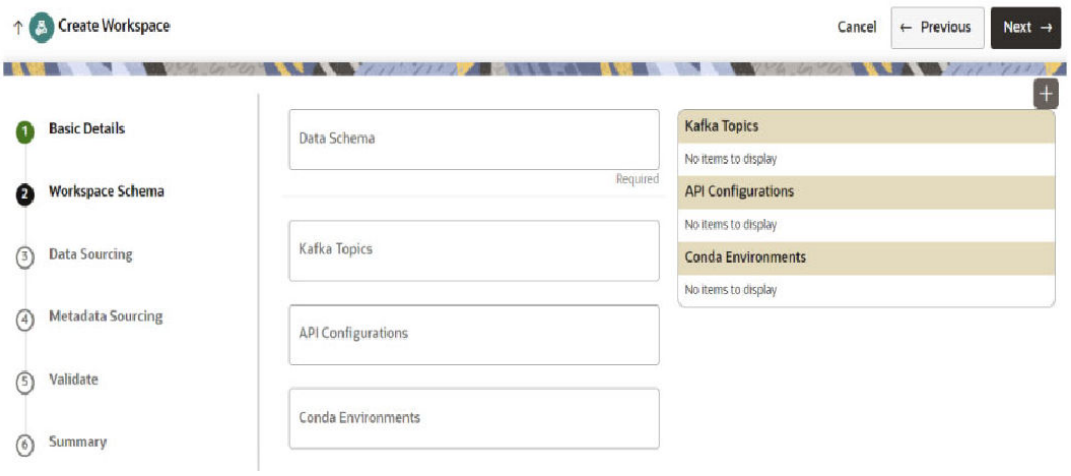
To create the workspace schema, follow these steps:

1. Select the **Data Schema** as **ASC BD Schema**.

> **Note:**
>
> Leave **Kafka Topics** and **API Configuration** fields as blank.

2. Select the following **Conda Environments**:
   a. **default_8.1.2.8.0**
   b. **ml4aml_8.1.2.8.0**
3. Click **Next**.

Figure 5-2 Workspace Schema



Data Sourcing

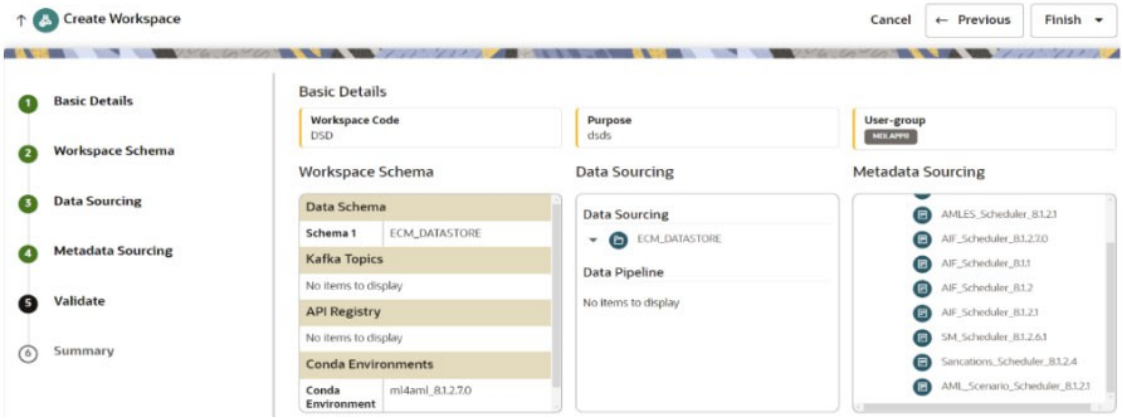
ASC uses valid BD schema as a target. Hence, all data is assumed to be available in the schema.

Metadata Sourcing

> **Note:**
>
> ASC use case does not require the metadata sourcing.

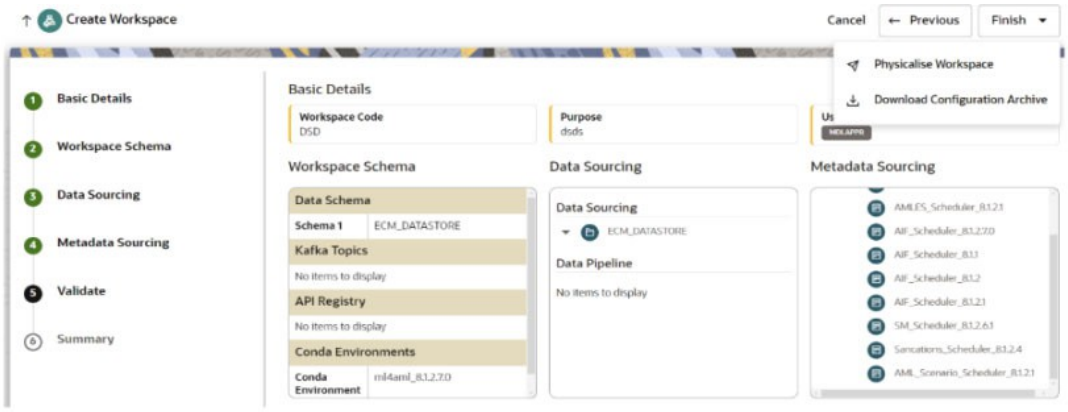
Validate Workspace

Figure 5-3 Validate Workspace



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

Figure 5-4 Physicalise Workspace



Summary
You can view summary of the created workspace.

Figure 5-5 Summary

5.1.2 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-1 Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh - w <workspace_wallet_alias>`

Table 5-1 (Cont.) Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
importNotebooksASC.sh	Yes	No	./importNotebooksASC.sh -w <workspace_code>

5.1.3 Using Scenario Conversion Utility for ASC

Scenario Conversion Utility (SCU) converts BD scenarios to Compliance Studio notebooks for the scenarios and respective thresholds.

> **Note:**
>
> The utility converts only BD AML scenarios to Compliance Studio notebooks based on the rule matcher algorithm.

Prerequisites

- Ensure that the OFS Compliance Studio v8.1.2.8.0 is installed and running.
- Make sure that all the Compliance Studio patches are applied.
- Ensure that the OFS BD v8.1.2.*.* is installed and running.
- Ensure that the following tables and sequence exist in the Compliance Studio Schema:
  - Tables:
    - ds_notebook
    - ds_paragraph
  - Sequence:
    - seq_paragraph

Calendar Notebook
The calendar notebook is used to set calendar information and processing batch date for execution of the scenario notebook.

Import `SCU_Set_Calendar.dsnb` notebook into the sandbox workspace and pointing to the non-prod BD atomic schema.

The following jars are available in the `<OFS_COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/Scenario-Conversion-Utility/db_tools/lib` directory.

- `dbtools.jar`
- `kddcore.jar`
- `log4j-api-<version>.jar`
- `log4j-core-<version>.jar`

> **Note:**
>
> This section is applicable only if you are using Scenario Conversion Utility and Automated Scenario Calibration use case.

5.1.3.1 Conversion Steps

To convert BD AML scenario, follow these steps:

- Import Workspace Metadata for SCU
- Accessing SCU Notebook
- Accessing Calendar Notebook
- Generating Threshold and Scenario Notebook
- Running the Scenario

Import Workspace Metadata for SCU

> **Note:**
>
> This section is applicable for both **fresh installation** and **upgrade**.

To import workspace metadata for SCU, follow these steps:

1. Create a public database link to Compliance Studio schema from ASC BD database server as SYS DBA using the following command.

```
set define off
/
DROP DATABASE LINK dl_studio
/
CREATE PUBLIC DATABASE LINK dl_studio
CONNECT TO <Compliance_Studio_atomic_user> IDENTIFIED BY
<Compliance_Studio_atomic_pwd>
USING ' (DESCRIPTION= (ADDRESS= (PROTOCOL = TCP) (HOST =
<replace_with_cs_db_ip> ) (PORT=1521) ) (CONNECT_DATA= (SERVICE_NAME =
<replace_with_cs_db_service_name> ) ) ) '
/
```

> **Note:**
>
> - Create DB link in the BD schema for connecting Compliance Studio.
> - Replace <compliance_studio_atomic_user>, <compliance_studio_atomic_pwd> with proper value for creating the required DB Link.
> - DB Link **should not** contain Domain Name of the DB and it should be just **DL_STUDIO** without additional character appended by DB.

2. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/Scenario-Conversion-Utility/bin` directory.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-2 Utilities for Workspace and Notebook

Utility	BD Workspace	Command
`importWorkspaceSQLSCU.sh`	`Yes`	`./importWorkspaceSQLSCU.sh -w <workspace_wallet_alias>`
`importNotebooksSCU.sh`	`Yes`	`./importNotebooksSCU.sh -w <workspace_code>`

Note:

After executing the `importWorkspaceSQLSCU.sh`, the following scripts are executed internally:

- `synonym.sql`
- `sequence.sql`
- `types.sql`
- `function.sql`
- `table.sql`
- `views.sql`
- `package_body_common.sql`
- `package_body.sql`

If any BD scenario xml files are modified, then ensure that the materialized views are refreshed using the following commands.

```
BEGIN DBMS_SNAPSHOT.REFRESH(
'"<BDSCHEMANAME>"."VW_SCNRO_BIND_MD"','C'); end;
/
BEGIN DBMS_SNAPSHOT.REFRESH(
'"<BDSCHEMANAME>"."VW_SCNRO_CHKPT_BIND_MD"','C'); end;
/
BEGIN DBMS_SNAPSHOT.REFRESH(
'"<BDSCHEMANAME>"."VW_SCNRO_CHKPT_MD"','C'); end;
/
BEGIN DBMS_SNAPSHOT.REFRESH(
'"<BDSCHEMANAME>"."VW_SCNRO_CONSTRAINT_MD"','C'); end;
/
BEGIN DBMS_SNAPSHOT.REFRESH(
'"<BDSCHEMANAME>"."VW_SCNRO_DATASET_JOB_MD"','C'); end;
/
BEGIN DBMS_SNAPSHOT.REFRESH(
'"<BDSCHEMANAME>"."VW_SCNRO_HIGHLIGHT_MD"','C'); end;
```

Replace the <BDSCHEMANAME> placeholder with the username/schema name of the target data source of the underlying workspace/sandbox. i.e., BD Atomic schema in case of a production workspace, target data source of ASC workspace.

4. If both ASC BD schema and Compliance Studio schema are part of same database server and you should not use the database link; in that case you should create synonyms with studio schema as a user name as follows:

```
CREATE OR REPLACE SYNONYM ds_notebook FOR
<STUDIO_SCHEMA_NAME>.ds_notebook
/
CREATE OR REPLACE SYNONYM ds_paragraph FOR
<STUDIO_SCHEMA_NAME>.ds_paragraph
/
```

```
CREATE OR REPLACE SYNONYM seq_paragraph FOR
<STUDIO_SCHEMA_NAME>.seq_paragraph
```

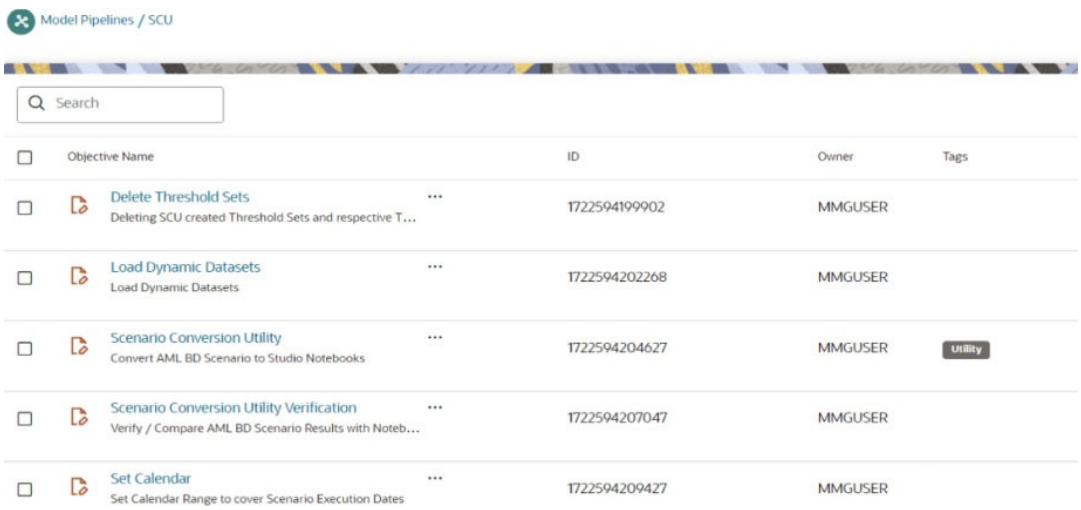
Accessing SCU Notebook

> **Note:**
>
> - This section is applicable for both **fresh installation** and **upgrade**.
> - Select **Show Empty Objectives** checkbox if SCU Objective is not visible in the Model Pipelines page.

To access the SCU notebook, follow these steps:

1. Navigate to the **Workspace Summary** page.
2. Select the Workspace.
3. On the **Modeling** menu, select **Pipelines**. The **Model Pipelines** window is displayed.
4. Click the **SCU** folder and you can see the following notebooks:
   - Delete Threshold Sets
   - Load Dynamic Datasets
   - Scenario Conversion Utility
   - Scenario Conversion Utility Verification
   - Set Calendar

Figure 5-6 SCU Notebooks



5. Click **Scenario Conversion Utility** to access the SCU notebook.

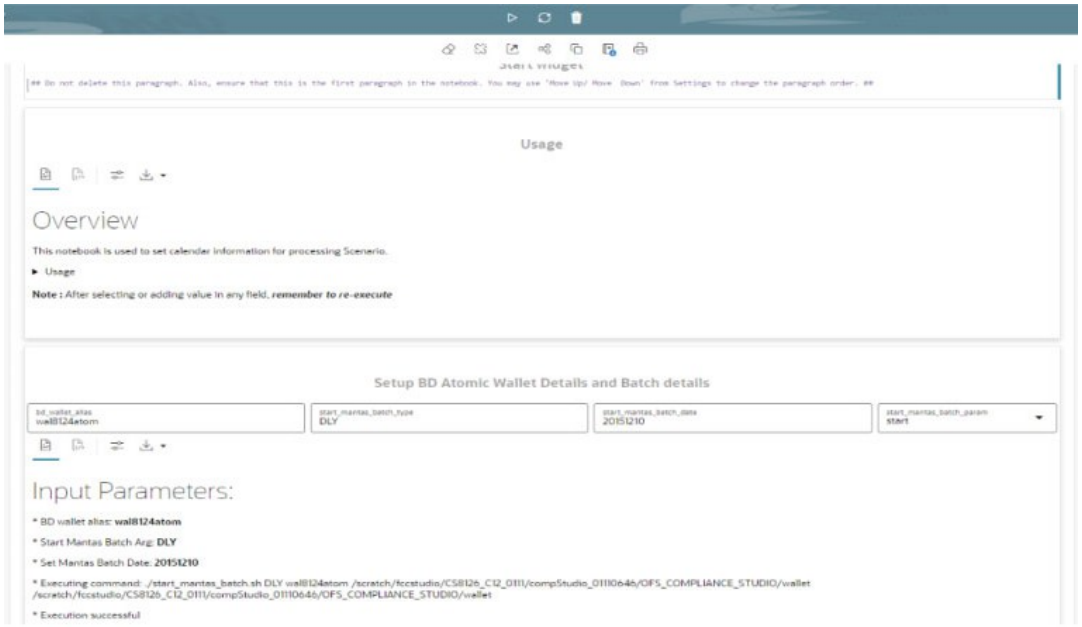
Accessing Calendar Notebook

To access the calender notebook, follow these steps:

1. Click **Launch** on the **Scenario Conversion Utility** workspace to launch workspace to display the **Dashboard** window with application configuration and model creation menu.

2. On the **Modeling** menu, click **Pipelines**.
3. Click **SCU** Objective Name.
   Generally, the notebooks are available where you imported.
4. Click **Set Calendar** notebook. The Pipeline canvas page is displayed.
5. From the **Python Runtime** drop-down list, select the **ml4aml_8.1.2.8.0**. The selected Python runtime parameter will be used during all the notebook executions.
6. Click the **Notebook** tab. The following page is displayed.

Figure 5-7 Calendar Notebook



In the **Setup BD Atomic Wallet Details and Batch details** paragraph,

7. Enter the Behavior Detection Atomic Schema's oracle wallet alias in the **bd_wallet_alias**. The wallet alias is available in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/wallet/tnsnames.ora` directory.
8. Enter the **start_mantas_batch_type**. By default, it is DLY.
   For example, DLY indicates the Daily Batch.
9. Enter the **start_mantas_batch_date** in the specified format YYYYMMDD. By default, enter the current date.
   For example, 20241010 for 10th Oct 2024.
10. Execute Run Paragraph to processing batch date for execution of the scenario notebook. After execution, you can verify the data are populating for the batch processing by executing the following queries:

```
select * from kdd_cal;
select * from kdd_prcsng_batch_control;
```

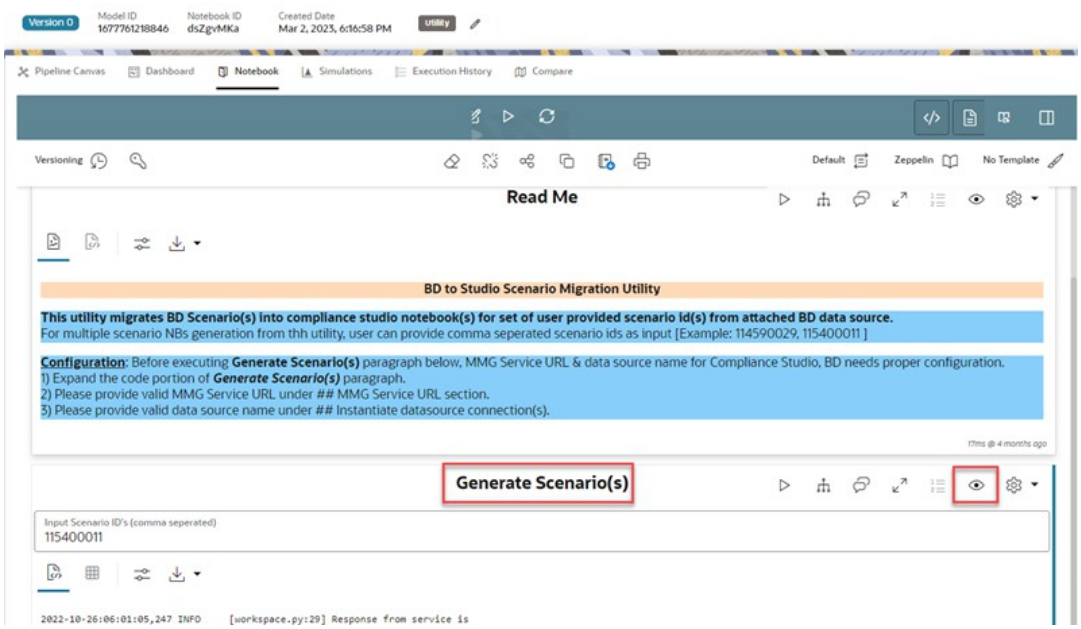
Generating Threshold and Scenario Notebook

> **Note:**
>
> - This section is applicable for both **fresh installation** and **upgrade**.
> - In case of upgrade scenario (CS 8.1.2.6.0/8.1.2.7.0 to CS 8.1.2.8.0), the user has to delete all the generated scenario notebooks which were generated on CS 8.1.2.6.0/8.1.2.7.0. in the 'AMLScenarios- Converted" objective. The scenario notebooks should be generated again in CS 8.1.2.8.0 by performing the below steps.

To generate a threshold and Scenario Notebook, follow these steps:

1. Navigate to the **Notebook** tab. The following page is displayed.

Figure 5-8 Generate Scenario(s)

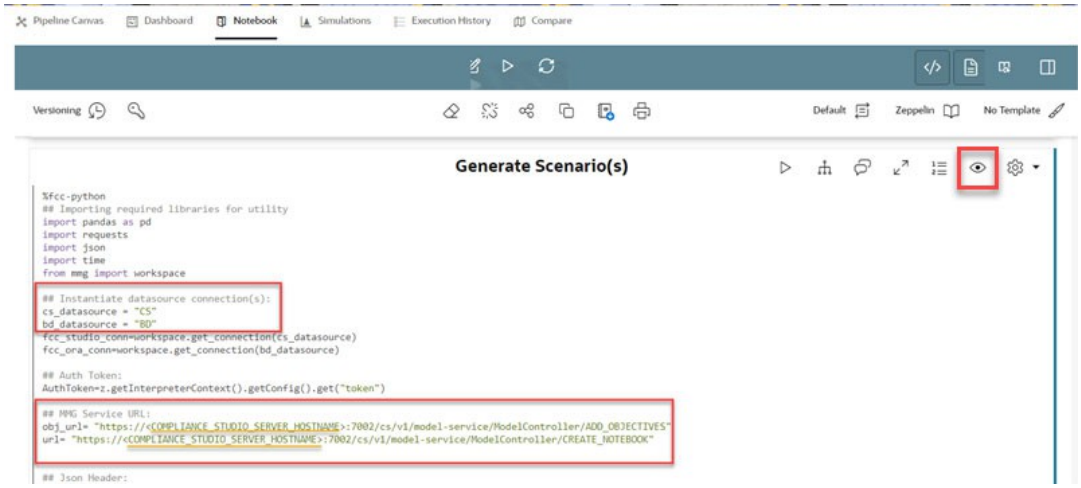


2. In the **Generate Scenario(s)** paragraph, click the **Visibility** icon and select the **Code** option.
3. Expand the Code in **Generate Scenario(s)** paragraph. Replace the Hostname for **obj_url** and **url** variables with the hostname of the Compliance Studio server.

> **Note:**
>
> If you are using custom ports, then replace **7002** with the port number available in the **BASE_URL** record in the **NEXTGEMEMF_CONFIG** table in the Studio schema.

4. Provide valid data source name for `cs_datasource` and `bd_datasource` variables under `## Instantiate datasource connection(s)` section in the **Generate Scenario(s)** paragraph.

Figure 5-9 Editing MMG Server URL and Data Source Name



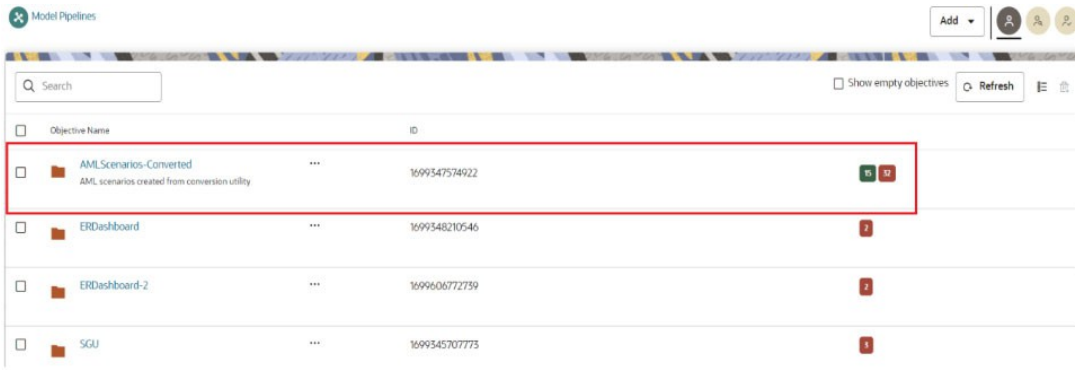
5. Enter the required Scenario ID(s) in the **Input Scenario ID's (comma-separated)** text box. You can enter multiple IDs with commas separated.
6. Click **Run Paragraph** to generate threshold and Scenario Notebook. Once it is executed successfully, you can view the success message and scenario notebook details in **Generate Scenario(s)** paragraph.

Figure 5-10 Successful Output Message

> **Note:**
>
> The folder structure for the generated scenario and threshold set notebooks is as follows.
>
> The objective will be created by the name AMLScenarios-Converted.
>
> - Inside the AMLScenarios-Converted objective, one or more objectives will be created by the name <Scenario ID>. This depends upon scenario ids provided in the Scenario Conversion Utility notebook.
> - Inside the <Scenario ID> objective, the threshold set notebook will be generated as "Threshold."
> - Inside the <Scenario ID> objective, one or more than one scenario notebooks will be generated based on the number of jobs for each scenario.

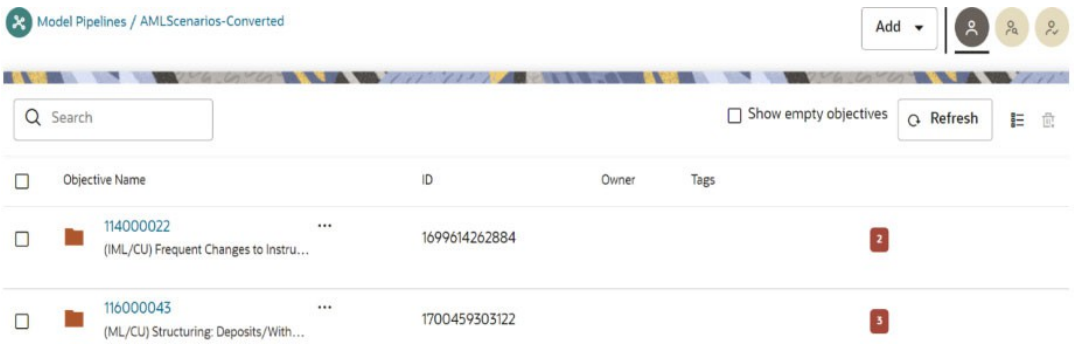
7. After successful execution, navigate to **Pipelines** and click **AMLScenarios-Converted** objective to display the list of scenarios created from the conversion utility.

Figure 5-11 AMLScenarios-Converted Folder



8. Select the **<Scenario ID>** objective from the **AML Scenarios-Converted** folder.

Figure 5-12 Scenario ID

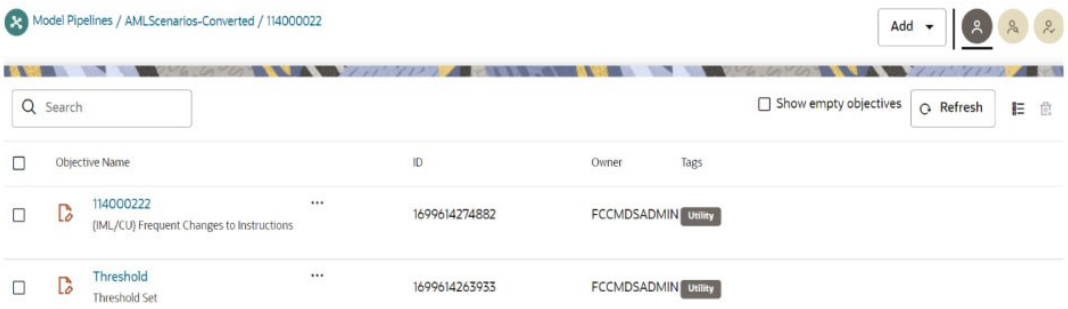


Use the threshold and Scenario Notebook ID generated during the execution to identify the Scenario Notebook folder.

> **Note:**
>
> For **n** number of jobs **n** number of Notebook IDs are created for scenario.

9. Click **<Scenario ID>** objective. The Pipeline canvas page is displayed.

Figure 5-13 Scenario ID Objective



10. On the right pane, select **default_8.1.2.8.0**, from the **Python Runtime** drop-down list. The selected Python runtime parameter will be used during all the notebook executions.

> **Note:**
>
> If generated scenario notebook is used with ASC feature, then **ml4aml_8.1.2.8.0** should be selected from the Python Runtime drop-down list.

11. Navigate to **Notebook** tab to run paragraph and generate the events

Running the Scenario

> **Note:**
>
> This section is applicable for both **fresh installation** and **upgrade**.

To run the scenario, follow these steps:

1. In Threshold Notebook, configure the following paragraphs:
   - Metadata
   - Base threshold set
   - Custom threshold set

   Follow the markdown provided under **(1) About and (5) Custom Threshold Builder – Instructions** paragraphs for instructions.

Figure 5-14 (1) About Paragraph

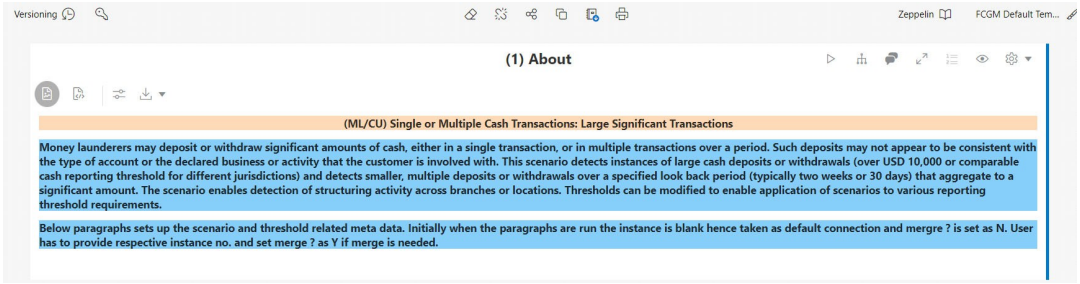
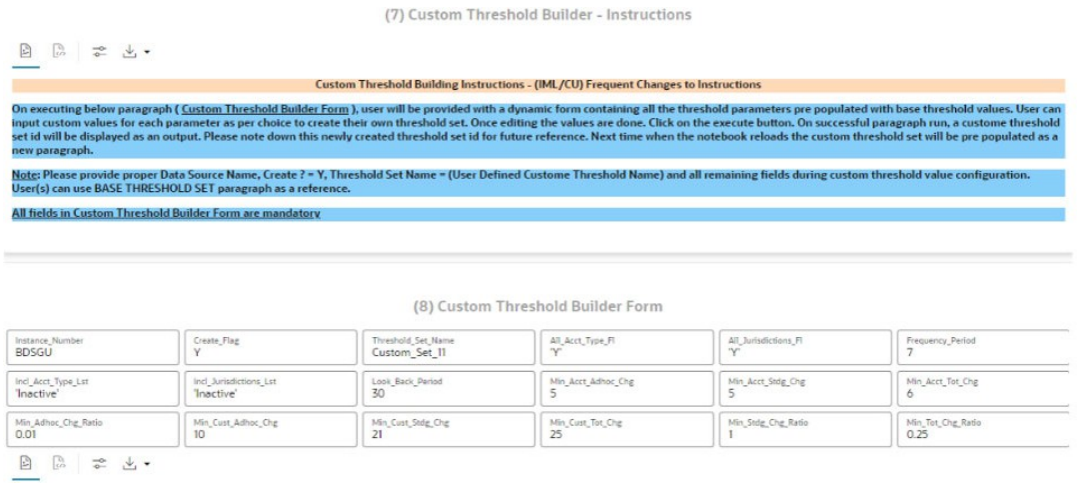


Figure 5-15 (5) Custom Threshold Builder – Instructions Paragraph



> **Note:**
>
> - Instance number is Data source of the BD atomic schema.
> - Threshold Notebook naming is created in the following format: **ML_<Scenario Catalogue Name>_<Focus>_TS**

2. Once the threshold configuration is completed, navigate to the respective scenario Notebook and click **Run Paragraphs** to generate events.
   Enter the custom value in the **Custom Threshold Set ID** field to run the scenario with a custom threshold set id.

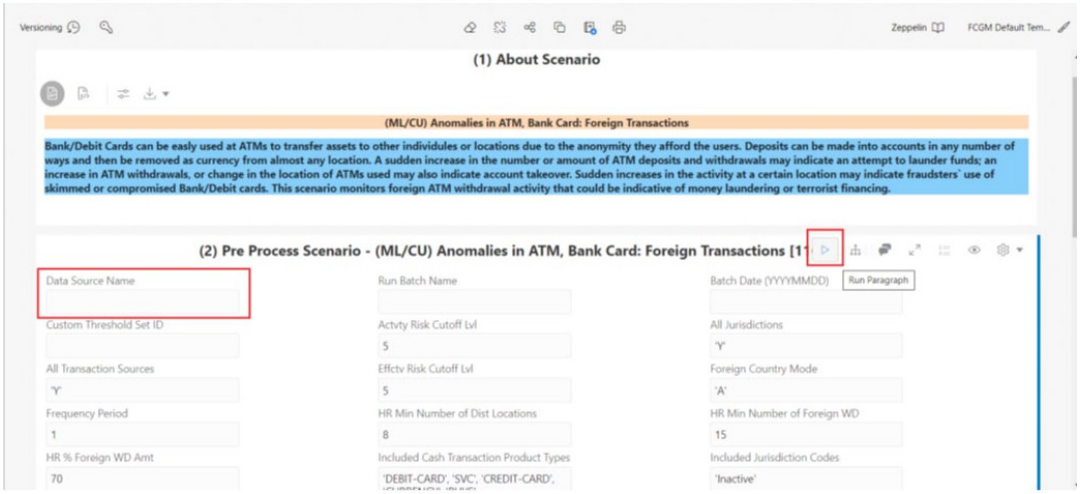
> **Note:**
>
> Scenario Notebook naming is created in the following format:
>
> **ML_<Scenario Catalogue Name>_<Focus>_<Job ID>**
>
> You can use the **Run ID** from the **Create Event** paragraph to verify the data for whom the events are generated with the event details created in the BD schema.

Using Custom Data Source

To use a custom data source (example: sandbox) for running the scenario(s), follow these steps:

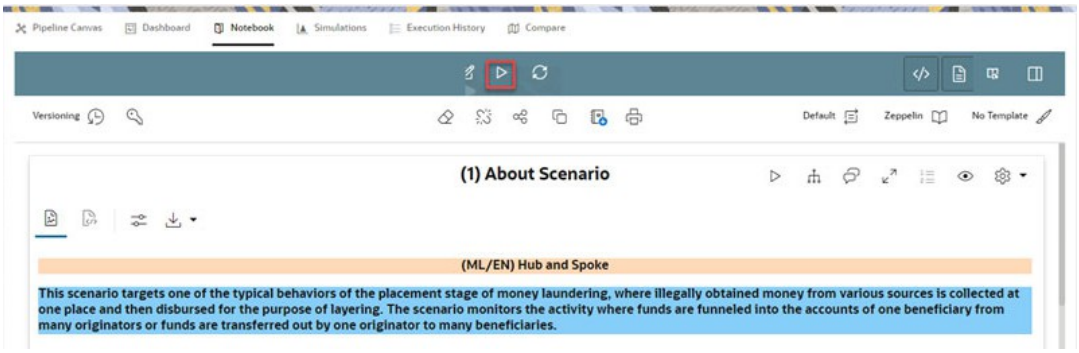
1. To create a data source, see the How to Create Data Store section.
2. Navigate to **Workspace summary** > **Managed Data Sources**.
3. Select the custom data source name and enter the same name in the **Data Source Name** text box of the**(2) - Pre Process Scenario -<Scenario notebook>** paragraph in the respective scenario notebooks.

Figure 5-16 Custom Data Source



4. Click **Run Paragraph** to execute the scenario using a custom data source.

Figure 5-17 Run Paragraphs Action



5. Once it is executed, click **Run Paragraphs** at the top of the Notebook to run the scenario notebook entirely rather than executing each paragraph in sequence.

> **Note:**
>
> By default, scenario notebooks use data sources as BD.

Supported Scenario

The following terms are used in the scenario:

- AC – Account
- CU – Customer
- ML – Money Laundering
- ML/AC – Money Laundering Account focus Scenario
- ML/CU – Money Laundering Customer focus Scenario

The following scenarios are supported:

- (ML/AC) Transactions in Round Amounts
- (ML/CU) Early Payoff or Paydown of a Credit Product
- (ML/AC) CIB: High Risk Geography Activity
- (ML/AC) Anticipatory Profile - Expected Activity
- (ML/AC) CIB: Significant Change from Previous Peak Activity
- (ML/AC) CIB: Significant Change from Previous Average Activity
- (ML/AC) CIB: Product Utilization Shift
- (ML/CU) Anomalies in ATM, Bank Card: Foreign Transactions
- (ML/CU) Anomalies in ATM, Bank Card: Excessive Withdrawals
- (ML/CU) Rapid Movement of Funds - All Activity
- (ML/AC) Rapid Movement of Funds - All Activity
- (ML/CU) Single or Multiple Cash Transactions: Large Significant Transactions
- (ML/AC) Deposits / Withdrawals in Same or Similar Amounts
- (ML/AC) Patterns of Funds Transfers Between Customers and External Entities
- (IML/CU) Frequent Changes to Instructions
- (IML/CU) High Risk Electronic Transfers
- (IML/EN) High Risk Electronic Transfers
- (ML/CU) High Risk Transactions: Focal High Risk Entity
- (ML/AC) High Risk Transactions: Focal High Risk Entity
- (ML/AC) High Risk Transactions: High Risk Counter Party
- (ML/HH) High Risk Transactions - High Risk Counter Party
- (ML/AC) High Risk Transactions: High Risk Geography
- (ML/CU) Single or Multiple Cash Transactions: Possible CTR
- (ML/HH) Single or Multiple Cash Transactions: Possible CTR
- (ML/AC) Anomalies in ATM, Bank Card: Excessive Withdrawals
- (ML/AC) Deposits/Withdrawals in Same or Similar Amounts
- (ML/EN) Hub and Spoke
- (ML/AC) Terrorist Financing
- (ML/EN) Terrorist Financing
- (ML/EN) Transactions in Round Amounts (MI)

- (ML/CU) Large Reportable Transactions
- (ML/EN) Patterns of Recurring Originators/Beneficiaries in Funds Transfers
- (ML/CU) Patterns of Funds Transfers Between Customers and External Entities
- (ML/AC) Patterns of Funds Transfers Between Internal Accounts and Customers
- (ML/CU) Hub and Spoke
- (ML/AC) CIB: Foreign Activity
- (ML/CB) CIB: Significant Change from Previous Peak Activity
- (ML/CB) CIB: Significant Change from Previous Average Activity
- (ML/CU) Routing of Funds Through Multiple Location
- (ML/CU) High Risk Transactions: High Risk Geography
- (ML/CU) Terrorist Financing
- (ML/AC) Escalation in Inactive Account
- (ML/CU) Structuring: Deposits/Withdrawals of Mixed Monetary Instruments
- (ML/AC) Anomalies in ATM, Bank Card: Foreign Transactions
- ML/CU) Structuring: Avoidance of Reporting Threshold

> **Note:**
>
> Other BD scenarios apart from the above mentioned list are under the verification process.

5.1.4 Comparison of Events between BD and Conversion Utility

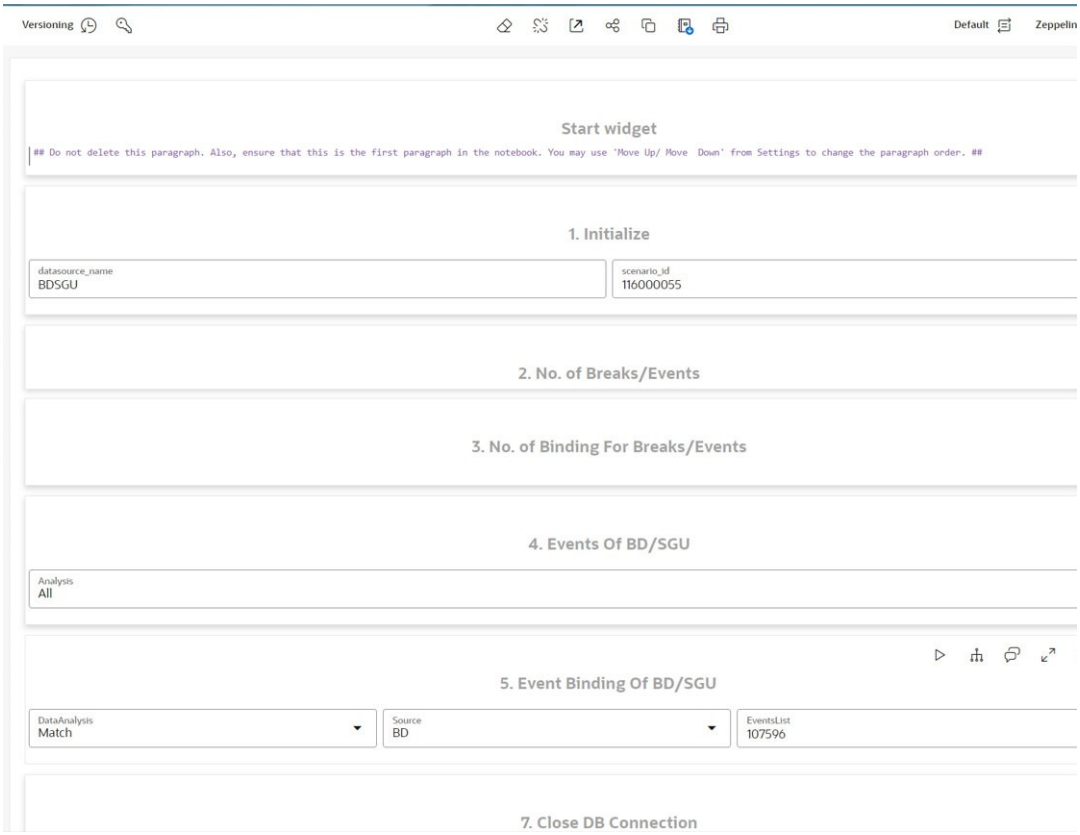
Scenario Conversion Utility allows you to compare scenario execution results from BD Engine with the Scenario Conversion utility notebook of Compliance Studio.

Accessing the Verification Notebook
To access the verification notebook, follow these steps:

1. Click **Launch** on the S**cenario Conversion Utility** workspace to launch workspace to display the **Dashboard** window with application configuration and model creation menu.
2. On **Modeling** menu, select **Pipelines**.
3. Click **SCU** folder. The following notebooks are displayed:
   - Delete Threshold Sets
   - Load Dynamic Datasets
   - Scenario Conversion Utility
   - Scenario Conversion Utility
   - Verification
   - Set Calendar

   Generally, the notebooks are available where you imported.
4. Click **Scenario Conversion Utility Verification** and click the **Notebook** tab. The following page is displayed.

Figure 5-18 Verification Notebook



5. Enter the **Datasource Name** and **Scenario ID** in the **Initialize** paragraph.
6. Run the **Initialize** paragraph. After successful execution, the following paragraphs are displayed.
   a. **No. of Breaks/Events**: Displays the number of breaks and events generated by BD and SGU for the scenario entered in the **Initialize** paragraph.
   b. **No. of Binding for Breaks/Events**: Displays the number of break bindings from BD and event bindings from SGU for the scenario entered in the **Initialize** paragraph.
7. In the **Events of BD/SGU** paragraph, select the required option from the **Analysis** drop-down list. The available options are:
   - **All**: Displays all the events from the BD and SGU
   - **BD Side Available**: Displays all events from the BD that are not available in the SGU
   - **SGU Side Available**: Displays all events from the SGU that are not available in the BD
   - **Mismatch**: Displays if any event attributes do not match between BD and SGU
   - **Match**: Displays all events with similar attributes matching between BD and SGU
8. Click **Run Paragraph** to view the event analysis.
9. In the **Event Binding of BD/SGU** paragraph, select the required option from the **Data Analysis** drop-down list. The available options are:
   - **All**: Displays all the event binding from the BD and SGU
   - **Mismatch**: Displays if any event binding attributes do not match between BD and SGU

- **Match**: Displays all event binding with similar attributes matching between BD and SGU

10. Click **Run Paragraph** to view data for the event binding.

> **Note:**
>
> If you need to close connection from the database then **Run Paragraph** in the Close DB Connection.

5.1.5 Using Delete Threshold Sets Notebook

This notebook provides functionality for deleting threshold sets created for SCU, especially when migrating threshold sets from the lower to the higher version of the BD environment by deleting unwanted threshold sets from the systems.

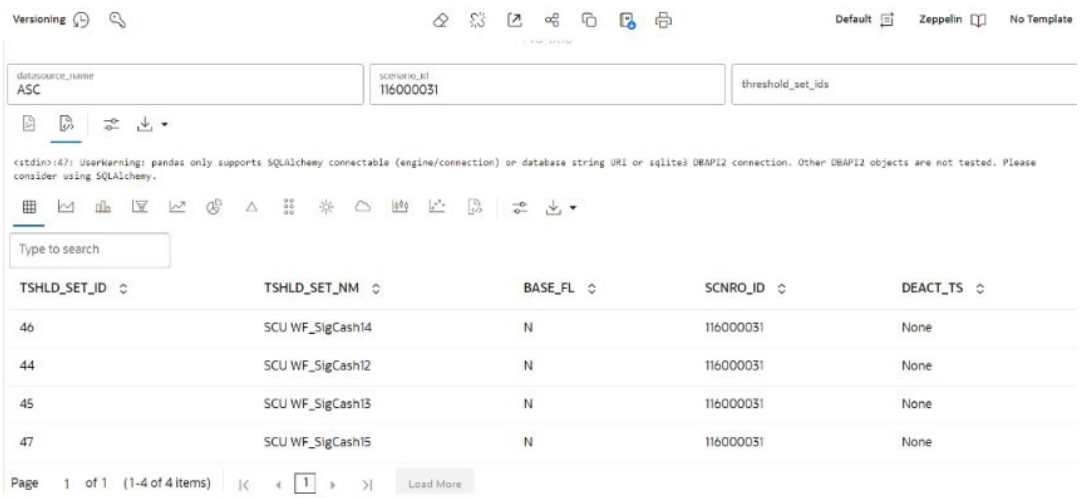
To access the delete threshold sets notebook, follow these steps:

1. Navigate to the **Workspace Summary** page.
2. Select the Workspace.
3. On **Modeling** menu, select **Pipelines**. The **Model Pipelines** window is displayed.
4. Navigate to the **SCU** folder and open the **Delete Threshold Sets** notebook in Pipeline Designer.
5. From the **Python Runtime** drop-down list, select the **default_8.1.2.8.0**. The selected Python runtime parameter will be used during all the notebook executions.
6. Click the **Notebook** tab. The notebook contains a couple of markdown paragraphs and a single Python paragraph.

View Threshold Sets for SCU
To view the threshold sets created for SCU, follow these steps:

1. Open the **Delete Threshold Sets** notebook.
2. Navigate to the **Last/Python** paragraph.

Figure 5-19 Python Paragraph



3. Enter the valid **datasource_name**.
4. Enter the **scenario_id**. You must enter only one scenario_id.
5. Do not enter any value (leave it blank) in the **threshold_set_ids** field.
6. Execute the paragraph to list all threshold sets created for SCU based on the provided scenario.

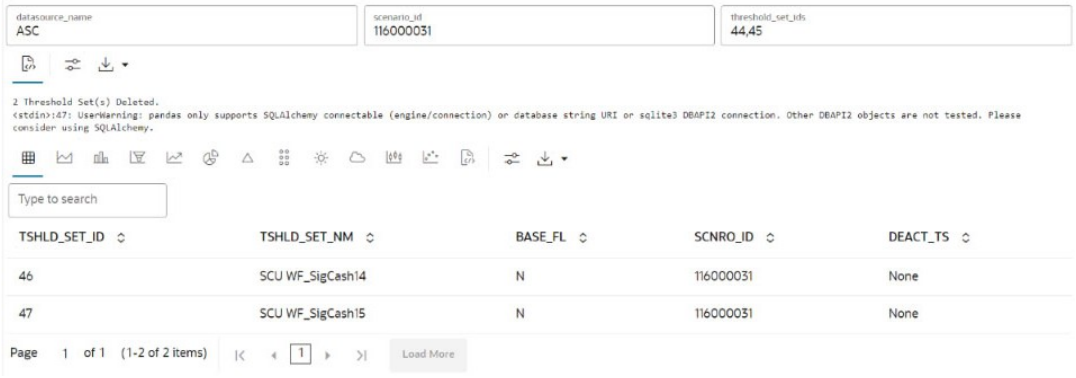
Delete Threshold Sets for SCU

> **Note:**
>
> The running below paragraph will permanently delete the specified threshold sets. Ensure the other users do not need this threshold set before completing this action.

To delete the threshold sets created for SCU, follow these steps:

1. Open the **Delete Threshold Sets** notebook.
2. Navigate to the **Last/Python** paragraph.

Figure 5-20 Python Paragraph



3. Enter the valid **datasource_name**.
4. Enter the **scenario_id**. You must enter only one scenario id.
5. Enter the **threshold_set_ids** which needs to be deleted.
   You can enter multiple IDs with commas separated. For example: 44, 45.
6. Execute the paragraph to delete specified threshold id(s).
   By default, the remaining threshold sets will be available in the paragraph after successful deletion.

5.1.6 Using Dynamic Datasets with AML Scenario Conversion

This notebook provides functionality for loading dynamic datasets while AML scenario conversion using SCU.

To access the Load Dynamic Datasets notebook, follow these steps:

1. Navigate to the **Workspace Summary** page.
2. Select the Workspace.

3. On the **Modeling** menu, click **Pipelines** to display the Model Pipelines page.
4. Navigate to the **SCU** folder and open the **Load Dynamic Datasets** notebook in Pipeline Designer.
5. From the **Python Runtime** drop-down list, select the **default_8.1.2.8.0**. The selected Python runtime parameter will be used during all the notebook executions.
6. Click the **Notebook** tab. The notebook contains a couple of markdown paragraphs and a single Python paragraph.

Dynamic Dataset Text File
If OOB datasets performance are inefficient, then dataset tuning is required. Tuned datasets using DB hints can be configured in the Compliance Studio through **Dynamic Datasets**.

To copy dynamic dataset text file into the Compliance Studio, follow these steps:

1. Tune dataset query in the DB schema using SQL hints.
2. Once the dataset query is tuned, copy the tuned SQL query in a text file and name the text file as `dataset<DATASET_CODE>.txt`.
   - Where `<DATASET_CODE>` is the original DATASET code used in the AML scenario. For example, `dataset114012498.txt`.
3. Copy the newly created dynamic dataset file into the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/Scenario-Conversion- Utility/DynamicDatasets` directory.
4. In the Compliance Studio Unix server, convert text file using the following command.

   ```
   Execute dos2unix dataset<DATASET_CODE>.txt
   ```

5. Delete the sample dataset file provided (`dataset9999999.txt`) as a reference template in the same directory.

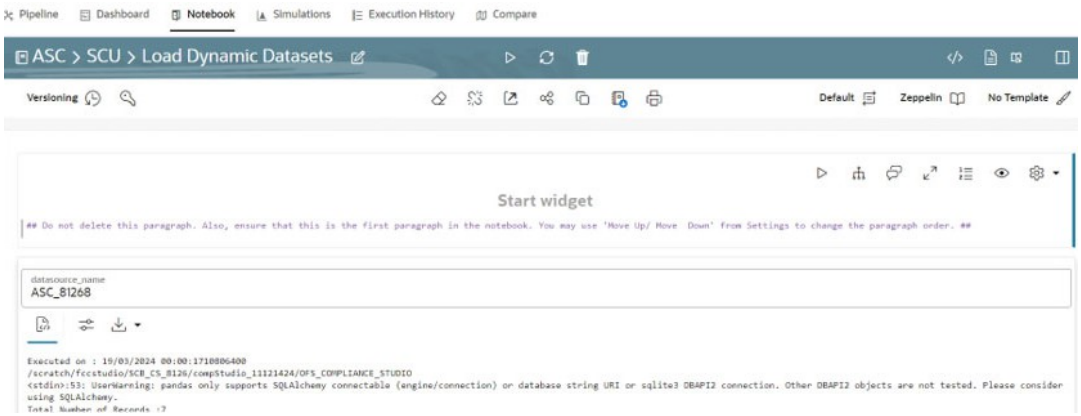
> **Note:**
>
> The user should provide correct queries in the dataset text files for a successful scenario execution.

Executing Dynamic Dataset Notebook
To execute the Load Dynamic Dataset notebook, follow these steps:

1. Click the **Launch** icon on the workspace which contains **Scenario Conversion Utility**.
2. On the **Modeling** menu, click **Pipelines**.
3. Click the **SCU** folder. The following notebooks are displayed:
   - Delete Threshold Sets
   - Load Dynamic Datasets
   - Scenario Conversion Utility
   - Scenario Conversion Utility Verification
   - Set Calendar

   Generally, the notebooks are available where you imported.
4. Click the **Load Dynamic Dataset** notebook and click the **Notebook** tab. The following page is displayed.

Figure 5-21 Dynamic Dataset Notebook



5. Enter the **Datasource Name**.
6. Run the paragraph.
   After successful execution, the dynamic dataset in the file system will be loaded into the **SCU_DYNAMIC_DATASET** table. The notebook can load all dynamic dataset files in the file system at a time.

   Once dynamic datasets are loaded into the table, the User can run the scenario conversion notebook, which converts the AML scenario into the Compliance Studio notebook by considering the Dynamic Dataset is present in the table.

   Post Scenario Conversion

   If Dynamic datasets are re-loaded into the table, Users need to re-run the scenario conversion steps by deleting the scenario notebook folder manually.

5.1.7 Advanced Concepts for ASC

For more feature about ASC, see Advanced Feature for ASC Use Case in the Appendix.

5.2 Behavioral Model

This section explains about Behavioral Model use case.

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
4. Create a new data store for the sandbox schema
5. Register Conda Environment in BD Production Workspace

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

To register Conda Environment in BD Production Workspace, see How to Register Conda Environment in BD Production Workspace section.

5.2.1 Creating Sandbox Workspace

To create the sandbox workspace, see How to Create Sandbox Workspace section.

5.2.2 Populating Sandbox Workspace

To populate the sandbox workspace, see How to Populate the Sandbox Workspace section.

5.2.3 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-3 Utilities for Workspace and Notebook

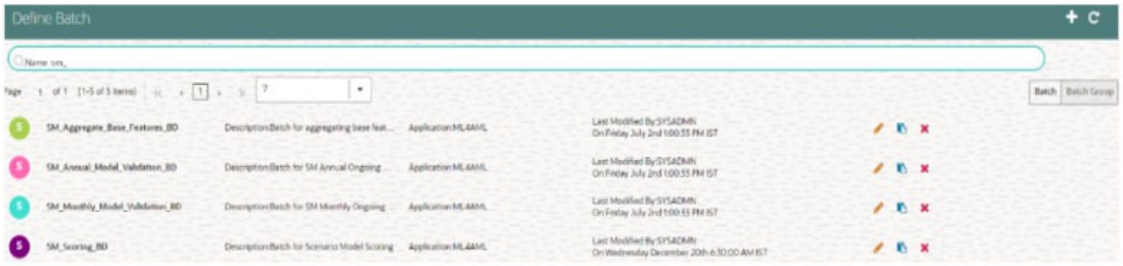
Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh -w <workspace_wallet_alias>`
`importNotebooksSM.sh`	Yes	Yes	`./importNotebookSM.sh -w <workspace_code>`

5.2.4 Batch Framework for Behavioral Model

The following batches are available in the out-of-the-box for the scenario model framework:

- **Behavioral Model Aggregate Base Features**
- **Behavioral Model Scoring**
- **Behavioral Model Annual Model Validation**
- **Behavioral Model Monthly Model Validation**
- **Behavioral Model SAR Extraction**

Figure 5-22 Define Batch for Scenario Model



5.2.4.1 Behavioral Model Aggregate Base Features

- This pre-seeded batch will be available in all the workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the **Sandbox** workspace.

- This batch creates base features for scenario model training in the sandbox workspace.

Batch and Task Parameters
The batch contains a single task named **Aggregate_Base_Features**.

Figure 5-23 Define Task for Aggregate_Base_Features

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_date**: 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 5-24 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 5-25 Edit Task for Aggregate Base Features

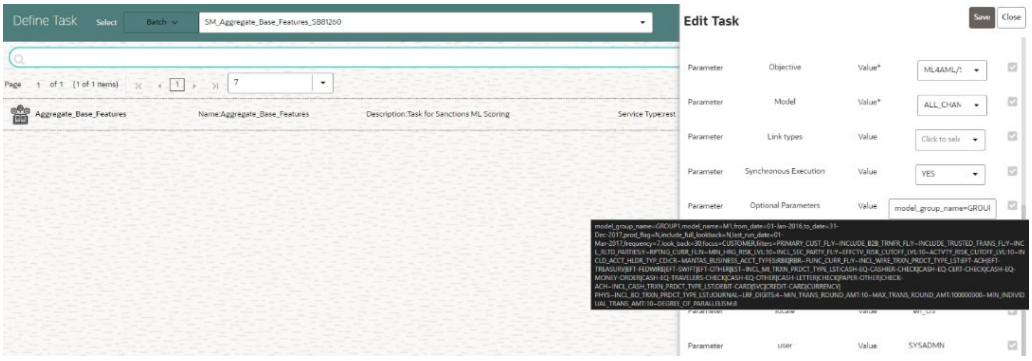


Table 5-4 Task Parameters for Scenario Model Aggregate Base Features

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.

Table 5-4 (Cont.) Task Parameters for Scenario Model Aggregate Base Features

Parameter	Description
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**.
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.

Table 5-4 (Cont.) Task Parameters for Scenario Model Aggregate Base Features

Parameter	Description
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.
STRUCTURED_CASH_LIMIT_MIN	Lower limit used to be considered by Financial Institutions
STRUCTURED_CASH_LIMIT_MAX	Reporting limit used to be considered by Financial Institutions
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~I
NCL_RLTD_PARTIES:Y~RPTNG_CURR_FL:N~MIN_HRG_RISK_LVL:10~INCL_SEC_PARTY_FL:Y~E
FFCTV_RISK_CUTOFF_LVL:10~ACTVTY_RISK_CUTOFF_LVL:10~INCLD_ACCT_HLDR_TYP_CD:C R~MANTAS_BUSINESS_ACCT_TYPES:RBK|
RBR~FUNC_CURR_FL:Y~INCL_WIRE_TRXN_PRDCT_ TYPE_LST:EFT-ACH|EFT-TREASURY|EFT-FEDWIRE|EFT-SWIFT|EFTOTHER|
EST~INCL_MI_TRXN_PRDCT_TYPE_LST:CASH-EQ-CASHIER-CHECK|CASH-EQ-CERTCHECK| CASH-EQ-MONEY-ORDER|CASH-EQ-TRAVELERS-CHECK|CASH-EQ-OTHER|CASHLETTER| CHECK|PAPER-OTHER|CHECK-
ACH~INCL_CASH_TRXN_PRDCT_TYPE_LST:DEBITCARD| SVC|CREDITCARD|
CURRENCY|PHYS~INCL_BO_TRXN_PRDCT_TYPE_LST:JOURNAL~LRF_DIGITS:4~MIN_TRANS_ROUND_AMT:10~MAX_TRANS_ROUND_AMT:100000000~MIN_INDIVIDUAL_TRANS_A
MT:10~STRUCTURED_CASH_LIMIT_MIN:10~STRUCTURED_CASH_LIMIT_MAX:1000~DEGREE_ OF_PARALLELISM:8

- **Edit** Task Parameters and **Save**.

5.2.4.2 Behavioral Model Scoring

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the **Production** workspace

Batch and Task Parameters
The batch contains the following tasks:

- **Task 1: Aggregate_Scoring_Base_Features**
- **Task 2: ML_Scoring**
- **Task 3: Event_Processing**

Figure 5-26 Define Task for SM Scoring

Task 1: Aggregate_Base_Features, Task Parameters

- Objective folder for this task:

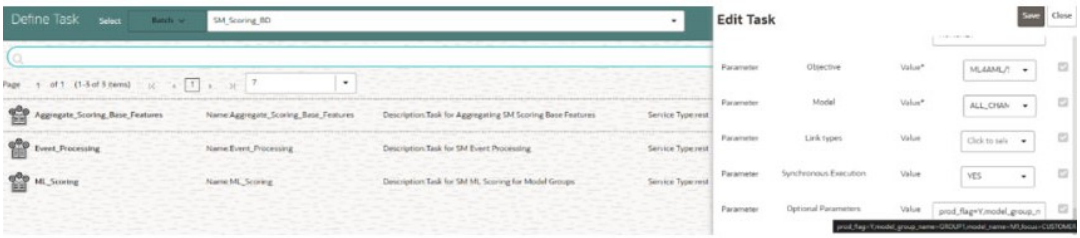
```
Home / Modeling / Pipelines / ML4AML / Scenario Model / Batch / Base
Features
```

> **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 5-27 Edit Task for SM Scoring



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 5-28 Edit Task Parameter for ML Scoring

> **Note:**
>
> Once the batch execution is successful, the results are available in the **SM_EVENT_SCORE** and **SM_EVENT_SCORE_DETAILS** tables. For more information on these table structure, see the OFS Compliance Studio Data Model Reference Guide.

Task 3: Event_Processing Task Parameters

- Objective folder for this task:

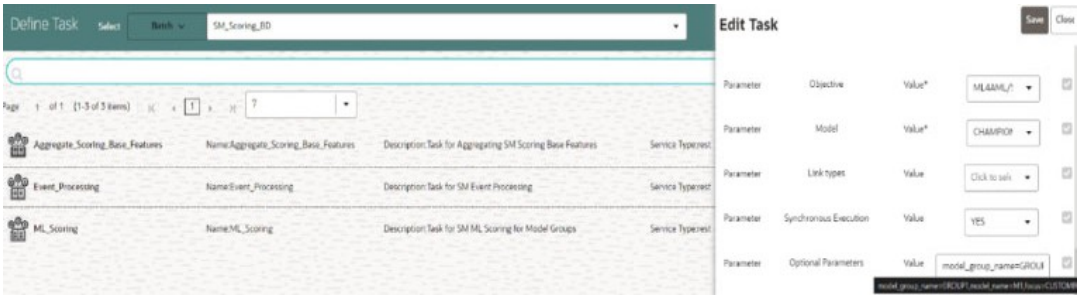
```
Home / Modeling / Pipelines / ML4AML / Scenario Model / Batch / Event
Processing
```

> **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 Task Parameters and **Save**

Figure 5-29 Edit Task Parameter for Event Processing



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.

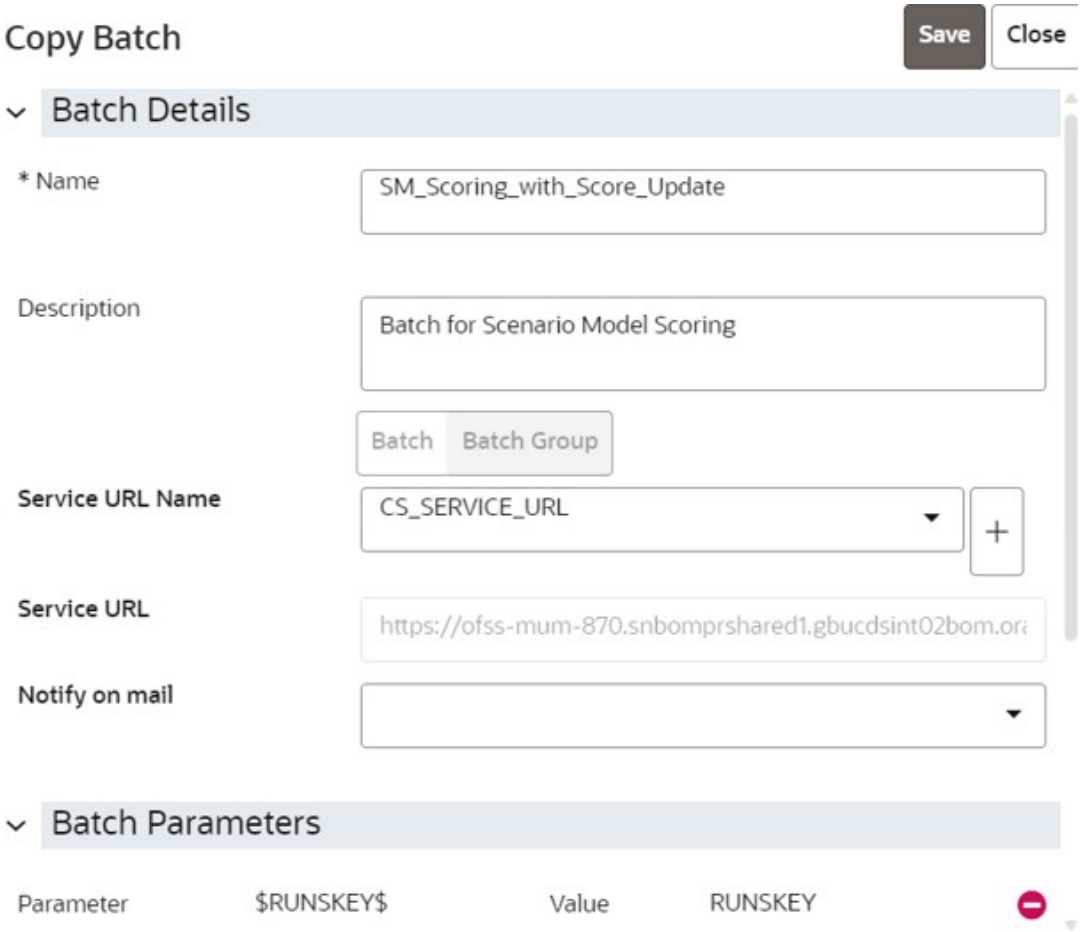
> **Note:**
>
> **Prerequisites**: See the **Score Update Notebook for Scenario Model** section in theOFS Compliance Studio Use Case Guide.

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:

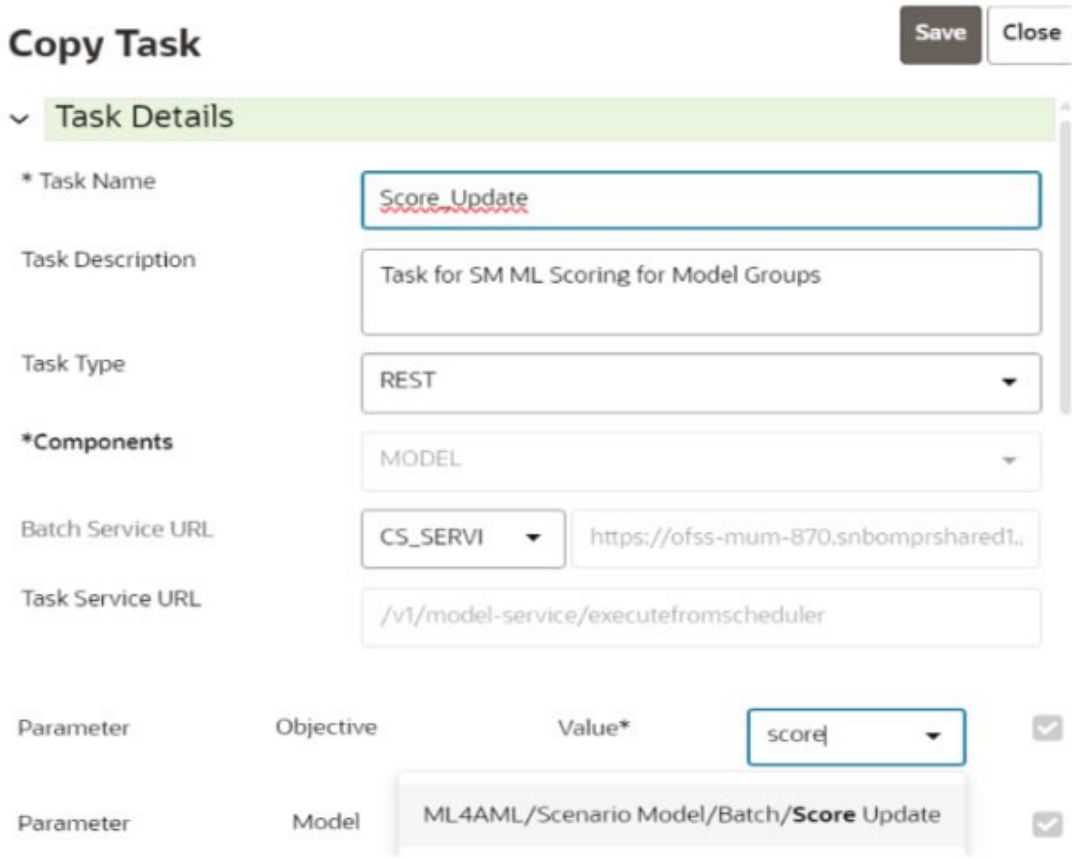
1. On the **Orchestration** mega menu, click **Define Batch**.
2. Search SM_Scoring Batch, and clone the batch using the **Copy** icon. The Copy Batch page is displayed.

Figure 5-30 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 **Copy** icon. The Copy Task page is displayed.

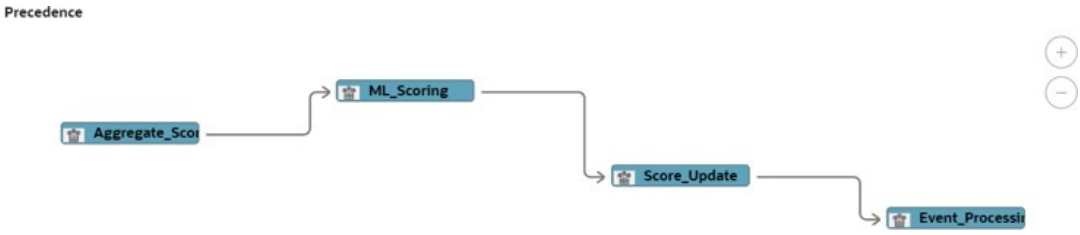
Figure 5-31 Copy Task



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 **Menu** 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 5-32 Precedence Mapping

Figure 5-33 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 5-34 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
```

> **Note:**
>
> The code in the new notebook will update the scores directly into the production table (SM_EVENT_SCORE_DETAILS). For more information on the table structure, see the OFS Compliance Studio Data Model Reference Guide.

5.2.4.3 Behavioral Model Annual Model Validation

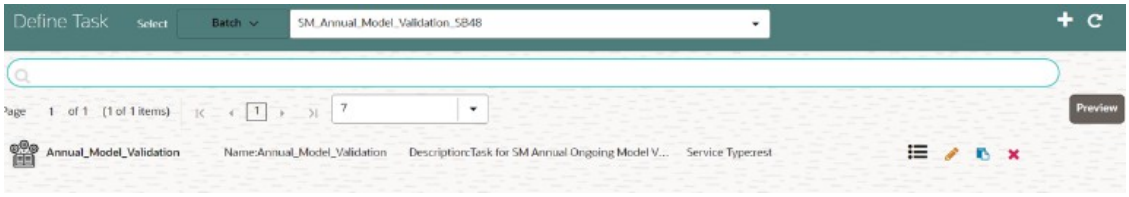
- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the **Production** workspace.

- This batch shows ongoing model performance annually.

Batch and Task Parameters
The batch contains a single task named Annual_Model_Validation.

Figure 5-35 Annual Model Validation for SM



Task: Annual_Model_Validation, Task Parameters

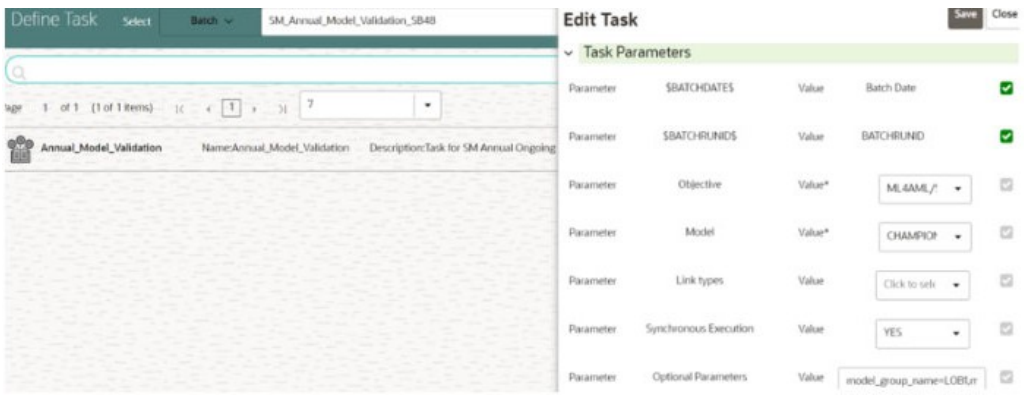
- Objective folder for this task:

```
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 5-36 Edit Task for Annual Model Validation



> **Note:**
>
> The Annual Model Validation batch shows output metrics in the notebook only and it will not store in any of the data tables.

5.2.4.4 Behavioral Model Monthly Model Validation

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the **Production** workspace.

- This batch shows ongoing model drift and data quality with respect to new data every month (monthly).

Batch and Task Parameters
The batch contains a single task named Monthly_Model_Validation.

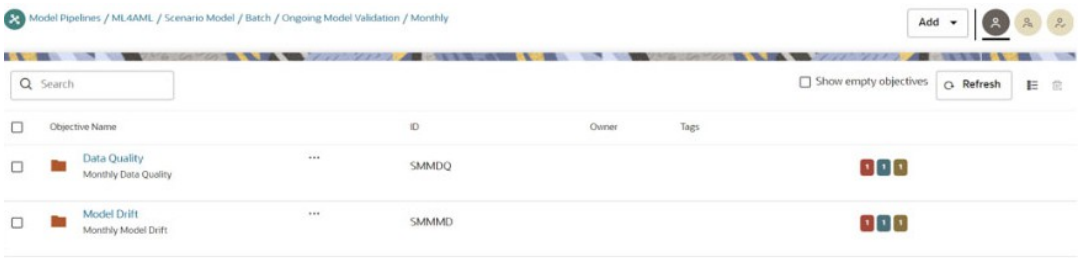
- Objective folder for Data Quality:

```
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 5-37 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**.

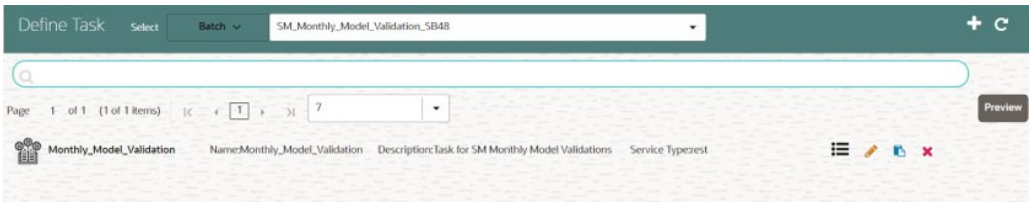
- **model_name**: Name of the Model for which the model has been trained. Example: **RMF**.
- **focus**: Name of the entity type or segment. Example: **CUSTOMER**.
- **model_id**: User passes parameter as Deployed to use the deployed model. Example: **Deployed**.
- **FEATURE_INCLUDE**: List of features 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_Str ap_Samples=5,Standard_Deviation_Band_Sigma=2,look_back_months=5,FEATURE_INCLUDE=None,FEATURE_EXCLUDE=None

Figure 5-38 Define Task for Monthly Model Validation



> **Note:**
>
> The Monthly Model Validation batch shows output metrics in the notebook only and it will not store in any of the data tables.

5.2.4.5 Obtain the SAR Information

This section provides information about how to Obtain the SAR Information.

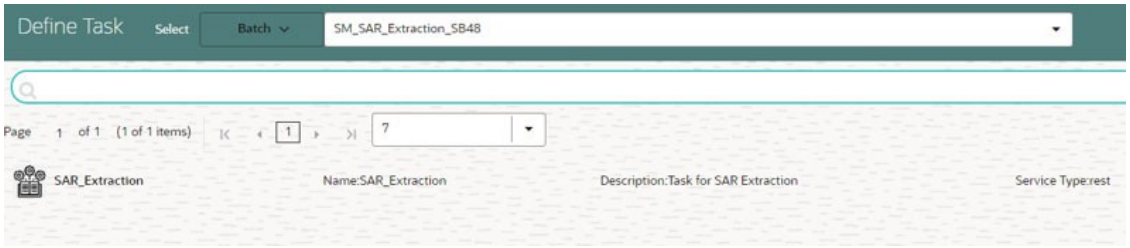
Populate Investigated Entity Details
SM_SAR_Extraction batch is available in the out-of-the-box for the Scenario Model framework. This is a pre-seeded batch and will be available in all the workspaces.

This batch loads SAR Information to AIF_INVESTIGATED_ENTITY table.

Batch and Task Parameters

The batch contains a single task named SAR_Extraction.

Figure 5-39 Define Task for SAR_Extraction



Task: SAR_Extraction, Task Parameters

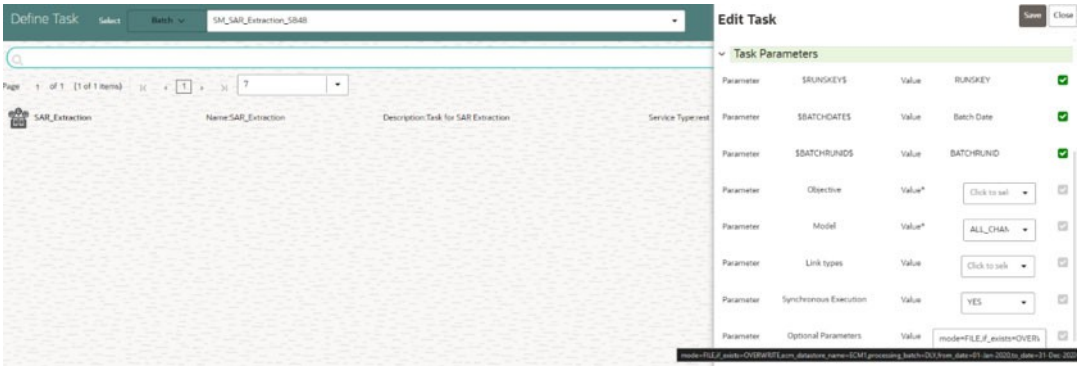
- Objective folder for this task:

  ```
  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 5-40 Edit Task for SAR_Extraction



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 5-41 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 (Nonsuspicious)**.
- **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.

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

5.2.5 Execute Batch

To execute the batch, see How to Execute Batch section.

5.2.6 Monitor Batch

To monitor the batch, see How to Monitor Batch section.

5.3 Sanctions Event Scoring

This section explains about Sanctions Event Scoring use case.

Prerequisites for Creating Production Workspace
Before creating the production workspace, user should follow these steps:

1. The target schema used for production workspace should be a valid Sanction Atomic Schema
2. Create the Tablespace
3. Assign grants to Sanction Atomic Schema
4. Create a new data store for Sanction Atomic Schema
   For more information on creating tablespace and assigning grants to Sanction Atomic Schema, see the OFS Compliance Studio Installation Guide.

   To create the data store, see How to Create Data Store section.

5.3.1 Creating Production Workspace

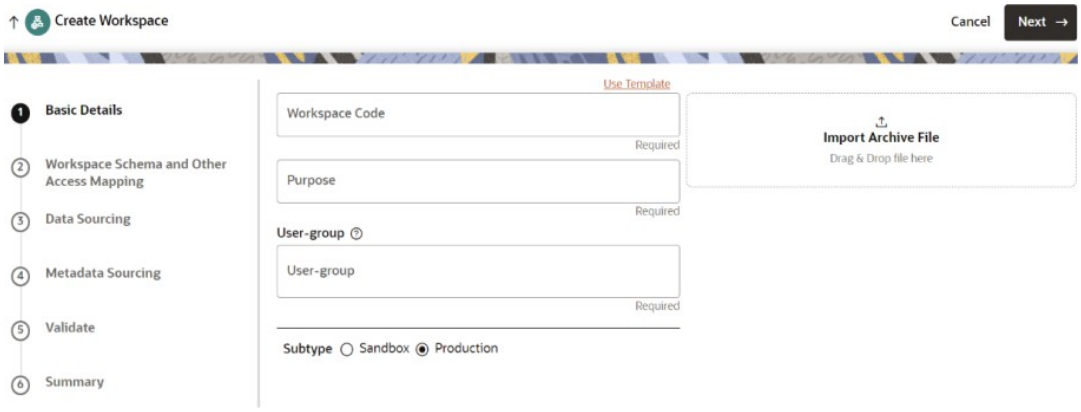
On the Workspace Summary page, click Add Workspace. The Create Workspace window is displayed with the following process:

1. Basic Details
2. Workspace Schema
3. Data Sourcing
4. Metadata Sourcing
5. Validate
6. Summary

Basic Details
To create a basic details of the production workspace, follow these steps:

1. Enter the **Workspace Code** and **Purpose** of the workspace.
2. From the drop-down list, select the **User-group**.
3. Select the subtype as **Production**.
4. Click **Next**.

Figure 5-42 Basic Details



Workspace Schema

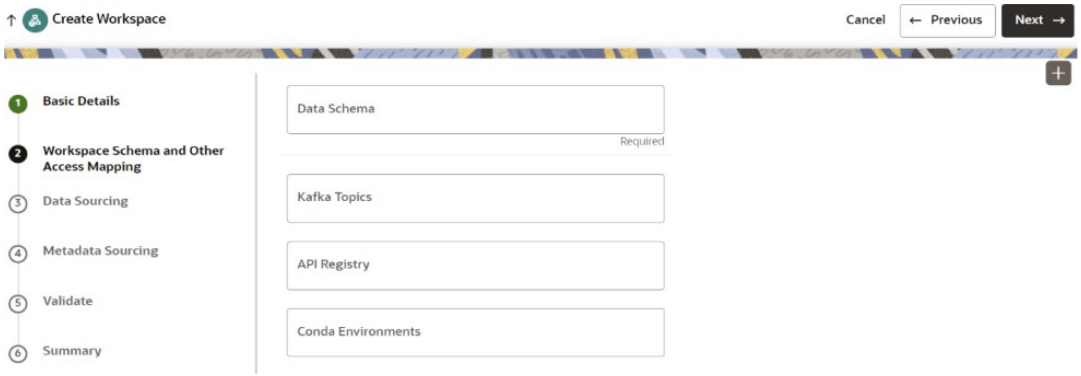
To create the workspace schema, follow these steps:

1. Select the **Data Schema** as **Sanction Atomic Schema**.

> **Note:**
>
> Leave the **Kafka Topics** and **API Configurations** fields as blank.

2. Select the following **Conda Environments**:
   a. **default_8.1.2.8.0**
   b. **ml4aml_8.1.2.8.0**
3. Click **Next**.

Figure 5-43 Workspace Schema



Data Sourcing

Data sourcing is not required as the production workspace is attached to the Sanction Atomic Schema. Click **Next** to navigate to the Metadata Sourcing tab.

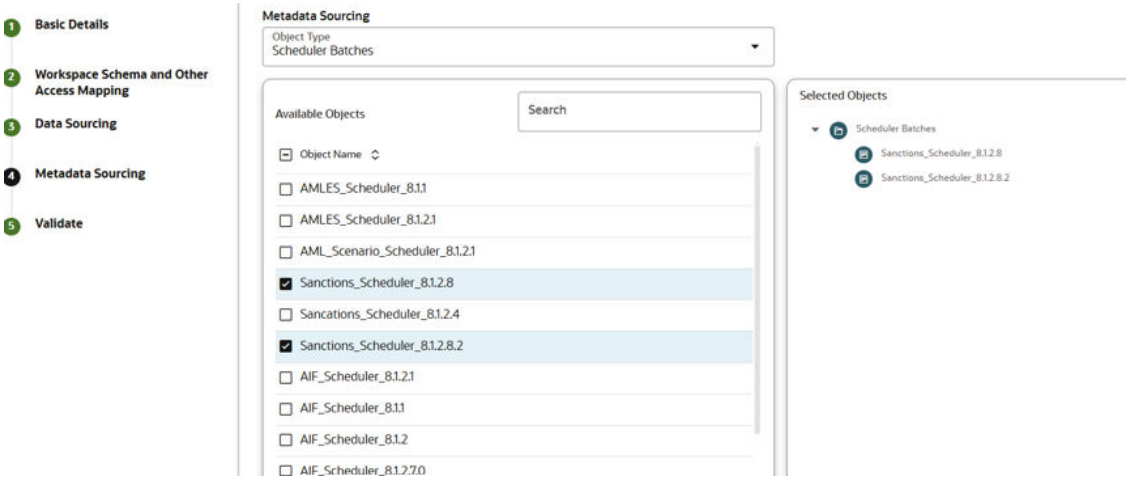
Metadata Sourcing

To select available objects from the Metadata Sourcing, follow these steps:

1. From the **Object Type** drop-down list, select **Scheduler Batches**.
2. In the **Available Objects**, select **Sanctions_Scheduler_8.1.2.8** and **Sanctions_Scheduler_8.1.2.8.2**.

3. Click **Next**.

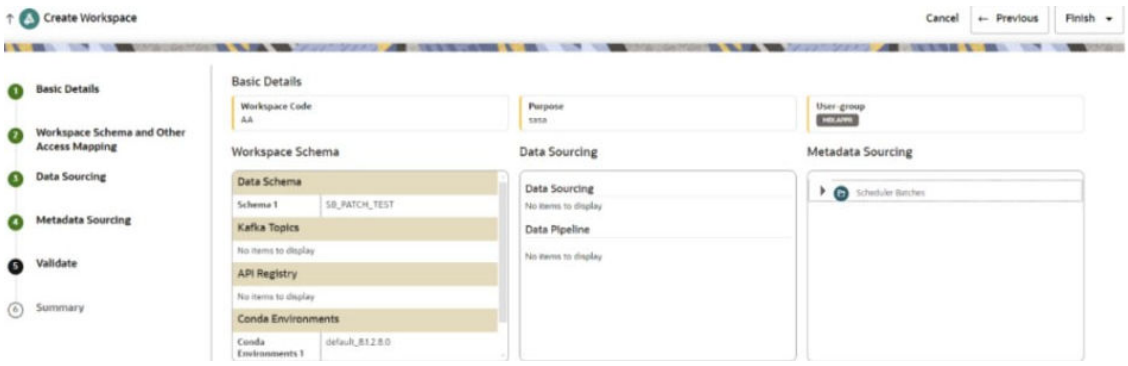
Figure 5-44 Metadata Sourcing



Validate Workspace

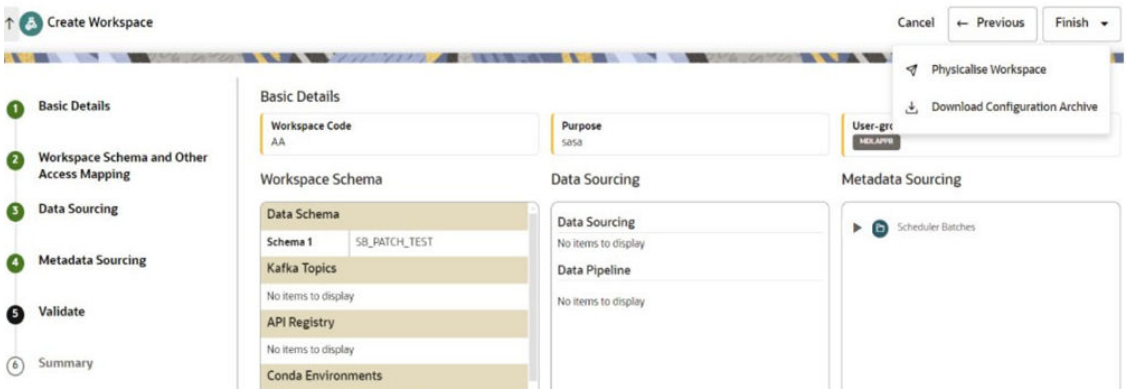
You can validate the Basic details, Workspace schema, Data Sourcing and Metadata sourcing before you physicalize the workspace.

Figure 5-45 Validate Workspace



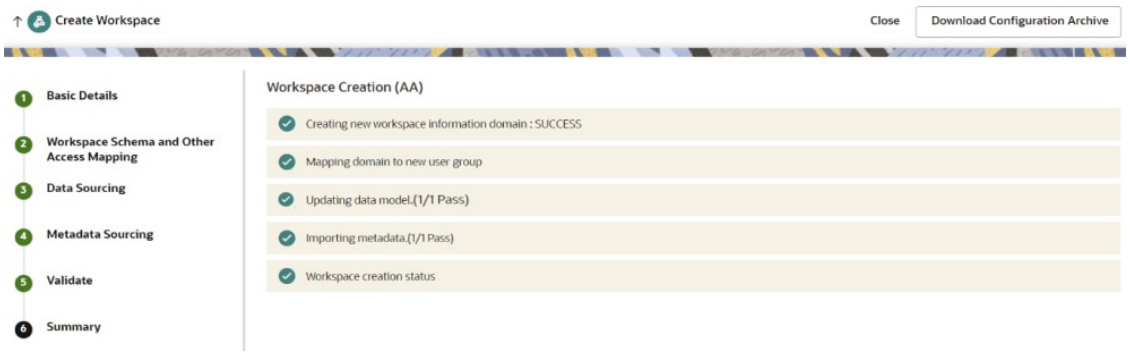
To Physicalize the workspace, Click **Finish** and then select **Physicalize Workspace**.

Figure 5-46 Physicalize Workspace



Summary
You can view summary of the created workspace.

Figure 5-47 Summary



5.3.2 Prerequisites for Creating Sandbox Workspace

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
4. Create a new data store for the sandbox schema

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

5.3.3 Creating Sandbox Workspace

On the **Workspace Summary page**, click **Add Workspace**. The Create Workspace window is displayed with the following process:

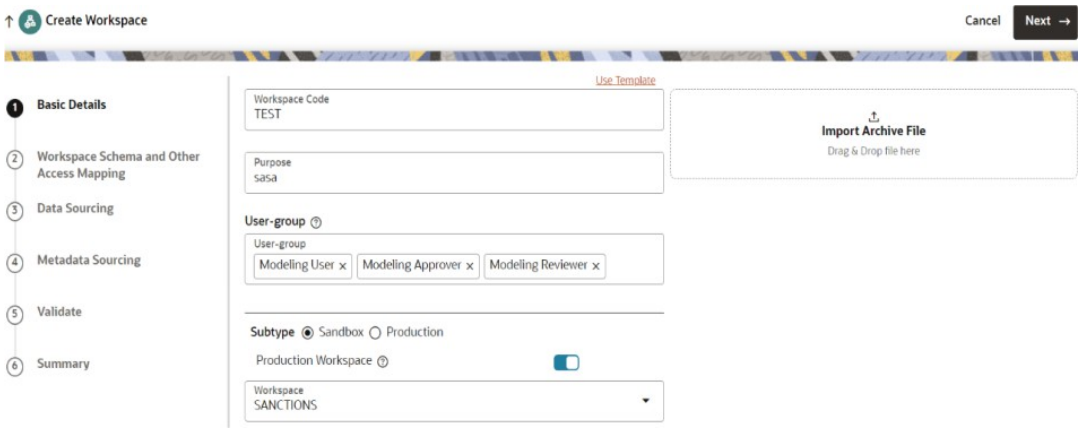
1. Basic Details
2. Workspace Schema
3. Data Sourcing
4. Metadata Sourcing
5. Validate
6. Summary

Basic Details
To create basic details of the sandbox workspace, follow these steps:

1. Provide the requested details for **Workspace Code** and **Purpose**.
2. From the drop-down list, select the **User-group**.
3. Select the subtype as **Sandbox Workspace**.
4. Enable the **Production Workspace** button.

5. From the drop-down list, select the **Sanction Production Workspace**.
6. Click **Next**.

Figure 5-48 Basic Details



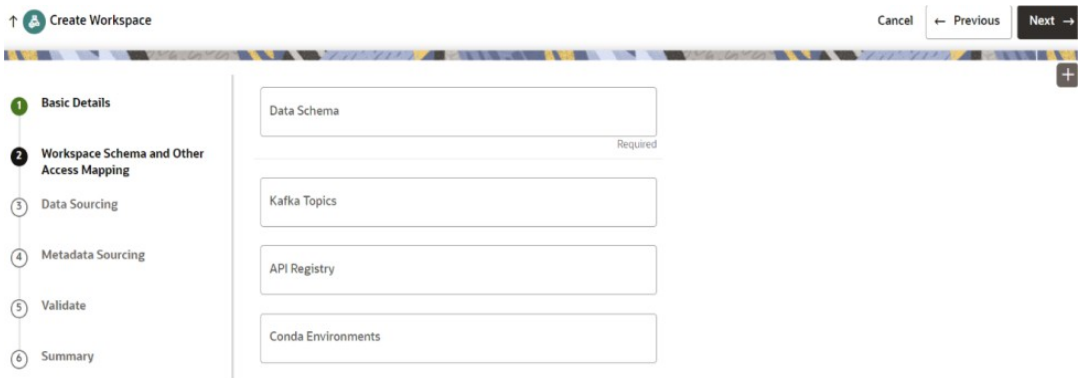
Workspace Schema
To create the workspace schema, follow these steps:

1. Select the newly created data store as **Data Schema**.

> **Note:**
>
> Leave the **Kafka Topics** and **API Configurations** fields as blank.

2. Select the following **Conda Environments**:
   a. **default_8.1.2.8.0**
   b. **ml4aml_8.1.2.8.0**
3. Click **Next**.

Figure 5-49 Workspace Schema



Data Sourcing
To select Database objects from the data stores, follow these steps:

1. From the **Source Data Schema** drop-down list, select the **Data Store**.

2. From the **Object Type** drop-down list, select the **Table**.
3. Select following tables from the sanction production data store where it is having sufficient historical data.
   CUST

   CS_ALERTS

   EXTERNAL_ENTITY

   CUST_ADDR

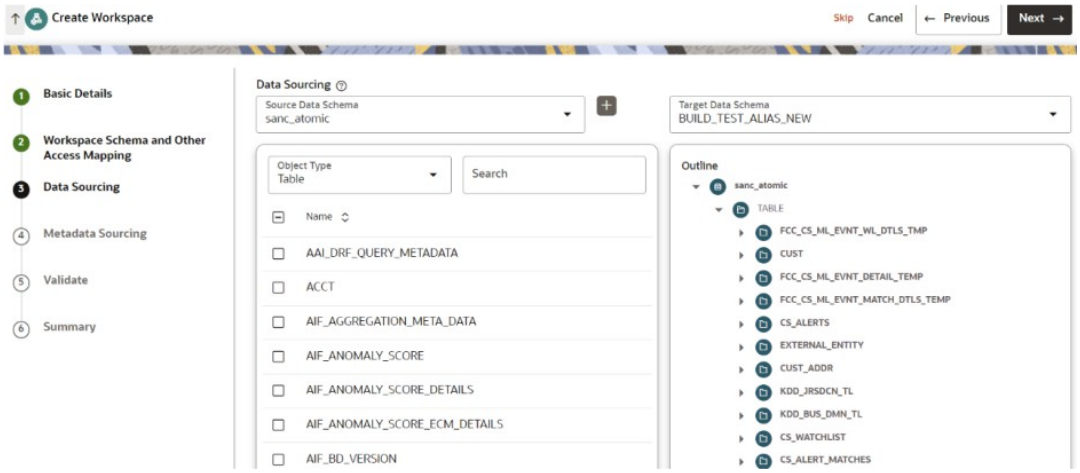
   KDD_JRSDCN_TL

   KDD_BUS_DMN_TL

   CS_WATCHLIST

   CS_ALERT_MATCHES

Figure 5-50 Data Sourcing

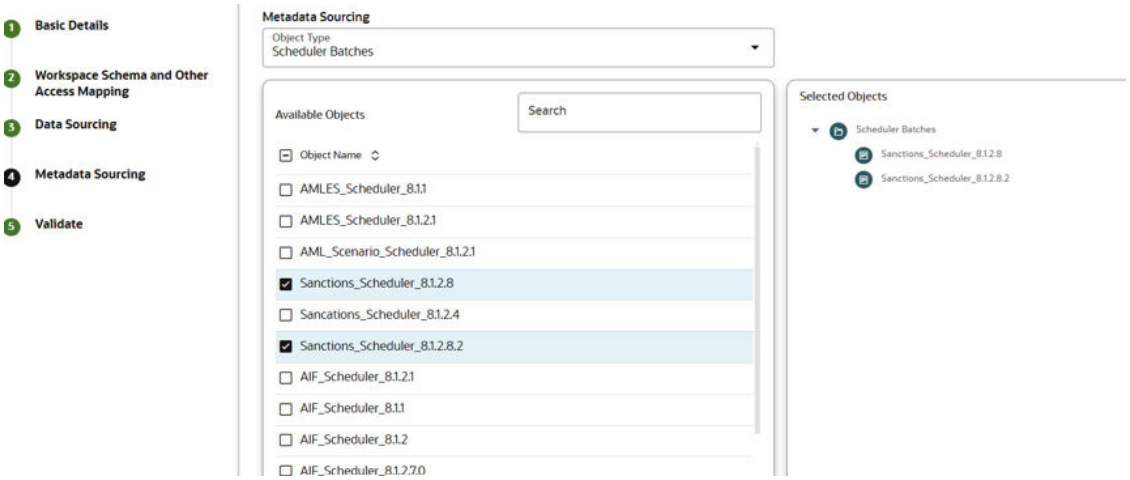


Metadata Sourcing

To select available objects from the Metadata Sourcing, follow these steps:

1. From the **Object Type** drop-down list, select **Scheduler Batches**.
2. In the **Available Objects**, select **Sanctions_Scheduler_8.1.2.8** and **Sanctions_Scheduler_8.1.2.8.2**.
3. Click **Next**.

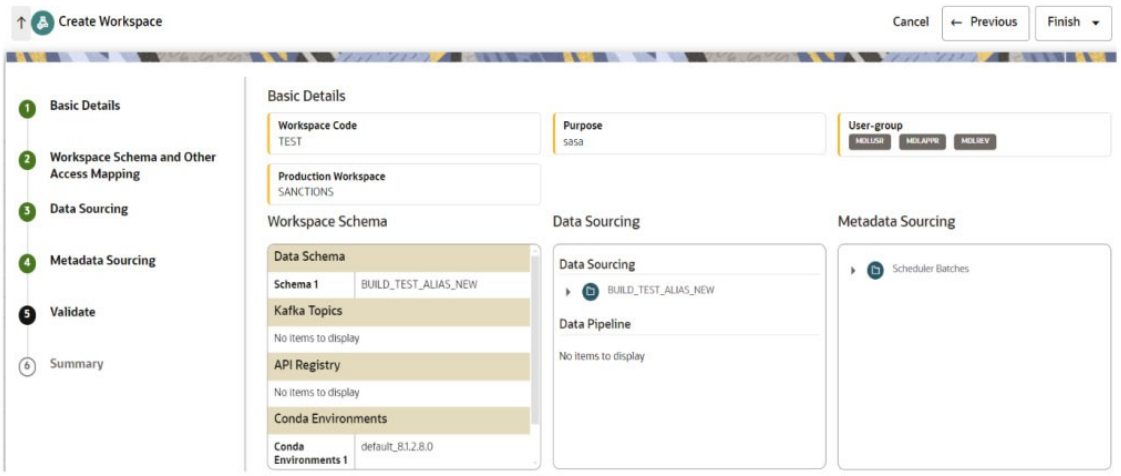
Figure 5-51 Metadata Sourcing



Validate Workspace

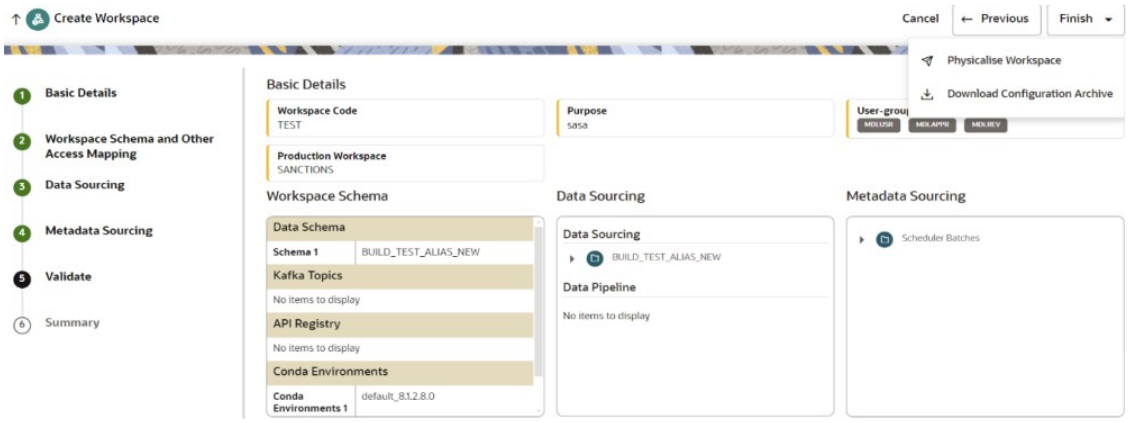
You can validate the Basic details, Workspace schema, Data Sourcing and Metadata sourcing before you physicalize the workspace.

Figure 5-52 Validate Workspace



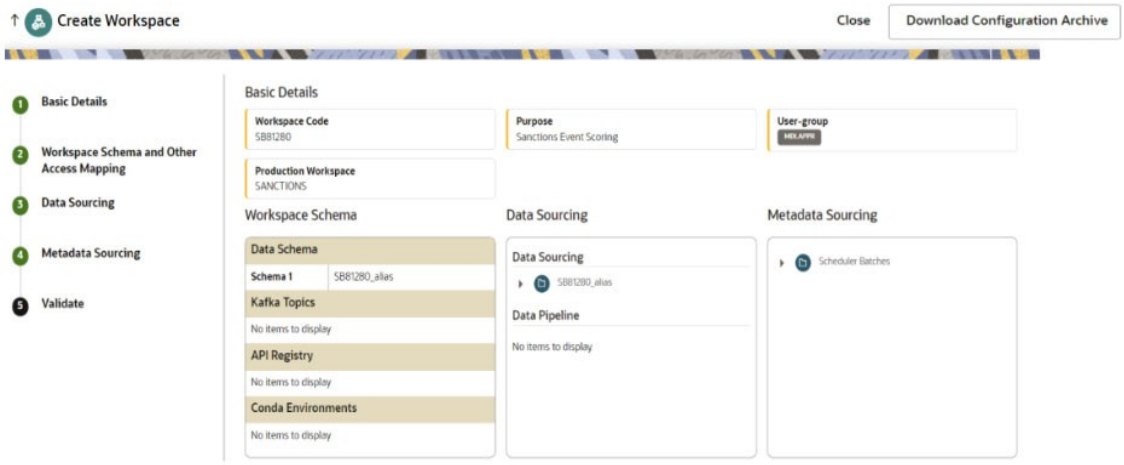
To Physicalise the workspace, Click **Finish** and then select **Physicalise Workspace**.

Figure 5-53 Physicalise Workspace



Summary
You can view summary of the created workspace

Figure 5-54 Summary



5.3.4 Populating Sandbox Workspace

To populate the sandbox workspace, see How to Populate the Sandbox Workspace section.

5.3.5 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-5 Utilities for Workspace and Notebook

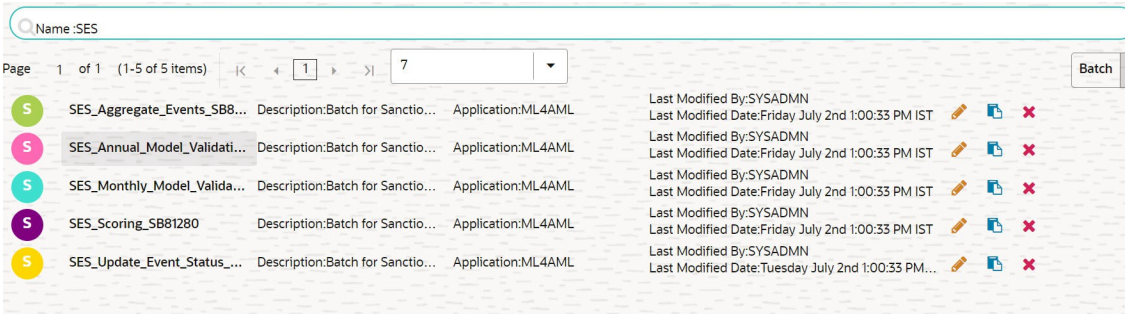
Utility	Sandbox Workspace	Production Workspace	Command
importWorkspaceSQL.sh	Yes	Yes	./importWorkspaceSQL.sh - w <workspace_wallet_alias>
importNotebooksSES.sh	Yes	Yes	./importNotebooksSES.sh -w <workspace_code>

5.3.6 Batch Framework for Sanctions Event Scoring

The following batches are available in the out-of-the-box for Sanctions Event Scoring (SES) framework:

- **SES_Aggregate_Events**
- **SES_Scoring**
- **SES_Annual_Model_Validation**
- **SES_Monthly_Model_Validation**
- **SES_Update_Event_Status**

Figure 5-55 Define Batch for Sanctions Event Scoring



SES Aggregate Events

- This pre-seeded batch will be available in all the workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the **Sandbox** workspace.

- This batch creates base features for sanctions event scoring training in the sandbox workspace.

Batch and Task parameters
The batch contains a single task named **Aggregate_Events**.

Figure 5-56 Define Task for Aggregate_Events

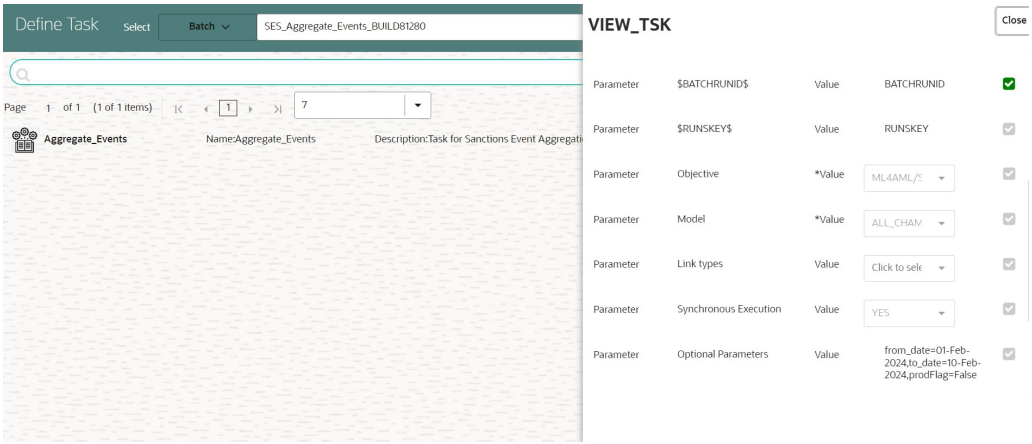
Task: Aggregate_Events, Task Parameters

- Objective folder for this task:

  ```
  Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring / Batch / Aggregate Events
  ```

- Optional Parameters:
  - **from_date**: Start date for Historic Data lookup in **DD-Month-YYYY** format.
  - **to_date**: End Date for Historic Data lookup in **DD-Month-YYYY** format.
  - **prodFlag**: Flag to indicate Training/Scoring scenario. The option is **True** or **False**. For sandbox/historic training scenarios, the **prodFlag** should be set to **False**.
  - **is_aai_batch**: This flag indicates whether the batch is called from the Sanctions application or from Compliance Studio. The default setting is True, indicating that the batch originates from the Sanctions application. Conversely, setting the flag to False signifies that the batch is called from Compliance Studio.
  - **sanctions_batch_run_id**: The batch run ID will be used when the is_aai_batch flag is set to False.
    For Example: from_date=01-Feb-2024,to_date=10-Feb-2024,prodFlag=False, is_aai_batch=True, sanctions_batch_run_id=SES_B1.
  - **Edit** Task Parameters and **Save**.

Figure 5-57 Aggregate_Events



> **Note:**
>
> * Once the batch execution is successful, the results are available in the **ML4AML_SES_EVENT_INPUT** table. For more information on the table structure, see the OFS Compliance Studio Data Model Reference Guide.
> * You can view execution status of the batch in the ML4AML_SES_EXECUTION_STATUS table and if batch execution fails, you can view ML4AML_SES_EXECUTION_ERRORS table for debugging.

SES Scoring
This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the Production workspace.

Batch and Task Parameters

The batch contains the following tasks:

- Task1: **Aggregate_Events**
- Task2: **ML_Scoring**

Figure 5-58 SES Scoring

Task 1: Aggregate_Events, Task Parameters

- Objective folder for this task:

```
Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring /Batch /
Aggregate Events
```

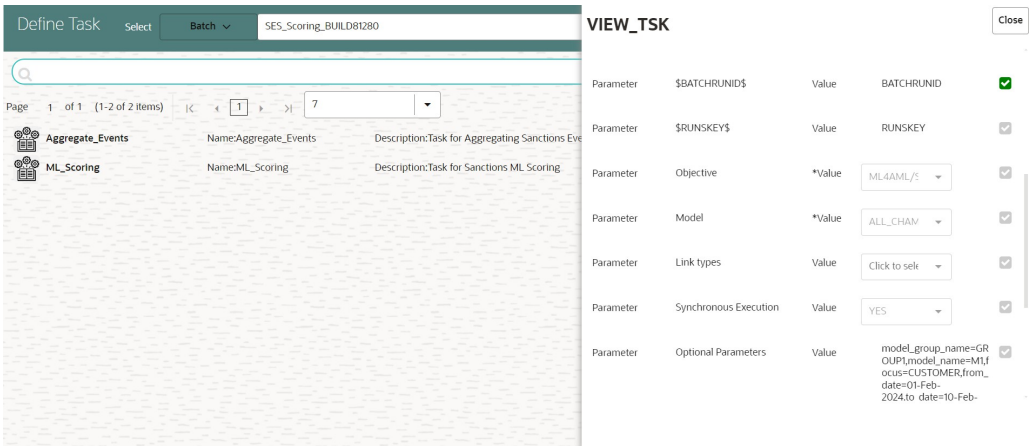
> **Note:**
>
> Do not change any parameter, except Optional Parameters.

- Optional Parameters:
  - **prodFlag**: Flag to indicate Training/Scoring scenario. The option is **True** or **False**. For production / scoring scenarios, the **prodFlag** should be set to **True**.
  - **is_aai_batch**: This flag indicates whether the batch is called from the Sanctions application or from Compliance Studio. The default setting is True, indicating that the

batch originates from the Sanctions application. Conversely, setting the flag to False signifies that the batch is called from Compliance Studio.

- **sanctions_batch_run_id**: The batch run id that will be used when the flag is_aai_batch is set to False. The value of V_BATCH_RUN_ID for this should be taken from cs_batch_run table corresponding to the D_MIS_DATE.
  For Example: prodFlag=True, is_aai_batch=True, sanctions_batch_run_id=SES_B1.
- **Edit** Task Parameters and **Save**.

Figure 5-59 Edit Task for Aggregate_Events



> **Note:**
>
> * Once the batch execution is successful, the results are available in the **ML4AML_SES_EVENT_INPUT** table. For more information on the table structure, see the OFS Compliance Studio Data Model Reference Guide.
> * You can view execution status of the batch in the ML4AML_SES_EXECUTION_STATUS table and if batch execution fails, you can view ML4AML_SES_EXECUTION_ERRORS table for debugging.

Task 2: ML_Scoring, Task Parameters

- Objective folder for this task:

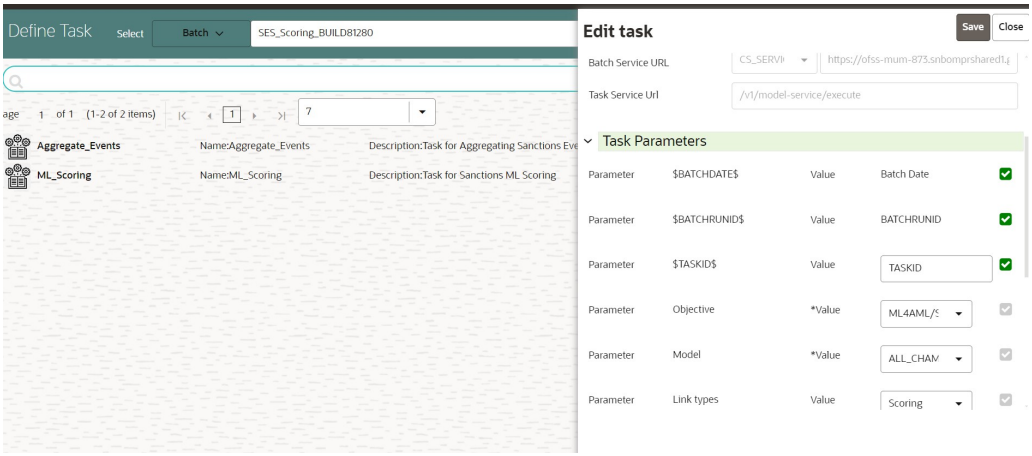
```
Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring / Batch /
Model
```

> **Note:**
>
> Do not change any parameter, except **Optional Parameters**.

- Optional Parameters:
  - **threshold**: Input threshold or cutoff. Events will be created if the score of an entity exceeds the threshold.

- **debug_flag**: It is used for debugging purpose. By default, it is **False**.
- **is_aai_batch**: This flag indicates whether the batch is called from the Sanctions application or from Compliance Studio. The default setting is True, indicating that the batch originates from the Sanctions application. Conversely, setting the flag to False signifies that the batch is called from Compliance Studio.
- **sanctions_batch_run_id**: The batch run id that will be used when the flag is_aai_batch is set to False. The value of V_BATCH_RUN_ID for this should be taken from cs_batch_run table corresponding to the D_MIS_DATE.
  For Example: threshold=0,debug_flag=False,is_aai_batch=True, sanctions_batch_run_id=SES_B1.
- **Edit** Task Parameters and **Save**.

Figure 5-60 Edit Task for Scoring



> **Note:**
>
> Once the batch execution is successful, the results are available in the ML4AML_SES_EVENT_SCORE and ML4AML_SES_EVENT_SCORE_DETAILS tables. For more information on these table structure, see the OFS Compliance Studio Data Model Reference Guide.

SES Annual Model Validation

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

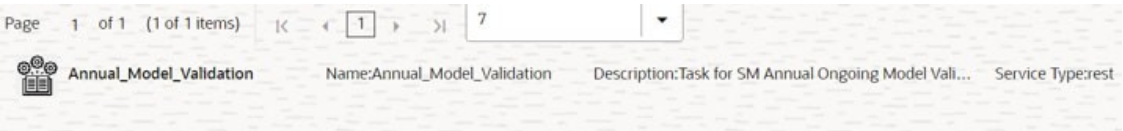
> **Note:**
>
> This batch has to be executed in the **Production** workspace.

- This batch shows ongoing model performance annually.

Batch and Task Parameters

The batch contains a single task named Annual_Model_Validation.

Figure 5-61 Annual Model Validation



Task: Annual_Model_validation, Task Parameters

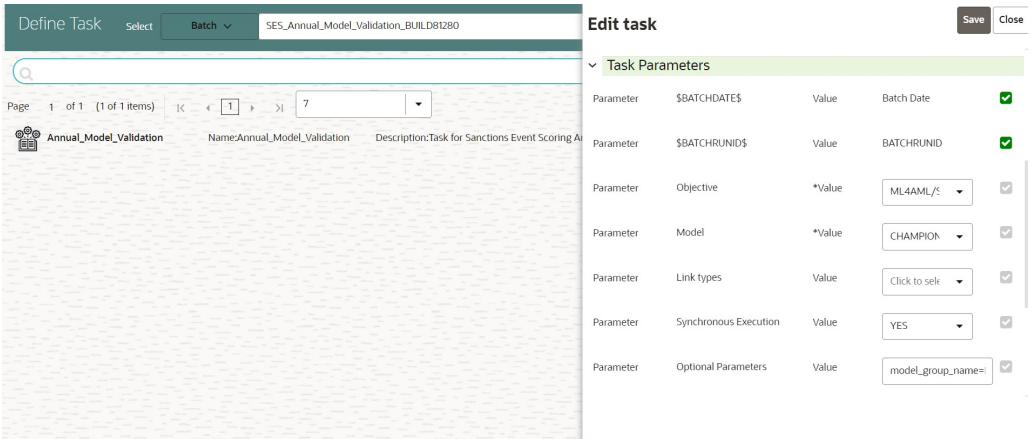
- Objective folder for this task:

  ```
  Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring / Batch /
  Monitoring / Annual
  ```

> **Note:**
>
> Do not change any parameter except **Optional Parameters** and this 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.
  - **from_date**: Start Date for Historic Data lookup in **DD-MM-YYYY** format.
  - **to_date**: End Date for Historic Data lookup in **DD-MM-YYYY** format.
  - **model_id_list**: To use the deployed model, you need to pass the parameter as Deployed.
    For example: model_group_name=LOB1,model_id_list=Deployed,from_date=01-Jan-2016,to_date=31-Dec-2017
  - **Edit** Task Parameters and **Save**.

Figure 5-62 Edit Task for Annual Model Validation



> **Note:**
>
> The Annual Model Validation batch shows output metrics in the notebook only and it will not store in any of the data tables.

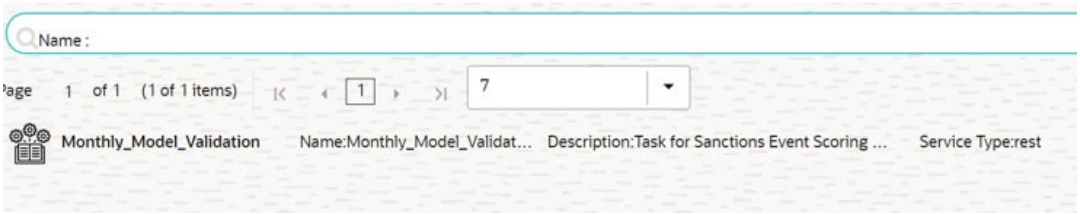
SES Monthly Model Validation

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).

> **Note:**
>
> This batch has to be executed in the **Production** workspace.

- This batch shows ongoing model drift analysis and data quality with respect to new data every month (monthly).

Figure 5-63 Monthly Model Validation



Batch and Task Parameters

The batch contains a single task named Monthly_Model_Validation.

Task: Monthly_Model_Validation, Task Parameters

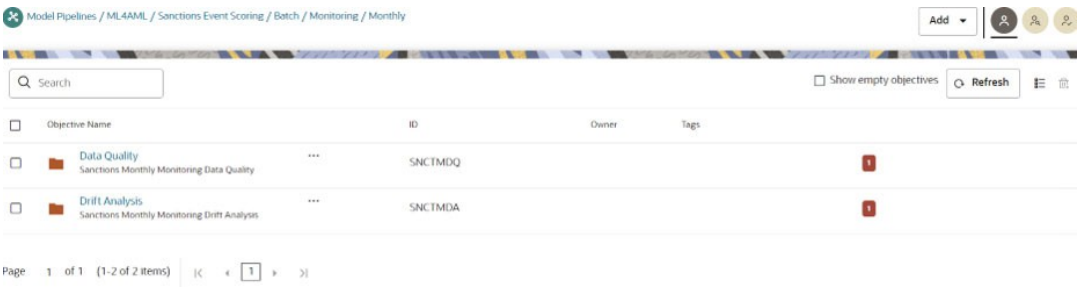
- Objective folder for Data Quality:

```
Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring / Batch /
Monitoring / Monthly / Data Quality
```

- Objective folder for Drift Analysis:

```
Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring / Batch /
Monitoring / Monthly / Drift Analysis
```

Figure 5-64 Monthly Model Validation

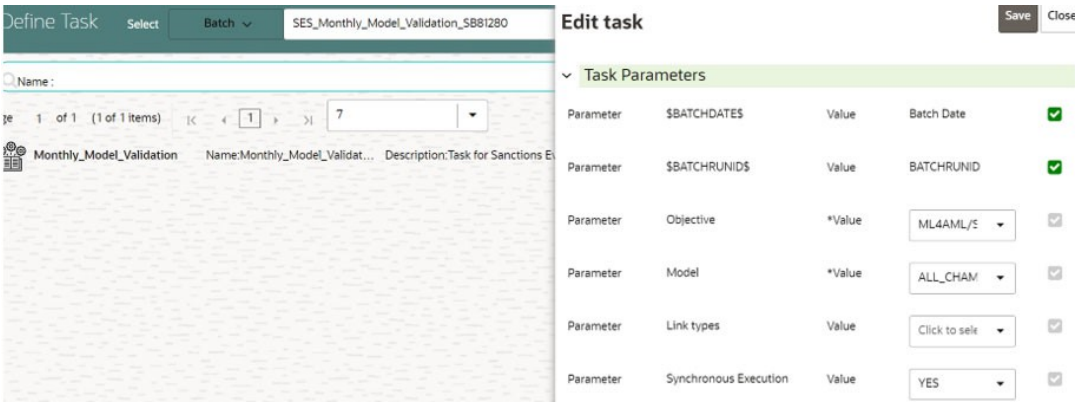


> **Note:**
>
> Do not change any parameter except **Optional Parameters** and this Optional Parameters can be edited from the **Schedule Batch** option.

- Optional Parameters:
  - **model_group_name**: Name of the Model Groups for which the model has been trained. Example: LOB1.
  - **from_date**: Start Date for Historic Data lookup in **DD-MM-YYYY** format.
  - **to_date**: End Date for Historic Data lookup in **DD-MM-YYYY** format.
  - **model_id**: To use the deployed model, you need to pass the parameter as Deployed.
  - **FEATURE_INCLUDE**: List of features to be included for data quality. The default value is **None** which means it includes everything.
  - **FEATURE_EXCLUDE**: List of features to be excluded for data quality. The default value is **None** which means it excludes nothing.
  - **Number_Of_Bins**: Number of bins to be used in discretizing (scalar). The default value is **9**.
  - **Boot_Strap_Samples**: Number of bootstrap samples 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,from_date=01-Jan-2016,to_date=31-Dec-2017, model_id=Deployed,Number_Of_Bins=9,Boot_Strap_Samples=5,Standard_Deviation_Band _Sigma=2,FEATURE_INCLUDE=None,FEATURE_EXCLUDE=None
- Edit Task Parameters and Save.

Figure 5-65 Edit Task for Monthly Model Validation



> **Note:**
>
> The Monthly Model Validation batch shows output metrics in the notebook only and it will not store in any of the data tables.

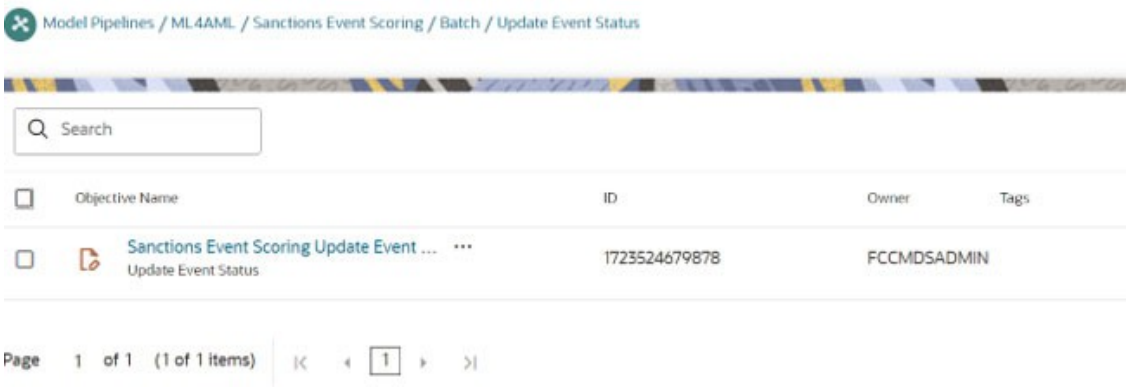
SES Update Event Status

- This pre-seeded batch will be available in all workspaces (Production and Sandboxes).
- This batch is used to update the Event Status in the ML4AML_SES_EVENT_INPUT table.

Batch and Task Parameters

The batch contains a single task named Update_Event_Status.

Figure 5-66 Update Event Status



Task: Sanctions Event Scoring Update Event Status, Task Parameters

- Objective folder for this task:

```
Home / Modeling / Pipelines / ML4AML / Sanctions Event Scoring / Batch /
Update Event Status
```

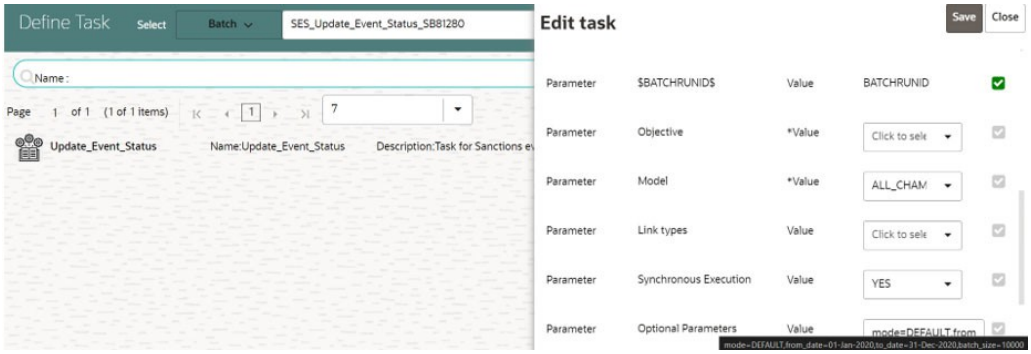
> **Note:**
>
> Do not change any parameter except **Optional Parameters** and this Optional Parameters can be edited from the **Schedule Batch** option.

- Optional Parameters:
  - **mode**: Name of the mode which can be either **FILE** when user uploads the CSV file or **DEFAULT** which internally updates the event status.

    > **Note:**
    >
    > By default, it is set to mode=DEFAULT. If you are using mode=FILE, then execute batch using this Update Event Status from CSV File section.

  - **from_date**: Start Date for Historic Data lookup in **DD-MM-YYYY** format.
  - **to_date**: End Date for Historic Data lookup in **DD-MM-YYYY** format.
    For example: mode=DEFAULT,from_date=01-Jan-2020,to_date=31-Dec-2020
  - **Edit** Task Parameters and **Save**.

Figure 5-67 Edit Task for Update Event Status



> **Note:**
>
> * Once the batch execution is successful, the results are available in the ML4AML_SES_EVENT_INPUT table. For more information on the table structure, see the OFS Compliance Studio Data Model Reference Guide.
> * You can view execution status of the batch in the ML4AML_SES_EXECUTION_STATUS table and if batch execution fails, you can view ML4AML_SES_EXECUTION_ERRORS table for debugging.

Update Event Status from CSV File

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

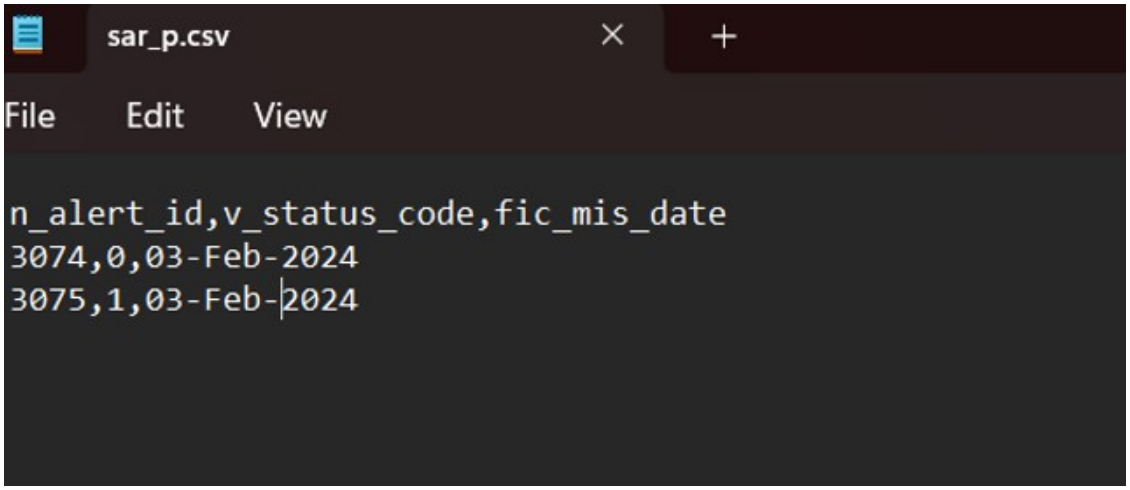
- **mode**: Name of the mode which can be either **FILE** when user uploads the CSV file or **DEFAULT** which internally updates the event status.
- **from_date**: Start Date for Historic Data lookup in **DD-MM-YYYY** format.
- **to_date**: End Date for Historic Data lookup in **DD-MM-YYYY** format.
- **batch_size**: It reads data in the corresponding batch size from the CSV file.

For example: mode=FILE, from_date=01-Jan-2020, to_date=31-Dec-2024,batch_size=52

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

This sample CSV is shipped with headers and two sample rows showing the format of each column.

Figure 5-68 Snapshot of sar_p.csv



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

- **N_ALERT_ID**: The Alert/Event ID for the event to be updated.
- **V_STATUS_CODE**: Indicates status as True / False Positive Events. The available options are:
  - **1**: Indicates the True Positive Events
  - **0**: Indicates the False Positive Events
  - **None**: If Event is neither 1 nor 0, then it is categorized as None.
- **FIC_MIS_DATE**: The date on which event is generated. The format should be **DD-Mon-YYYY**.

> **Note:**
>
> - In the CSV file, the user is expected to populate non-Null data for all the columns except V_STATUS_CODE.
> - Once the batch execution is successful, the results are available in the ML4AML_SES_EVENT_INPUT table. For more information on the table structure, see the OFS Compliance Studio Data Model Reference Guide

5.3.7 Execute Batch

To execute the batch, see How to Execute Batch section.

5.3.8 Monitor Batch

To monitor the batch, see How to Monitor Batch section.

5.4 AML Event Scoring

This section explains about AML Event Scoring use case.

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
4. Create a new data store for the sandbox schema
5. Register Conda Environment in BD Production Workspace

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

To register Conda Environment in BD Production Workspace, see How to Register Conda Environment in BD Production Workspace section.

Schema Grants for AML Event Scoring

In Production Workspace:

For AML event scoring, production workspace is attached to the BD atomic schema, whereas AML event scoring reads tables/data from ECM atomic schema.

Assuming both BD and ECM atomic schemas are on the same database, hence SELECT grant on few of the listed ECM tables, to be provided to BD atomic schema by running set of grants in ECM atomic schema.

To grant schema for AML Event scoring in the Production workspace, follow these steps:

1. Execute the following query in the ECM atomic schema and replace <BD_ATOMIC_SCHEMA> placeholder with actual value of the BD atomic schema:

```
select 'GRANT SELECT ON '||TABLE_NAME ||' TO <BD_ATOMIC_SCHEMA>;'
GRANTS FROM USER_TABLES WHERE TABLE_NAME like 'FCC_%';


select 'GRANT SELECT ON '||TABLE_NAME ||' TO <BD_ATOMIC_SCHEMA>;'
GRANTS FROM USER_TABLES WHERE TABLE_NAME like 'KDD_%';
```

   These two queries will return you set of grants which needs to be executed in the ECM atomic schema.

2. Copy the grants and execute them in the ECM atomic schema.

In Sandbox Workspace:

Create a new schema in the database where sandbox target schema exists. Use ECM production data dump to populate this new schema and let's call this new schema as ECM Dump Schema.

For AML event scoring, Sandbox workspace is attached to empty target schema called as Sandbox Schema, where data has to come from ECM Dump schema. The ECM Dump and

Sandbox schemas are on the same database, hence SELECT grant on few of the listed ECM Dump tables, to be provided to Sandbox schema by running set of grants in the ECM Dump atomic schema.

To grant schema for AML Event scoring in the Sandbox workspace, follow these steps:

1. Execute the following query in the ECM Dump schema and replace <SANDBOX_SCHEMA> placeholder with actual value of the sandbox schema:

```
select 'GRANT SELECT ON '||TABLE_NAME ||' TO <SANDBOX_SCHEMA>;' GRANTS
FROM USER_TABLES WHERE TABLE_NAME like 'FCC_%';


select 'GRANT SELECT ON '||TABLE_NAME ||' TO <SANDBOX_SCHEMA>;' GRANTS
FROM USER_TABLES WHERE TABLE_NAME like 'KDD_%';
```

   These two queries will return the set of grants which needs to be executed in the ECM Dump schema.

2. Copy the grants and execute them in the ECM Dump schema.

5.4.1 Creating Sandbox Workspace

To create the sandbox workspace, see How to Create Sandbox Workspace section.

5.4.2 Populating Sandbox Workspace

This section is not applicable for AML Event Scoring use case as it does not source any data tables into the workspace schema. Users need to populate the data using pre-seeded batches.

5.4.3 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-6 Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh - w <workspace_wallet_alias>`
`importNotebooksAMLES. sh`	Yes	Yes	`./importNotebooksAMLES.sh - w <workspace_code>`

5.4.4 Batch Framework for AML Event Scoring

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

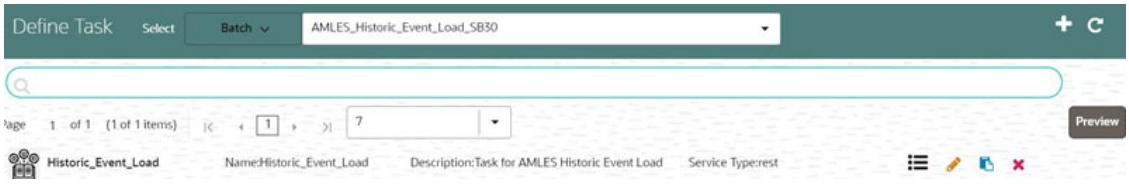
1. **AMLES Historic Event Load**
2. **AMLES Scoring**
3. **AMLES Update Event Labels**

5.4.4.1 AMLES Historic Event Load

- This is a pre-seeded batch and will be available in all workspaces (production & sandboxes)
- This Batch is to be executed in the **Sandbox** workspace.
- This Batch pulls data from the ECM system used for ML Model training in the sandbox.

Batch and Task Parameters
The batch contains a single task named **Historic_Event_Load**.

Figure 5-69 Task Details for Historic Event Load



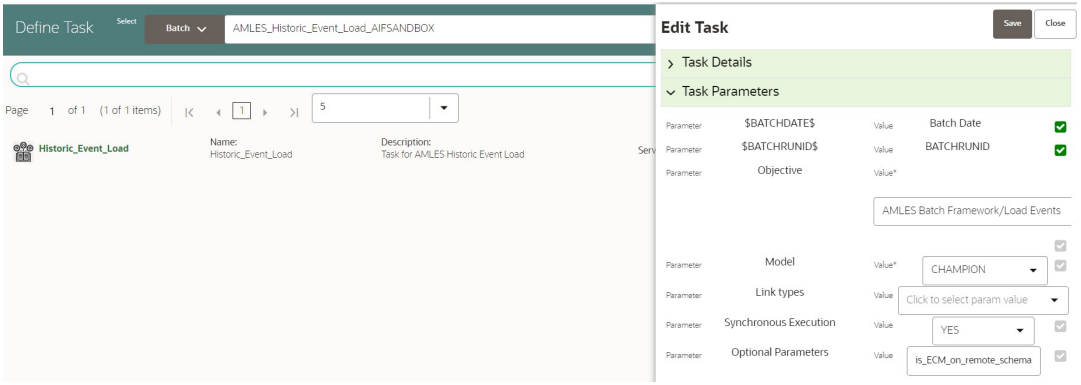
Historic_Event_Load, Task Parameters

- 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 5-70 Define Task



5.4.4.2 AMLES Scoring

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

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.

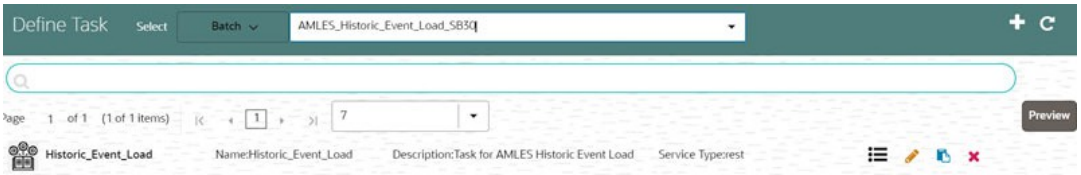
Output of AMLES event-scoring is stored in following static tables in BD schema.

- amles_event_score
- amles_event_score_details

Batch and Task Parameters
The batch contains the following tasks:

- **Scoring_Event_Data_Load**
- **ML_Scoring**

Figure 5-71 Define Task



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 the **Schedule Batch** option.
  - Do not change any other batch /task parameters, except **Optional Parameters**.

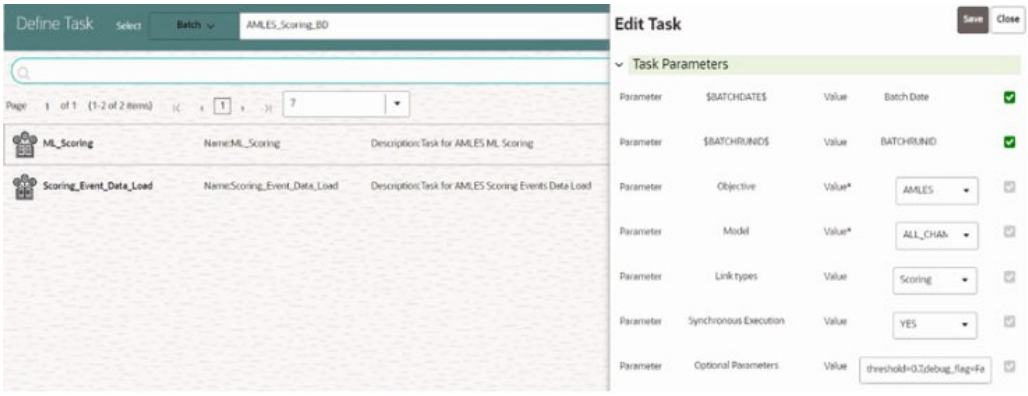
ML_Scoring, Task Parameters

- 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 5-72 Edit Task for AMLES_Scoring



5.4.5 Execute Batch

To execute the batch, see How to Execute Batch section.

5.4.6 Monitor Batch

To monitor the batch, see How to Monitor Batch section.

5.5 Customer Segmentation and Anomaly Detection

This section explains about Customer Segmentation and Anomaly Detection use case.

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
4. Create a new data store for the sandbox schema
5. Register Conda Environment in BD Production Workspace

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

To register Conda Environment in BD Production Workspace, see How to Register Conda Environment in BD Production Workspace section.

5.5.1 Creating Sandbox Workspace

To create the sandbox workspace, see How to Create Sandbox Workspace section.

5.5.2 Populating Sandbox Workspace

To populate the sandbox workspace, see How to Populate the Sandbox Workspace section.

5.5.3 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-7 Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh - w <workspace_wallet_alias>`

Table 5-7 (Cont.) Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
importNotebooksAIF.sh	Yes	Yes	./importNotebooksAIF.sh - w <workspace_code>

5.5.4 Data Movement

The Customer Segmentation and Anomaly Detection use case make use of time series data and time series length can go up to six months to one year. When model is freshly deployed to production, given production may not have enough history of time series data; hence data movement from sandbox to production is required.

> **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
> ```

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.

Outputs

`AIF_DATA_UNSUP.dmp` will be created as part of successful execution.

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'`

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.

Outputs

On successful execution, AIF_BEHAVIORAL_DATA_UNSUP will be populated for the model group.

Execution Logs

`IMP_AIF_DATA _UNSUP.log` will be created as part of the execution in case of any issues.

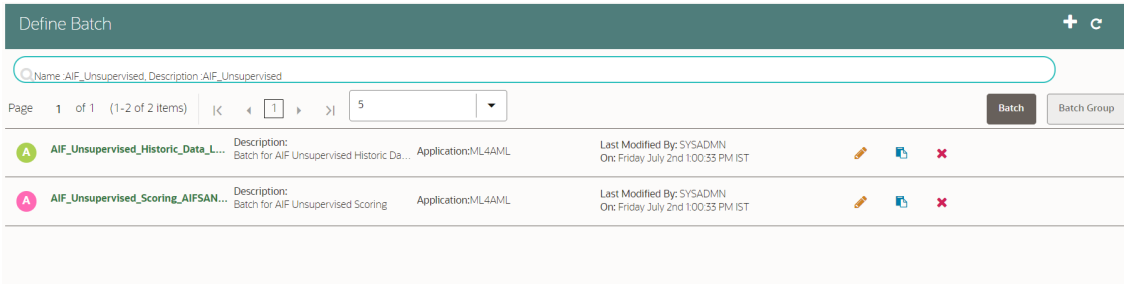
> **Note:**
>
> DATA_PUMP_DIR for any oracle database server can be found out using the following query (using sysdba).
>
> ```
> select * from dba_directories where directory_name =
> 'DATA_PUMP_DIR'
> ```
>
> .

5.5.5 Batch Framework for Customer Segmentation and Anamoly Detection

The following batches are available out of the box:

1. **Unsupervised Historic Data Load**
2. **Unsupervised Scoring**

Figure 5-73 Define Batch



5.5.6 Execute Batch

To execute the batch, see How to Execute Batch section.

5.5.7 Monitor Batch

To monitor the batch, see How to Monitor Batch section.

5.6 Customer Risk Scoring

This section explains about Customer Risk Scoring use case.

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
4. Create a new data store for the sandbox schema
5. Register Conda Environment in BD Production Workspace

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

To register Conda Environment in BD Production Workspace, see How to Register Conda Environment in BD Production Workspace section.

5.6.1 Creating Sandbox Workspace

To create the sandbox workspace, see How to Create Sandbox Workspace section.

5.6.2 Populating Sandbox Workspace

To populate the sandbox workspace, see How to Populate the Sandbox Workspace section.

5.6.3 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-8 Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh - w <workspace_wallet_alias>`
`importNotebooksAIF.sh`	Yes	Yes	`./importNotebooksAIF.sh - w <workspace_code>`

5.6.4 Obtaining SAR Labels for Customer Risk Scoring

Obtain the SAR Information for Sandbox
Disposition/SAR information of the historical alerts that are required in sandbox for the purpose of supervised machine learning model training. SAR information acts as a target/depended variable for the model training.

Populate Investigated Entity Details

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 Compliance Studio.

> **Note:**
>
> "**aif**" is just a package that is available as part of compliance Studio.

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

Figure 5-74 Aif Load SAR Data

```
2
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.

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 5-75 Aif Load Sars from CSV

```
2
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

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 Compliance Studio schema table, `aif_case_information`.

Figure 5-76 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.

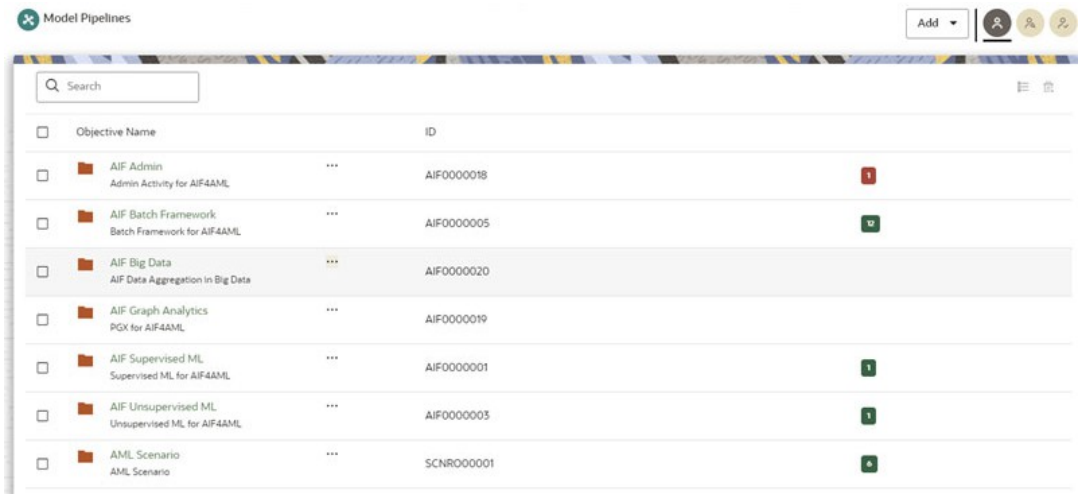
5.6.5 Obtain SAR information for Production

Disposition/SAR information of the production alerts sent for investigations that are required for model validations to see the deployed model is performing well in the production or model is deteriorating.

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 5-77 AIF Admin notebook



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

Figure 5-78 Open Notebook in Pipeline Designer

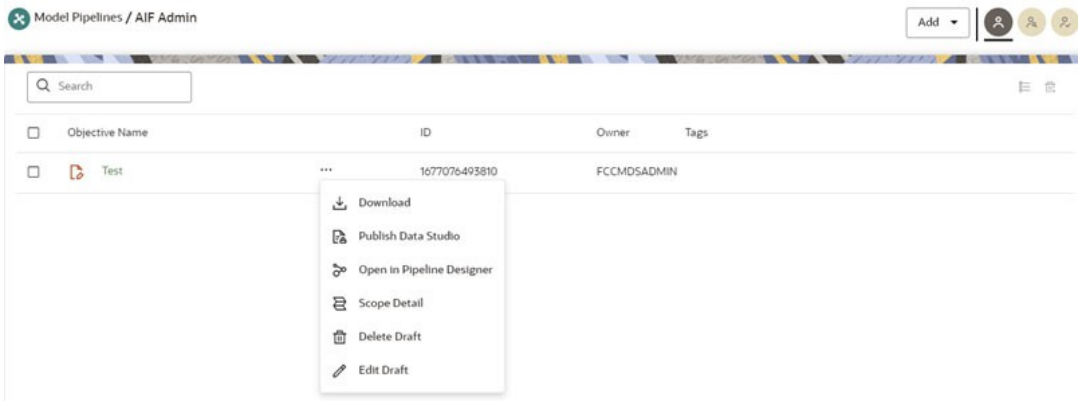
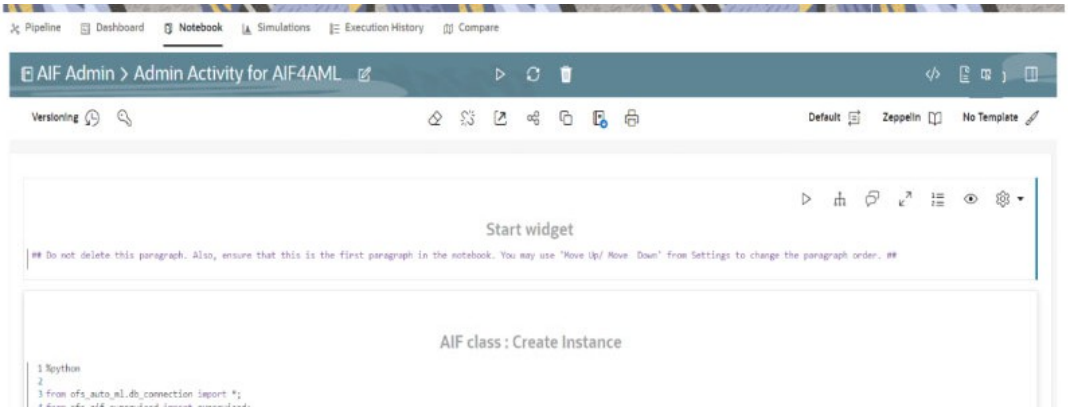


Figure 5-79 Notebook tab in Pipeline Designer



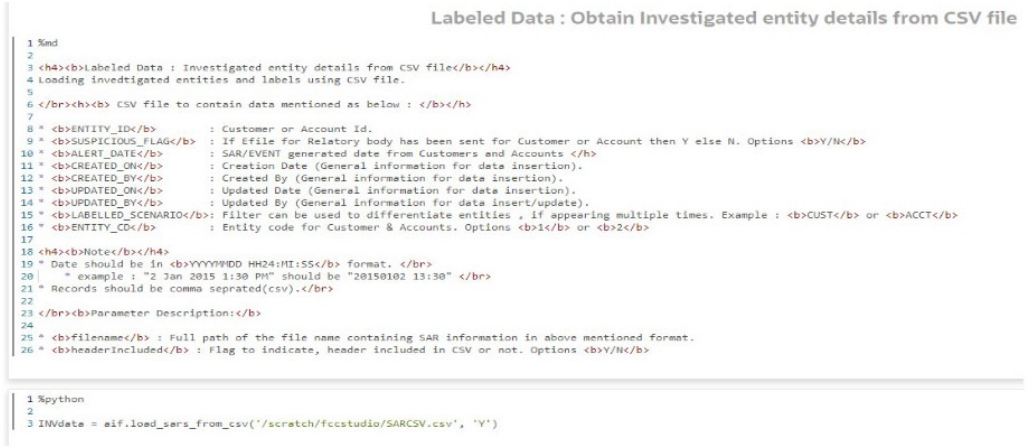
- Admin notebook facilitates the following functionalities to build Machine Learning Models:

- 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 5-80 Obtaining Investigated Labels from CRR-ECM

Figure 5-81 Obtaining Investigated Labels from CSV file



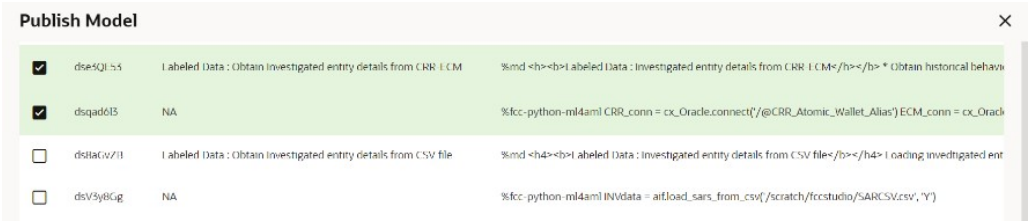
Obtain Labels in Production Workspace

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

Perform the following:

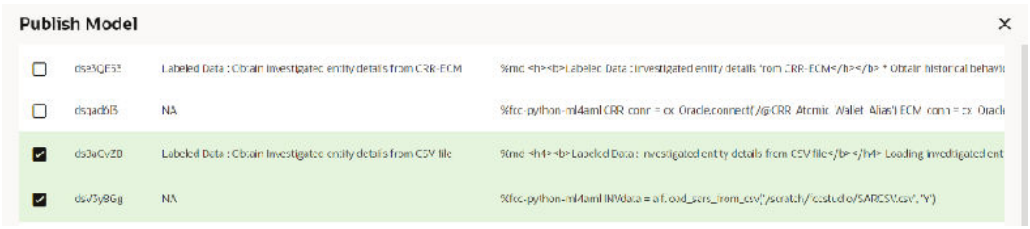
1. Obtaining labels for the following:
   - From CRR-ECM, Publish and Deploy the following two paragraphs:

Figure 5-82 Obtaining Investigated Labels from CRR-ECM



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

Figure 5-83 Obtaining Investigated Labels from CSV file



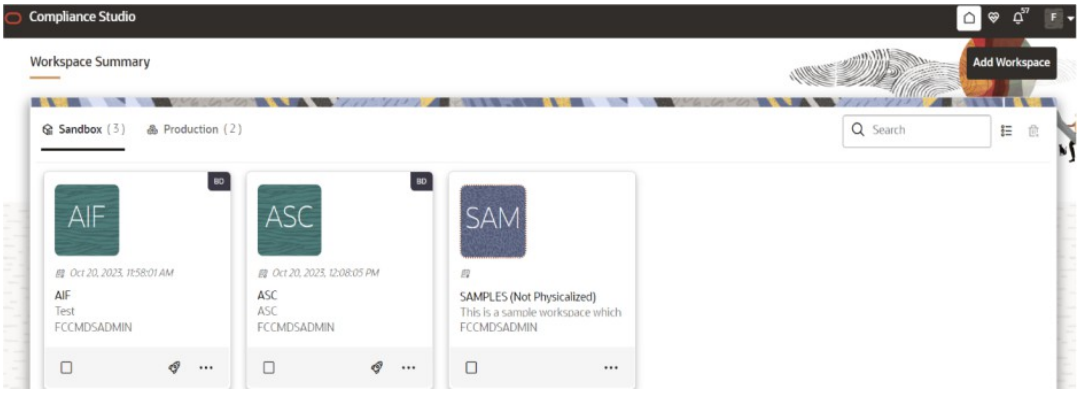
For more details on Publish and Deploy, see the How to Deploy the Model section in OFS Compliance Studio Use Case Guide.

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.

5.6.5.1 Create a New Batch for Obtaining Investigated Entities

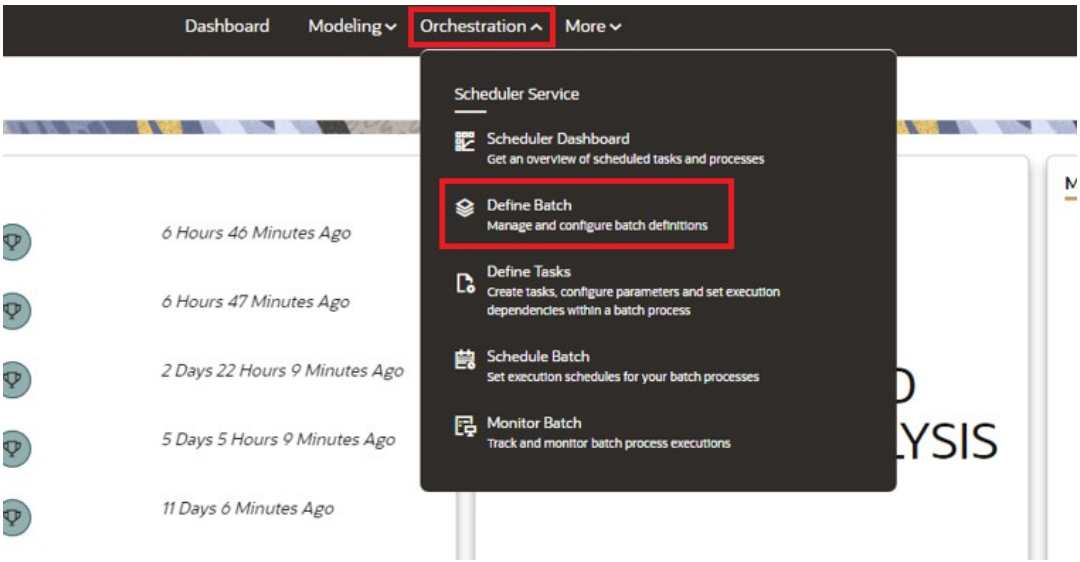
1. Launch **BD Production** workspace from the workspace summary screen.

Figure 5-84 Workspace



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

Figure 5-85 Scheduler Service

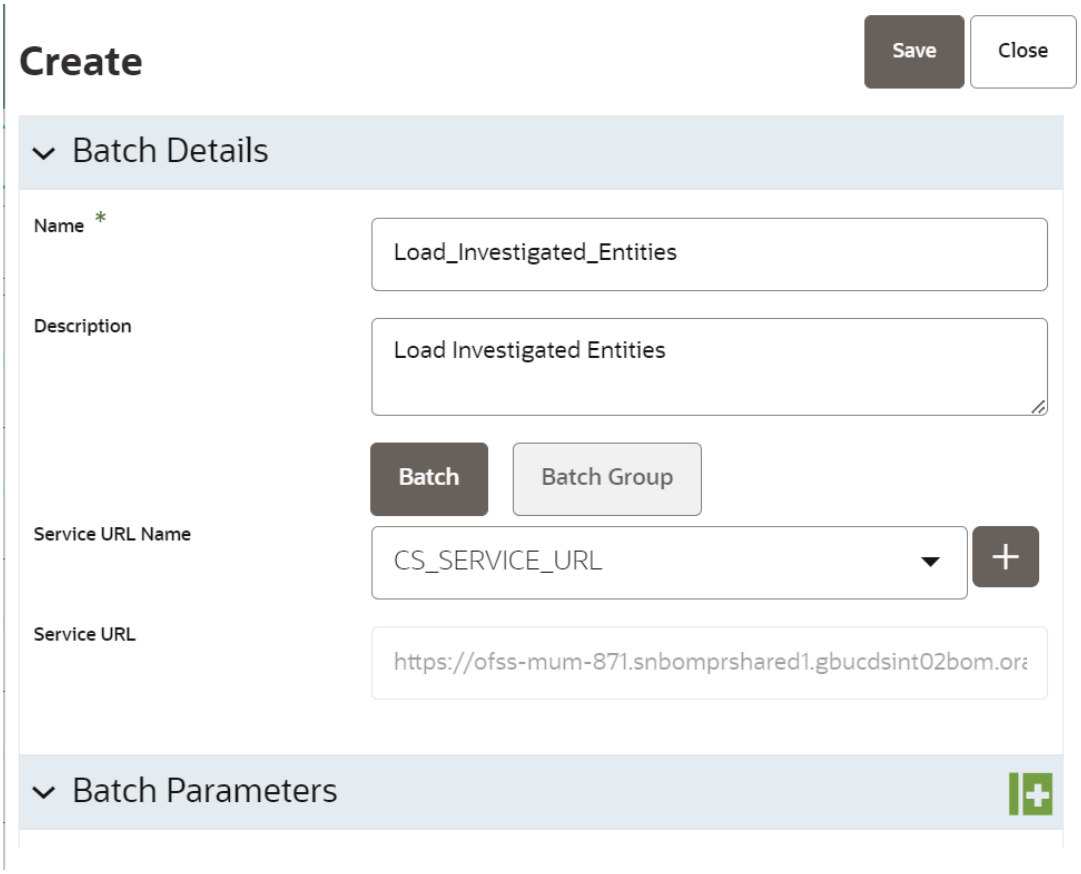


3. Click **Create** button on the top-right corner. The Create window is displayed.

Figure 5-86 Define Batch

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

Figure 5-87 Create Batch

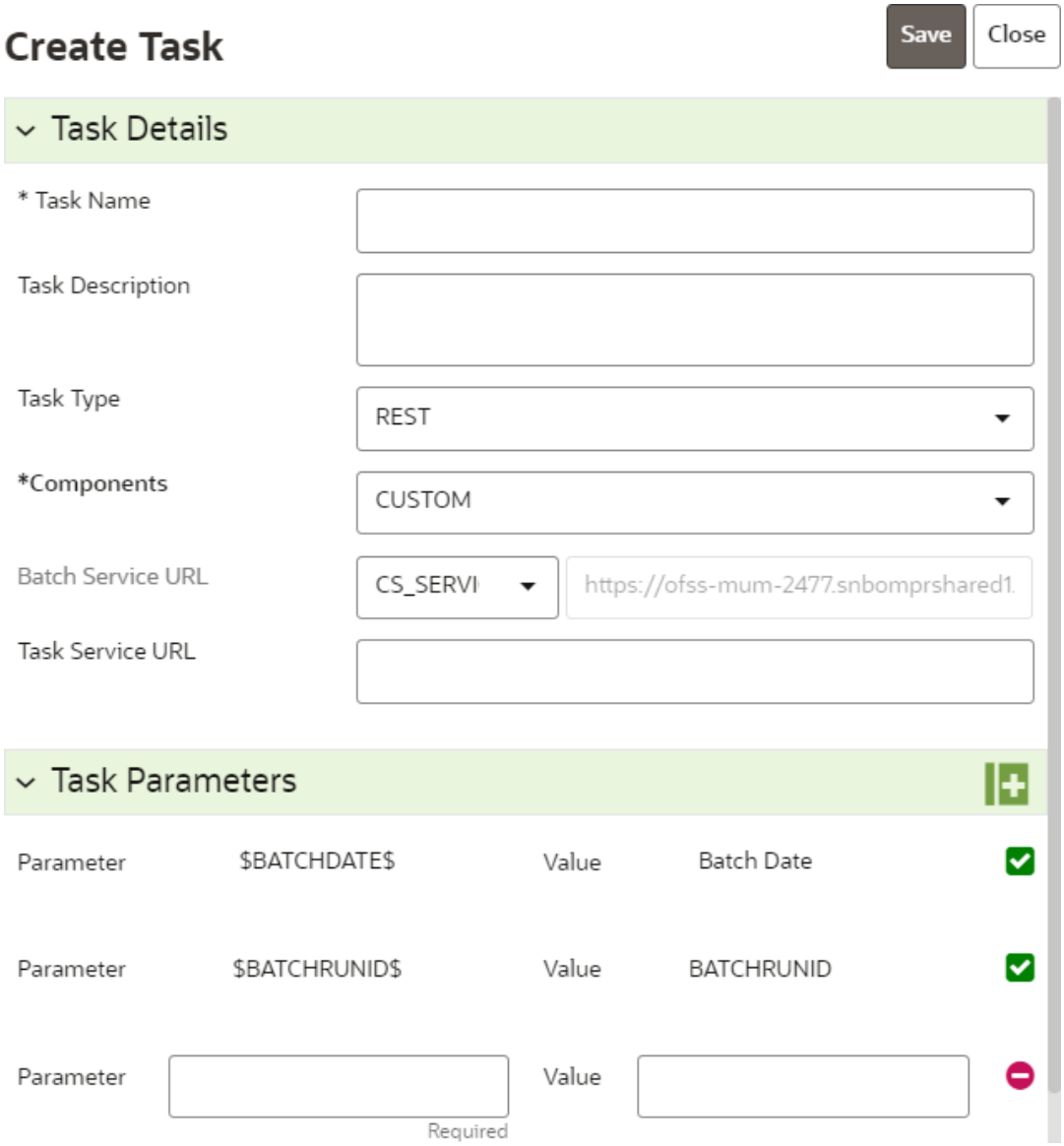


5. Click **Save** to **create a new batch**.
6. Navigate to **Scheduler Services** on the LHS pane and Click **Define Tasks** to create **New Task** in the newly created Batch.

Figure 5-88 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 5-89 Create Task



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

5.6.6 Data Movement

The Customer Risk Scoring use case make use of time series data and time series length can go up to six months to one year. When model is freshly deployed to production, given production may not have enough history of time series data; hence data movement from sandbox to production is required.

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
```

5.6.6.1 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.

Outputs
`AIF_DATA.dmp` will be created as part of successful execution.

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) `select * from dba_directories where directory_name = 'DATA_PUMP_DIR'`

5.6.6.2 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.

Outputs
On successful execution, AIF_BEHAVIORAL_DATA & AIF_NON_BEHAVIORAL_DATA will be populated for the model group.

Execution Logs
`IMP_AIF_DATA.log` will be created as part of the execution in case of any issues.

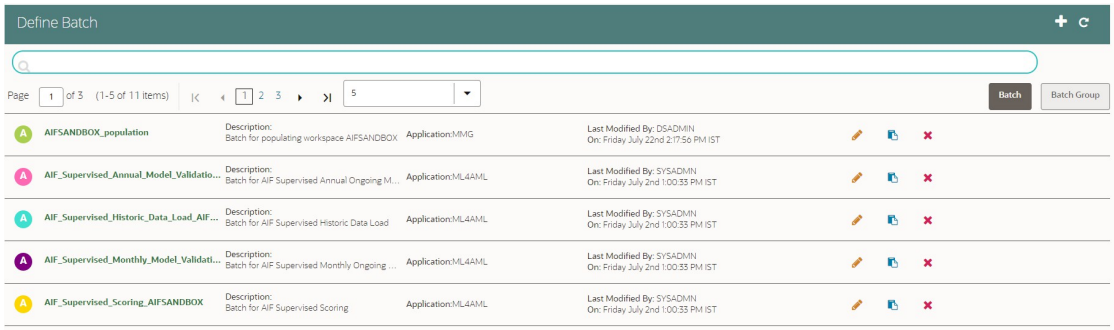
> **Note:**
>
> DATA_PUMP_DIR for any oracle database server can be found out using the following query (using sysdba)
>
> ```
> select * from dba_directories where directory_name = 'DATA_PUMP_DIR'
> ```

5.6.7 Batch Framework for Customer Risk Scoring

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 5-90 Define Batch

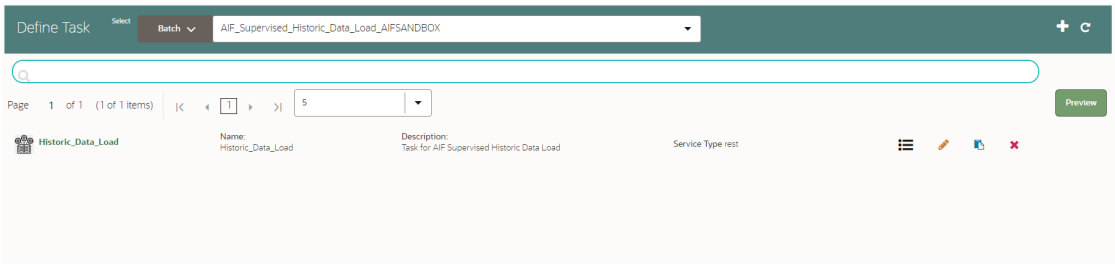


5.6.7.1 Supervised Historic Data Load

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.

Batch and Task Parameters
The batch contains a single task named **Historic_Data_Load**.

Figure 5-91 Task Details for Historic Data Load



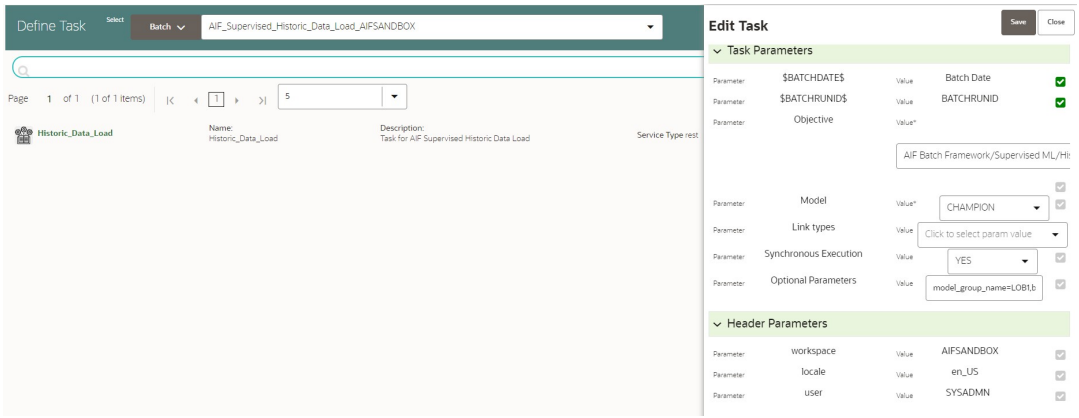
Task: Historic_Data_Load, Task Parameters

- 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 5-92 Edit Task Details for Historic Data Load



5.6.7.2 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.

Batch and Task Parameters
The batch contains the following tasks:

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

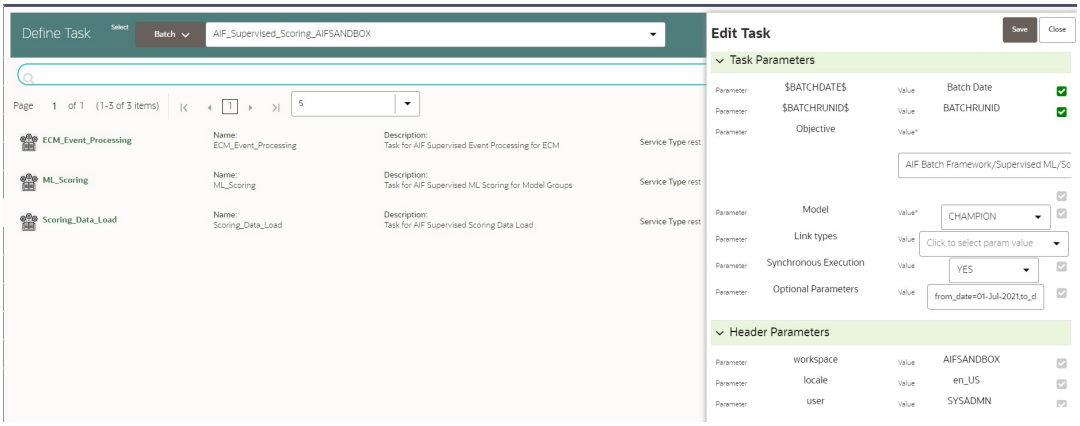
Task 1: Scoring_Data_Load, Task Parameters

- 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 5-93 Edit Task for Scoring Data Load



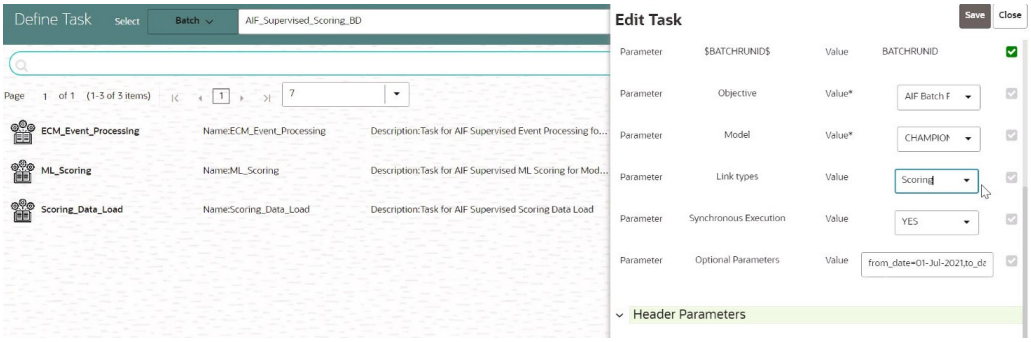
Task 2: ML Scoring, Task Parameters

- 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 5-94 Edit Task for ML Scoring



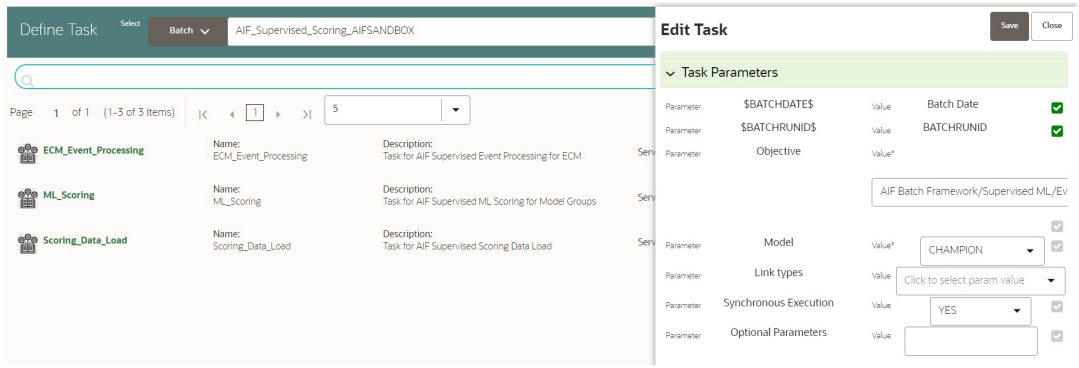
Task 3: ECM_Event_Processing, Task Parameters

- 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 5-95 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

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:

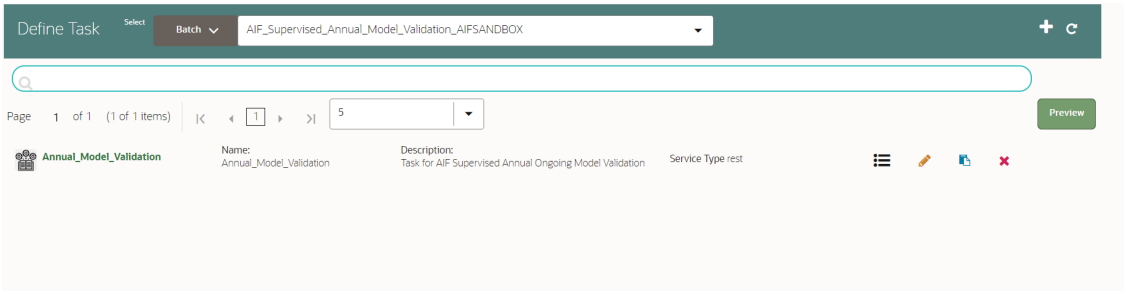
```
delete from aif_entity_score where d_fic_mis_date='DD-Mon-YYYY'
```

5.6.7.3 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.

Batch and Task Parameters
The batch contains a single task named Annual_Model_Validation

Figure 5-96 Define Task for Annual Model Validation

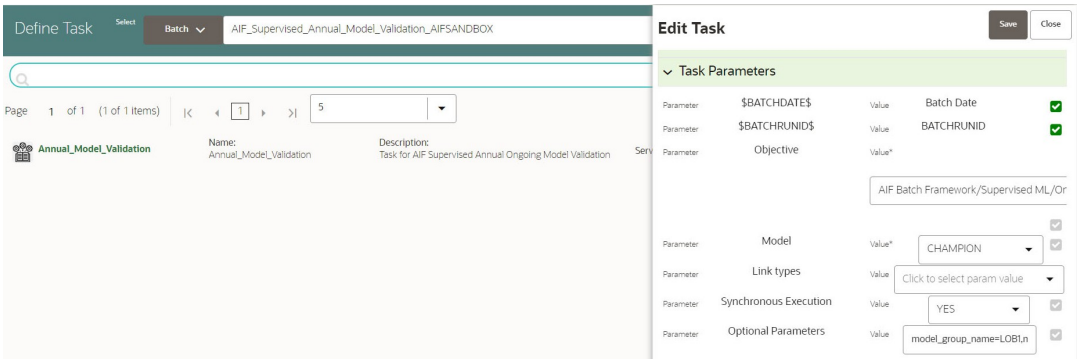


Task: Annual_Model_Validation, Task Parameters

- 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 5-97 Define Task for Annual Model Validation



5.6.7.4 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).

Batch and Task Parameters
The batch contains a single task named Monthly_Model_Validation.

Task: Monthly_Model_Validation, Task Parameters

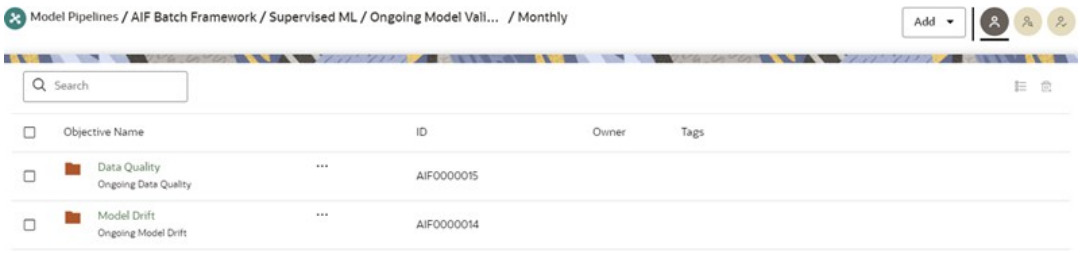
- Objective folder for Data Quality :

```
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
```

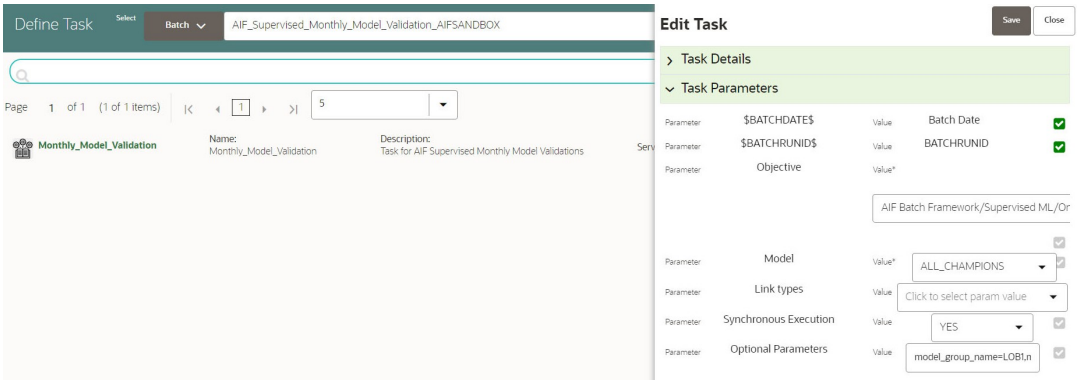
Figure 5-98 Monthly Validation



- 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 5-99 Define Task



5.6.8 Execute Batch

To execute the batch, see How to Execute Batch section.

5.6.9 Monitor Batch

To monitor the batch, see How to Monitor Batch section.

5.6.10 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.

5.7 Shell Account Detection Scenario for AML

This section explains about Shell Account Detection Scenario for AML use case.

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
4. Create a new data store for the sandbox schema
5. Register Conda Environment in BD Production Workspace

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

To register Conda Environment in BD Production Workspace, see How to Register Conda Environment in BD Production Workspace section.

5.7.1 Creating Sandbox Workspace

To create the sandbox workspace, see How to Create Sandbox Workspace section.

5.7.2 Populating Sandbox Workspace

To populate the sandbox workspace, see How to Populate the Sandbox Workspace section.

5.7.3 Importing Workspace Metadata

To import workspace metadata, follow these steps:

1. Login to Compliance Studio installed UNIX Machine.
2. Navigate to `<Compliance_Studio_HOME>/deployed/ml4aml/bin`.
3. Identify the utilities and execute commands as mentioned in the following table.

Table 5-9 Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh - w <workspace_wallet_alias>`
`importNotebooksScenar io.sh`	Yes	Yes	`./importNotebooksScenario.sh -w <workspace_code>`

5.7.4 Batch Framework for Shell Account Detection Scenario for AML

The **AML_Scenario_Processing** batch is available in the out of the box for the Typology scenario batch framework.

Figure 5-100 Define Batch for AML Scenario

5.7.4.1 AML Scenario Processing batch

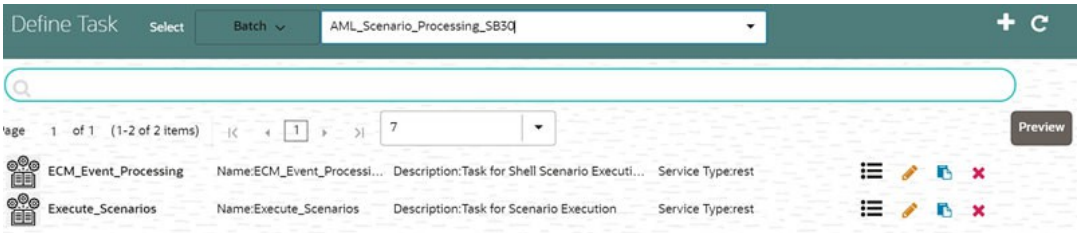
1. This is a pre-seeded batch and will be available in all workspaces (Production and Sandboxes).

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.

Batch and Task Parameters
The Batch contains the following task named as:

1. Execute_Scenario
2. ECM_Event_Processing

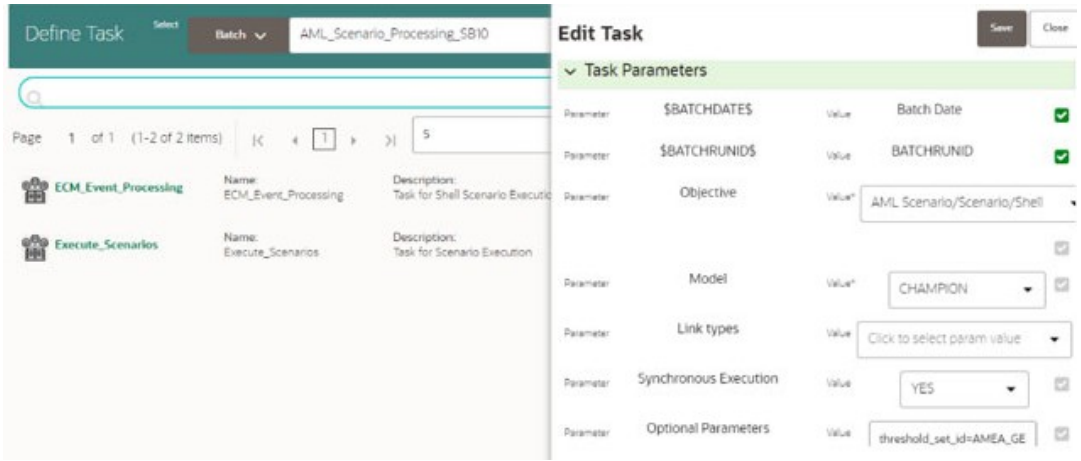
Figure 5-101 Define Task for AML Scenario



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**.

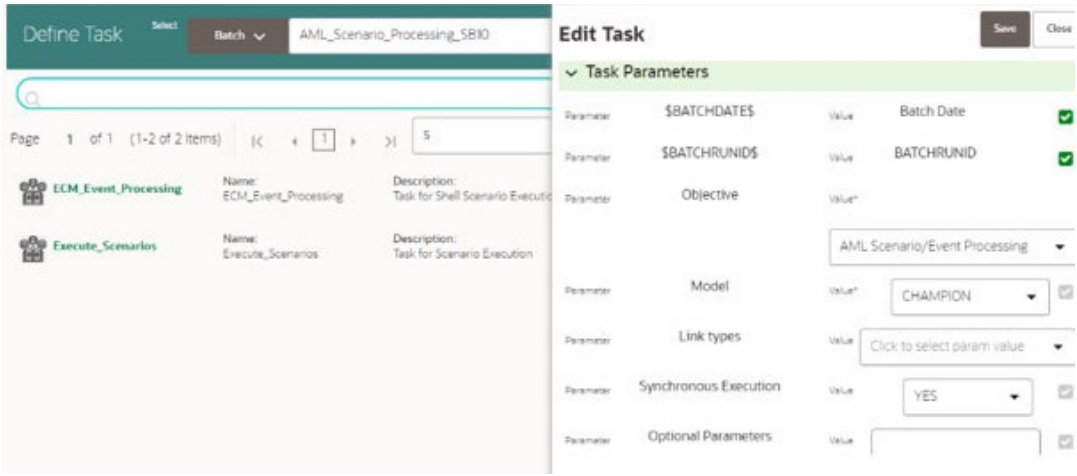
Figure 5-102 Define Task Parameter



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 5-103 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
```

5.7.5 ECM Connector Batch

Post Shell Account Detection 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.

5.8 Custom Scenario

This section explains about administration activity for Custom Scenario use case.

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
4. Create a new data store for the sandbox schema
5. Register Conda Environment in the BD Production Workspace

To create tablespace, sandbox schema and assign grants to sandbox schema, see the OFS Compliance Studio Installation Guide.

To create the data store, see How to Create Data Store section.

To register Conda Environment in BD Production Workspace, see How to Register Conda Environment in BD Production Workspace section.

Creating Sandbox Workspace

To create the sandbox workspace, see How to Create Sandbox Workspace section.

Populating Sandbox Workspace

To populate the sandbox workspace, see How to Populate the Sandbox Workspace section.

Importing Workspace Metadata

To import workspace metadata:

1. Login to Compliance Studio where UNIX machine is installed.
2. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ml4aml/bin` directory.
3. Identify utilities and execute the commands as mentioned in the following table.

Table 5-10 Utilities for Workspace and Notebook

Utility	Sandbox Workspace	Production Workspace	Command
`importWorkspaceSQL.sh`	Yes	Yes	`./importWorkspaceSQL.sh -w <workspace_wallet_alias>`
`importNotebooksCS.sh`	Yes	Yes	`./importNotebookCS.sh -w <workspace_code>`

5.8.1 Batch Framework for Custom Scenario

The following batches are available in the out-of-the-box for the Custom Scenario framework:

- **Aggregate Base Features for Custom Scenario (CS_Aggregate_Base_Features)**
- **Event Generation (CS_Event_Generation)**

Figure 5-104 Define Batch for Custom Scenario

5.8.1.1 Aggregate Base Features for Custom Scenario

- This pre-seeded batch is available in both Production and Sandbox workspaces.

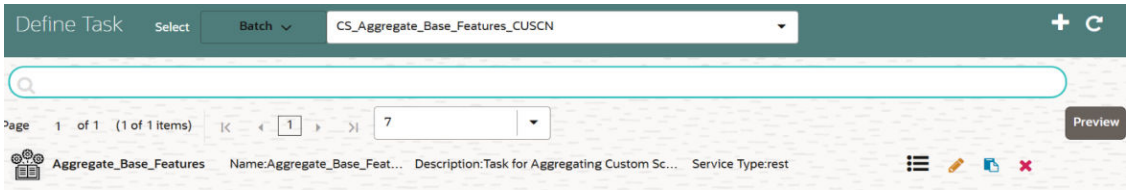
> **Note:**
>
> This batch has to be executed in the **Sandbox** workspace.

- This batch creates base features for custom scenario model training in the sandbox workspace.

Batch and Task Parameters

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

Figure 5-105 Define Task for CS_Aggregate_Base_Features



Task Parameters for Aggregate_Base_Features

- Objective Folder for this task:
  `Home / Modeling / Pipelines / ML4AML / Custom Scenario / Batch / Base Features`

> **Note:**
>
> You should 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-MMM-YYYY** format.
  - **to_date**: End Date for Historic Data lookup in **DD-MMM-YYYY** format.

- **prod_flag**: This flag indicates the Training/Scoring scenario. The option is either **Y** or **N**. For sandbox/historic training scenarios, the **prod_flag** should be set to **N**.
- **include_full_lookback**: This flag indicates whether lookback should consider data beyond the from_date to aggregating base features. The option is either **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-MMM-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 provided in the Admin notebook dataframe while creating the model group.

> **Note:**
>
> Custom Scenario supports **Customer** entity only.

- **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

Table 5-11 Task Parameters for Custom Scenario 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.

Table 5-11 (Cont.) Task Parameters for Custom Scenario Aggregate Base Features

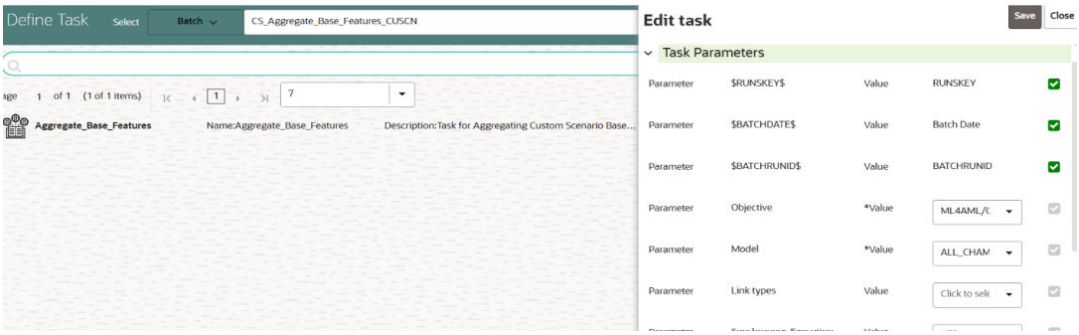
Parameter	Description
INCL_RLTD_PARTIES	It allows coverage of all transactions between related parties. The values are: * **Y**: Covers all transactions. * **N**: Excludes transactions between related parties.
RPTNG_CURR_FL	The value is Y or N. If Y, then all aggregation is to be done on reporting currency.
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**.
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~I
NCL_RLTD_PARTIES:Y~RPTNG_CURR_FL:N~MIN_HRG_RISK_LVL:10~INCL_SEC_PARTY_FL:Y~E
FFCTV_RISK_CUTOFF_LVL:10~ACTVTY_RISK_CUTOFF_LVL:10~INCLD_ACCT_HLDR_TYP_CD:C R~MANTAS_BUSINESS_ACCT_TYPES:RBK|
RBR~FUNC_CURR_FL:Y~INCL_WIRE_TRXN_PRDCT_ TYPE_LST:EFT-ACH|EFT-TREASURY|EFT-FEDWIRE|EFT-SWIFT|EFTOTHER|
EST~INCL_MI_TRXN_PRDCT_TYPE_LST:CASH-EQ-CASHIER-CHECK|CASH-EQ-CERTCHECK| CASH-EQ-MONEY-ORDER|CASH-EQ-TRAVELERS-CHECK|CASH-EQ-OTHER|CASHLETTER| CHECK|PAPER-OTHER|CHECK-ACH~INCL_CASH_TRXN_PRDCT_TYPE_LST:DEBITCARD| SVC|CREDITCARD| CURRENCY|
PHYS~INCL_BO_TRXN_PRDCT_TYPE_LST:JOURNAL~LRF_DIGITS:4~MIN_TRANS_ROUND_AMT:10~MAX_TRANS_ROUND_AMT:100000000~MIN_INDIVIDUAL_TRANS_A MT:10~DEGREE_ OF_PARALLELISM:8

- **Edit** Task Parameters and **Save**.

Figure 5-106 Edit Task for Aggregate Base Features



5.8.1.2 Event Generation for Custom Scenario

This pre-seeded batch is available in both Production and Sandbox workspaces.

> **Note:**
>
> This batch has to be executed in the **Production** workspace.

Batch and Task Parameters

The batch contains the following tasks:

- **Task 1: Aggregate_Scoring_Base_Features**
- **Task 2: Event_Processing**

Figure 5-107 Define Task for Event Generation

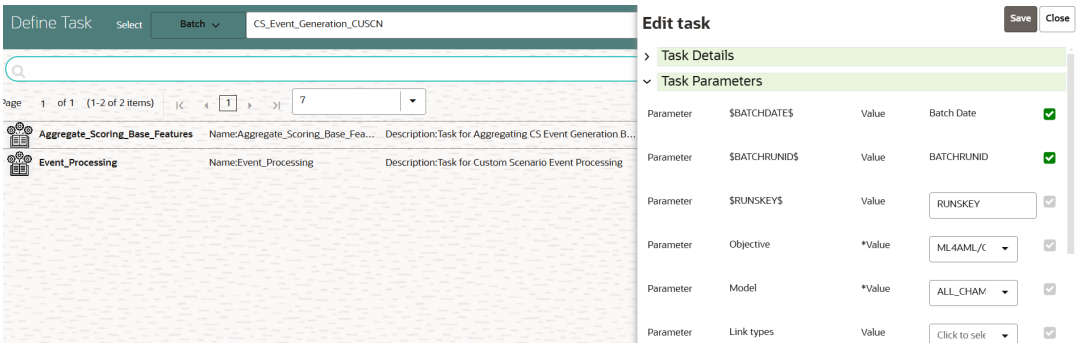
Task 1: Task Parameters for Aggregate_Scoring_Base_Features

- Objective folder for this task:
  `Home / Modeling / Pipelines / ML4AML / Custom Scenario / Batch / Base Features`

> **Note:**
>
> You should not change any parameter except **Optional Parameters**.

- Optional Parameters:
  - **prod_flag**: This flag indicates Training/Scoring scenario. The option is either **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. For example, **LOB1**.
  - **model_name**: Name of the Model used while importing model template using Admin Notebook. For example, **RMF**.
  - **focus**: The model entity name provided in the Admin notebook dataframe while creating the model group. The option is either **CUSTOMER** or **ACCOUNT**.
    For example:
    prod_flag=Y,model_group_name=GROUP1,model_name=M1,focus=CUSTOMER.
- **Edit** Task Parameters and **Save**.

Figure 5-108 Edit Task for Aggregate_Scoring_Base_Features for Custom Scenario



Task 2: Task Parameters for Event_Processing

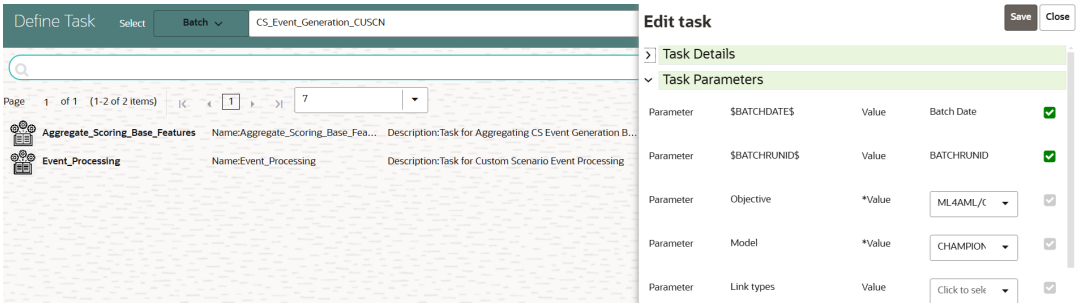
- Objective folder for this task:
  `Home / Modeling / Pipelines / ML4AML / Custom Scenario / Models / Event Processing`

> **Note:**
>
> You should not change any parameter except **Optional Parameters**.

- Optional Parameters:
  - **prod_flag**: This flag indicates Training/Scoring scenario. The option is either **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. For example, **RMF**.
  - **focus**: The model entity name provided in the Admin notebook dataframe while creating the model group. The option is either **CUSTOMER** or **ACCOUNT**.
    For example: prod_flag=y, model_group_name=GROUP1,model_name=M1,focus=CUSTOMER

    .
- **Edit** Task Parameters and **Save**.

Figure 5-109 Edit Task for Event_Processing of Custom Scenario



> **Note:**
>
> Once the batch execution is successful, the results are available in the **CS_EVENT_SCORE** and **CS_EVENT_SCORE_DETAILS** tables. For more information on these table structure, see OFS Compliance Studio Data Model Reference Guide.

5.8.2 Execute Batch

To execute the batch, see How to Execute Batch section.

5.8.3 Monitor Batch

To monitor the batch, see How to Monitor Batch section.

6
Restart Services

Use this section to understand how to stop or start the Compliance Studio service if you have an issue with the services.

6.1 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. Execute 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`

6.2 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. Execute 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.

A
Appendix

This section describes supplementary material, including detailed explanations, additional data, or technical information, that supports and enhances the main content without disrupting its flow.

A.1 Create Metadata Indexes using Logstash

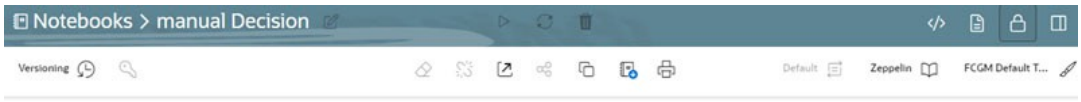
To create metadata indexes using Logstash, perform the following:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/load-to-opensearch/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.

A.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 A-1 Manual Decision notebook



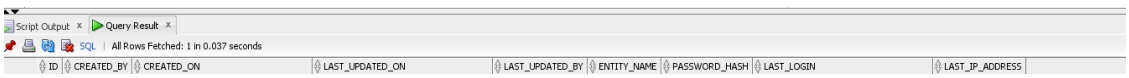
4. Click the **Write** Paragraphs icon at the top-right corner to unlock the notebook.

A.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 A-2 Output Table



You can check the LAST_IP_ADDRESS column, which will contain the IP address from where the user has last logged in.

A.4 Roles, Functions and Permissions

This section explains about Roles, Functions and Permissions.

A.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.

RoleCode	RoleName	Description
WKSPACC	Workspace Access	WorkspaceAccess Role
WKSPAUTH	Workspace Authorize	Workspace Authorize Role
WKSPREAD	Workspace Read	WorkspaceRead Role
WKSPWRITE	Workspace Write	WorkspaceWrite Role
FLDRACC	FolderAccess	FolderAccess Role
FLDRAUTH	FolderAuthorize	FolderAuthorize Role
FLDRREAD	FolderRead	FolderRead Role
FLDRWRITE	FolderWrite	FolderWrite Role
IDMGMTACC	IdentityMGMT access	Systemadmin access
IDMGMTADVN	IdentityMGMT advanced	Identitymanagement advanced
IDMGMTAUTH	IdentityMGMT authorize	Identitymanagement authorize
IDMGMTREAD	IdentityMGMT read	Identitymanagement read
IDMGMTWRIT	IdentityMGMT write	Identitymanagement write
FUNC_READ	FunctionRead Role	-
FUNC_WRITE	FunctionWrite Role	-
FUNC_ADV	FunctionAdvanced Role	-
ROLE_READ	RoleRead Role	-
ROLE_WRITE	RoleWrite Role	-
ROLE_ADV	RoleAdvanced Role	-
ROLE_AUTH	RoleAuthorize Role	-
GRP_READ	GroupRead Role	-
GRP_WRITE	GroupWrite Role	-
GRP_ADV	GroupAdvanced Role	-
GRP_AUTH	GroupAuthorize Role	-
USR_READ	UserRead Role	-
USR_WRITE	UserWrite Role	-
USR_ADV	UserAdvanced Role	-
USR_AUTH	UserAuthorize Role	-
SRVC_READ	ServiceRead Role	-
APP_READ	ApplicationRead Role	-
WRKSP_READ	WorkspaceRead Role	-
WRKSP_WRITE	WorkspaceWrite Role	-
WRKSP_ADV	WorkspaceAdvanced Role	-
FLDR_READ	FolderRead Role	-
FLDR_WRITE	FolderWrite Role	-
FLDR_ADV	FolderAdvanced Role	-

RoleCode	RoleName	Description
DTSRC_READ	DataStoreRead Role	-
ADMIN_LINK	AdminLink Role	-
BATCH_READ	BatchRead Role	Batchread role in scheduler service
BATCH_WRITE	BatchWrite Role	Batchwrite role in scheduler service
BATCH_ADV	BatchAdvance Role	Batchadvance role in scheduler service
BATCH_AUTH	BatchAuthorization Role	Batchauthorize role in scheduler service
BATCH_OPER	BathOperation Role	Batchoperation role in scheduler service
BATCH_MAINT	BatchMaintenance Role	Batchmaintenance role in scheduler service
MDLACCESS	ModelAccess	UserGroup mapped will have access to Model Link and Summary
MDLREAD	Model Read	Model Read
MDLWRITE	ModelWrite	ModelWrite
MDLPHANTOM	ModelPhantom	ModelPhantom
MDLAUTH	ModelAuthorize	ModelAuthorize
MDLADV	ModelAdvanced	ModelAdvanced
MDLREVIEW	ModelReview	ModelReview
MDLDEPLOY	ModelDeployment	ModelDeployment
MDLADMIN	ModelAdmin	ModelAdmin
DSREAD	DataStoreRead	DataStoreRead
DSWRITE	DataStoreWrite	DataStoreWrite
DSACCESS	DataStoreAccess	DataStoreAccess
DSADMIN	DSADMIN	ComplianceStudio Admin Role
DSBATCH	DSBATCH	Batch Role
DSINTER	DSINTER	ComplianceStudio Interpreter Configuration Role
DSUSER	DSUSER	ComplianceStudio User Role
DSAPPROVER	DSAPPROVER	ManualEdges Approver role
DSREDACT	DSREDACT	Redactionrole for Graph
MDLEXE	ModelExecute	ModelExecute
MDAPPROVER	MDAPPROVER	Approver
MDREQUESTER	MDREQUESTER	Requester

A.4.2 Default Roles Seeded in Notebook Server through permissions-int.yml file

Table A-1 Default Roles

Name	Description
DSADMIN	Admin Role (all permissions)

Table A-1 (Cont.) Default Roles

Name	Description
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

A.4.3 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.

Table A-2 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
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

Table A-2 (Cont.) Compliance Studio Functions

Function Code	Function Name	Description
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
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	-

Table A-2 (Cont.) Compliance Studio Functions

Function Code	Function Name	Description
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	-
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

Table A-2 (Cont.) Compliance Studio Functions

Function Code	Function Name	Description
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
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

Table A-2 (Cont.) Compliance Studio Functions

Function Code	Function Name	Description
DSDELETE	DataStore Delete	The user mapped to this function can delete DataStore
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

A.4.4 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.

Table A-3 Notebook Server Permissions

Name	Description
create_notebook	Create a notebook
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
interpreter_variant_delete	Delete an interpreter variant
interpreter_variant_view	View an interpreter variant
job_cancel	Cancel a job

Table A-3 (Cont.) Notebook Server Permissions

Name	Description
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
delete_role	Delete a role
update_group	Update a group
update_permission_template	Update a permission template
update_role	Update a role

Table A-3 (Cont.) Notebook Server Permissions

Name	Description
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)	User scan approve the manual similarity edge
review_request(deprecated)	User scan request for approving manual similarity edge
Approve	User scan approve scenario notebook

A.4.5 Group - Role Mapping

Table A-4 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
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

Table A-4 (Cont.) Role Mapping

Group Code	Group Name	Role Code	Role Name
		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
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

A.4.6 Role - Function Mapping

Table A-5 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
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

Table A-5 (Cont.) Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
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
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

Table A-5 (Cont.) Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
		GRP_SUMM	Group Summary
		GRP_VIEW	Group View
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
IDMGMTWRIT	Identity MGMT write	ADMINSCR	Administration Screen
		ROLEMAINT	Role Maintenance Screen
		UGFLROLMAP	User Group Folder Role Map Screen
		UGMAINT	User Group Maintenance Screen

Table A-5 (Cont.) Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
		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
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
SRVC_READ	Service Read Role	SRVC_SUMM	Service Summary
		SRVC_VIEW	Service View

Table A-5 (Cont.) Role - Function Mapping

Role Code	Role Name	Function Code	Function Name
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_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

A.4.7 Role - Permission Mapping

> **Note:**
>
> The role **DSREDACTGRP** is used for applying redaction in the graph.

Table A-6 Role - Permission Mapping

Permissions	DSADMIN	DSBATCH	DSINTER	DSUSER	DSAPPRROVER	MDAPPROVER	MDREQUESTOR
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	-	-	-	-

Table A-6 (Cont.) Role - Permission Mapping

Permissions	DSADMIN	DSBATCH	DSINTER	DSUSER	DSAPPRROVER	MDAPPROVER	MDREQUESTOR
interpreter_variant_view	Yes	Yes	Yes	Yes	-	-	-
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	-	-	-

Table A-6 (Cont.) Role - Permission Mapping

Permissions	DSADMIN	DSBATCH	DSINTER	DSUSER	DSAPPRROVER	MDAPPROVER	MDREQUESTOR
toggle_show_result	Yes	Yes	Yes	Yes	-	-	-
Update	Yes	Yes	Yes	Yes	-	-	-
View	Yes	Yes	Yes	Yes	-	-	-
view_code	Yes	Yes	Yes	Yes	-	-	-
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	-	-	-

Table A-6 (Cont.) Role - Permission Mapping

Permissions	DSADMIN	DSBATCH	DSINTER	DSUSER	DSAPPRROVER	MDAPPROVER	MDREQUESTOR
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

A.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-black:

   ```
   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
   ;;
   ```

A.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_RUNSKEY>"
   "<EXECUTION_MODE>" "<ER_SCHEMA_WALLET_ALIAS>" "<BATCH_GROUP>"
   "<PIPELINE_ID>" &
   ```

   For example, 8128 version:

   ```
   nohup ./ER_Cleanup.sh "CLEANUP-JOB1-INSTANCE"
   "20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8128" &
   ```

A.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_RUNSKEY>"
"<EXECUTION_MODE>" "<ER_SCHEMA_WALLET_ALIAS>" "<BATCH_GROUP>"
"<PIPELINE_ID>" &
```

For example, 8128 version:

```
nohup ./ER_Cleanup.sh "CLEANUP-JOB2-INSTANCE"
"20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8128" &
```

A.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_RUNSKEY>"
"<EXECUTION_MODE>" "<ER_SCHEMA_WALLET_ALIAS>" "<BATCH_GROUP>"
"<PIPELINE_ID>" &
```

For example, 8128 version:

```
nohup ./ER_Cleanup.sh "CLEANUP-JOB3-INSTANCE"
"20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8128" &
```

A.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_RUNSKEY>"
"<EXECUTION_MODE>" "<ER_SCHEMA_WALLET_ALIAS>" "<BATCH_GROUP>"
"<PIPELINE_ID> &
```

For example, 8128 version:

```
nohup ./ER_Cleanup.sh "CLEANUP-JOB4-INSTANCE"
"20150110" "148" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8128" &
```

A.10 Resetting Entity Resolution Back to Day 0

> **Note:**
>
> - This section is applicable only when you wipe out ER-related tables and indexes. This will bring the Entity Resolution back to **Day0**.
> - 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_RUNSKEY>"
"<EXECUTION_MODE>" "<ER_SCHEMA_WALLET_ALIAS>" "<BATCH_GROUP>"
"<PIPELINEID>" &
```

For example, 8128 version: `nohup ./ER_Cleanup.sh "RESET-TO-DAY0" "20151210" "182" "RUN" "ER_SCHEMA_PP_ALIAS" "CSA_812" "CSA_8128" &`

A.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;
```

A.11 Utility Scripts

This section describes about Utility Scripts.

A.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.

> **Note:**
>
> 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. For example, for any **V_PARTY_ID**, there should be the same **FIC_MIS_DATE** tagged in each input table.

A.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.

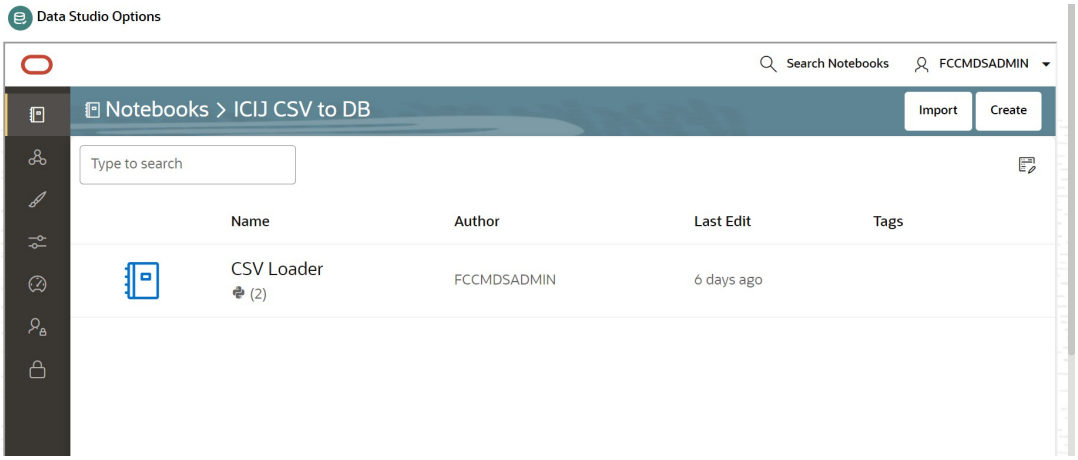
- ICIJ_NODES_ENTITY
- ICIJ_NODES_INTERMEDIARY
- ICIJ_NODES_OFFICER
- ICIJ_NODES_OTHERS

- 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 A-3 ICIJ Notebook



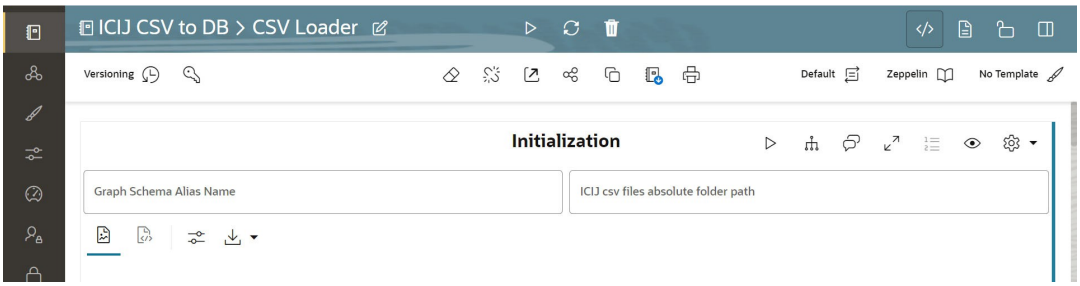
4. Open the Notebook, **ICIJ CSV to DB/CSV Loader.**
5. Click **Export Notebook** to download the notebook.
   The notebook is saved in the local machine.
6. Navigate to the **Modeling** drop-down list and select **Pipelines**.
7. Click **Add** and select **Objective** from the list to display the **Objective 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**.

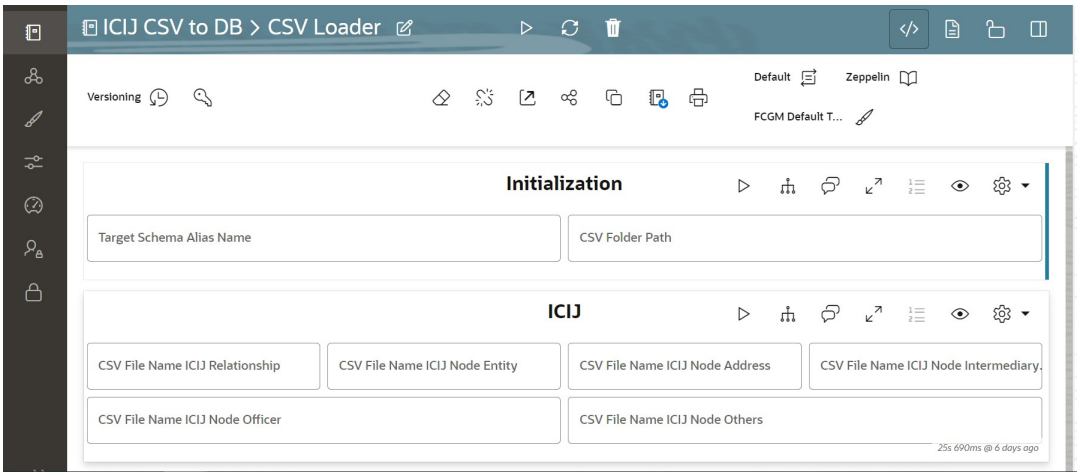
14. Enter the details for the **Draft Name** and **Description**.
15. Enter a tag in the **Tags** field.
16. Click **Import**. A new model is created by importing the model data dump.
    For more information on importing workspace models, see the **Import a Workspace Model Data into a New Model** section 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 the **Run** icon to run the paragraph.

Figure A-4 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 A-5 CSV Files Details



20. Click the **Run** icon 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.

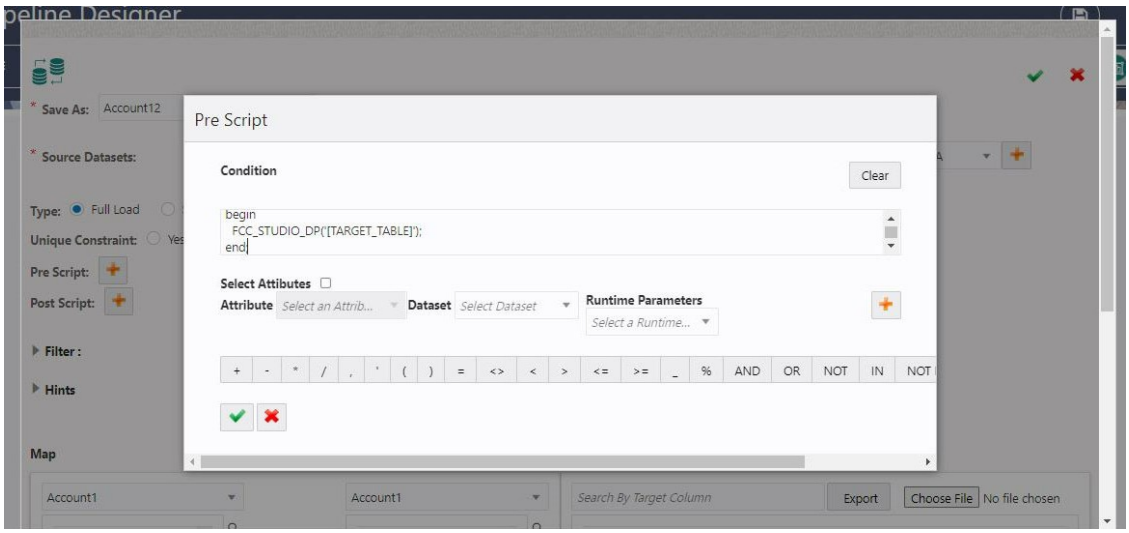
A.13 Prescript Condition

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 A-6 Prescript condition



For more details on the Data pipeline, see Managing Data Pipeline section in the OFS Compliance Studio User Guide.

A.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.

> **Note:**
>
> - User should provide "graph_id" value in lowercase when running the following cleanup scripts:
>
>   ```
>   GraphPipeline_cleanup_day0_in_studioschema.sql
>   GraphPipeline_cleanup_day0_in_graphschema.sql
>   ```
>
> - The graph_id value can also be fetched in the MMG_GRAPH_SCHEMA table from the Studio Schema.

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.

A.15 Disable 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 the **Revoke** Button.
5. Click **Save** to modify the changes.

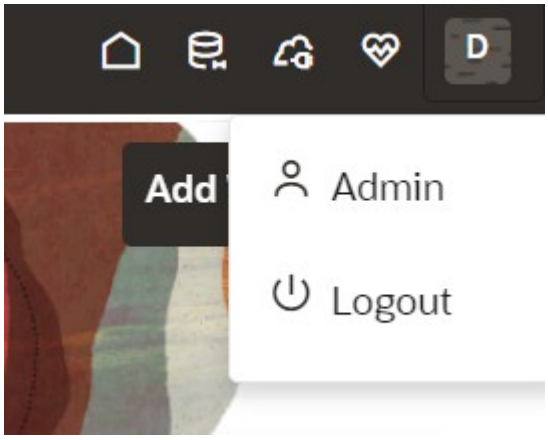
In Compliance Studio,

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 A-7 Identity Management



3. Select the same user of the Groups that are removed from the IDCS.
4. Navigate to the **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 the **Mapped Groups** tab and move the toggle switch to the right to enable **Authorization View**.
10. Select all the groups and click the **Authorize** button.
11. Restart the Compliance Studio.

A.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>/loadto-
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

A.17 Parameters for Entity Resolution Job execution

This section describes parameters for job execution and cleanup for Entity Resolution.

Table A-7 Parameter for Entity Resolution

Parameter	Description	ER Job Execution	Cleanup
Pipeline ID	ER Type has taken as Pipelined ID to execute. For example, CSA_8128.	Yes	Yes
ErSchemaID	The identifier of the schema on which Entity Resolution has to be run.	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
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

Table A-7 (Cont.) Parameter for Entity Resolution

Parameter	Description	ER Job Execution	Cleanup
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 A-7 (Cont.) 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

A.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 following table.

Table A-8 Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Admin.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
AMLES Admin Notebook.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table A-8 (Cont.) Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
ATL Analysis.dsnb	Pre-configured with ml4aml_<CS version> **NOTE**: There is no explicit selection of conda environmenti s required for the pre-configured notebooks during execution.
Auto-MLOutput Tracking.dsnb	Pre-configured with ml4aml_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Auto-MLOutput Viewing Using REST.dsnb	Pre-configured with ml4aml_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
BTLAnalysis.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
ICIJCSV to DB_CSV Loader.dsnb	default_<CS version>
ML_Address_Matching_Training_Admin.dsnb	sane_<CS version>
ML_Address_Matching_Training_ETL.dsnb	sane_<CS version>
ML_Name_Matching_Training_Admin.dsnb	sane_<CS version>
MLNamematchingTrainingAdminPublish.dsnb	sane_<CS version>
ML_Name_Matching_Training_ETL.dsnb	sane_<CS version>
ML_Name_Matching_Training_ETLPublish.dsnb	sane_<CS version>
Outcome Analysis.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Sanctions EDQ Update.dsnb	Pre-configured with ml4aml_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table A-8 (Cont.) Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Sanctions Event Scoring User Notebook.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
Supervised ML Scoring Data Load.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.

Table A-8 (Cont.) Builtin Python Notebooks and its corresponding Conda Environment

Notebook	Conda Environment
Transaction Analysis.dsnb	Pre-configured with ml4aml_<CS version> **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_<CS version> **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_<CS version> **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_<CS version> **NOTE**: There is no explicit selection of conda environment is required for the pre-configured notebooks during execution.
ERDASHBOARD Data Analysis.dsnb	sane_<CS version>
ERDASHBOARD Match And Merge Analysis.dsnb	sane_<CS version>
Scenario_Conversion_Utility.dsnb	default_<CS version>
Scenario_Conversion_Utility_Verification_NB.ds nb	default_<CS version>
SCU_Set_Calendar.dsnb	default_<CS version>

A.19 Python Libraries for Predefined Conda Environment

Compliance Studio comes with predefined Conda environments as follows:

- **default_<CS version>**
- **ml4aml_<CS version>**
- **sane_<CS version>**

Table A-9 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

Table A-9 (Cont.) Default Conda Python Environment

Package	Version
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
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

Table A-9 (Cont.) Default Conda Python Environment

Package	Version
platformdirs	3.8.0
plotly	5.8.0
prompt-toolkit	3.0.38
protobuf	4.23.3
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
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

Table A-9 (Cont.) Default Conda Python Environment

Package	Version
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 A-10 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
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.

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 --findlinks
$FULL_PATH_INCLUDING_PYSPARK_PACKAGE_NAME
```

Table A-11 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
cycler	0.11.0
deprecation	2.1.0
ds-interpreter-client	23.4.2
fonttools	4.40.0
globalparty	8.1.2.7.0rc8
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

Table A-11 (Cont.) Sane Conda Environment

Package	Version
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
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

A.20 Implementation of Connection Pooling in PGX Realm

PGX Server creates a connection and is used to load data. Implementing a connection pool for performance improvement is recommended to save time when creating and closing connections.

To implement the connection pool, follow these steps:

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

Table A-12 Config.sh File

Parameter	Significance	Default Value / Example
PGX_ENABLE_CP	It is used to enable or disable connection pooling for sub graph loading. The value for 'PGX_ENABLE_CP' is "true" or "false".	For example, PGX_ENABLE_CP=true
PGX_CP_INITIAL_SIZE	Indicates the initial number of connections that are created when the pool is started.	For example, 5
PGX_CP_MAX_TOTAL	Indicates the maximum number of active connections that can be allocated from this pool at the same time or negative for no limit.	For example, 25

Table A-12 (Cont.) Config.sh File

Parameter	Significance	Default Value / Example
PGX_CP_MAX_IDLE	Indicates the maximum number of connections that can remain idle in the pool, without extra ones being released or negative for no limit.	For example, 10
PGX_CP_MIN_IDLE	Indicates the minimum number of connections that can remain idle in the pool, without extra ones being created, or zero to create none.	For example, 5
PGX_CP_MAX_WAIT_ MILLIS	Indicates the maximum number of milliseconds that the pool will wait (when there are no available connections)for a connection to be returned before throwing an exception or -1 to wait indefinitely.	Forexample, 3000
PGX_CP_MIN_EVICTABLE_IDLE	Indicates the minimum amount of timea connection may sit idle in the pool before it is closed and a new connection is created if count of connections is less than PGX_CP_MIN_IDLE.	Forexample, PT30M
PGX_CP_SOFT_MIN_EVICTABLE	Indicates the minimum amount of timea connection may sit idle in the pool before it is closed and a new connection is created. **NOTE:**Thevalue is lesser than PGX_CP_MIN_EVICTABLE_IDL E_TIME will close all the idle connectionand create connection to match PGX_CP_MIN_IDLE.	For example, PT8H. PT30M= 30 minutes PT55S = 55 seconds PT2H = 2 hours

While executing the Refresh Graph task, the connection pooling parameters can be overridden by the run time parameters.

To configure run time parameters, follow these steps:

1. On the **Orchestration** menu, click **Schedule Batch**.
2. Select the Out-of-the-box (BD/ECM) graph and click **Edit Dynamic Params**.
3. On the Refresh Graph, provide the following value.
   initialSize=5, maxTotal=15, maxIdle=10, minIdle=5, maxWaitMillis=3000, minEvictableIdleTime=PT30M, softMinEvictableIdleTime=PT8H
4. Run the **Refresh Graph** task.

> **Note:**
>
> If you have more than one PGX server for load balancer, restart PGX servers.

5. To update connection pooling details in the existing graph, execute the following script in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-load-tograph/graph-service/utility/bin` directory.

```
./SetConnectionPoolConfig.sh --username <#username#> --graph-id <#graphid
#> --initial-size <#initial-size#> --max-total <#max-total#> --maxidle
<#max-idle#> --min-idle<# min-idle#> --max-wait-millis <#max-waitmillis#>
--min-evict-idle-time <#min-evict-idle-time#> --soft-min-evictidle-
time <#soft-min-evict-idle-time#>
```

> **Note:**
>
> This step is applicable only for the existing graph pipeline.

For example,

```
./SetConnectionPoolConfig.sh --username fccuser --graph-id --
initial-size 25 --max-total 50 --max-idle 35 --min-idle 10 --max-waitmillis
3000 --min-evict-idle-time PT30M --soft-min-evict-idle-time PT8H
```

You can refer place holder details in the following table.

Table A-13 Description for Connection Pool Parameter

#Place Holder#	Description
[-u|--username]	Compliance Studio User
[-g|--graph-id]	The graph id for which pool-able connection details are to be set.
[-i|--initial-size]	The initial number of connections created when the pool is started.
[-t|--max-total]	The maximum number of active connections that can be allocated from this pool at the same time.
[-mi|--max-idle]	The maximum number of connections that can remain idle in the pool without extra ones being released.
[-li|--min-idle]	The minimum number of connections that can remain idle in the pool without extra ones being created.
[-w|--max-wait-millis]	The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception.
[-et|--min-evict-idle-time]	The minimum amount of time a connection may sit idle in the pool before it is closed and a new connection is created if the count of connections is less than the min-idle value. **NOTE:** The value should be in ISO-8601 format. Refer to examples on ISO-8601 format values.

Table A-13 (Cont.) Description for Connection Pool Parameter

#Place Holder#	Description
[-st|--soft-min-evict-idle-time]	The minimum amount of time a connection may sit idle in the pool before it is closed and a new connection is created. **NOTE:** The value should be in ISO-8601 format. Refer to examples on ISO-8601 format values.
[-h|--help]	For any help required to execute this script.

A.21 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.

A.21.1 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).

A.21.2 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.

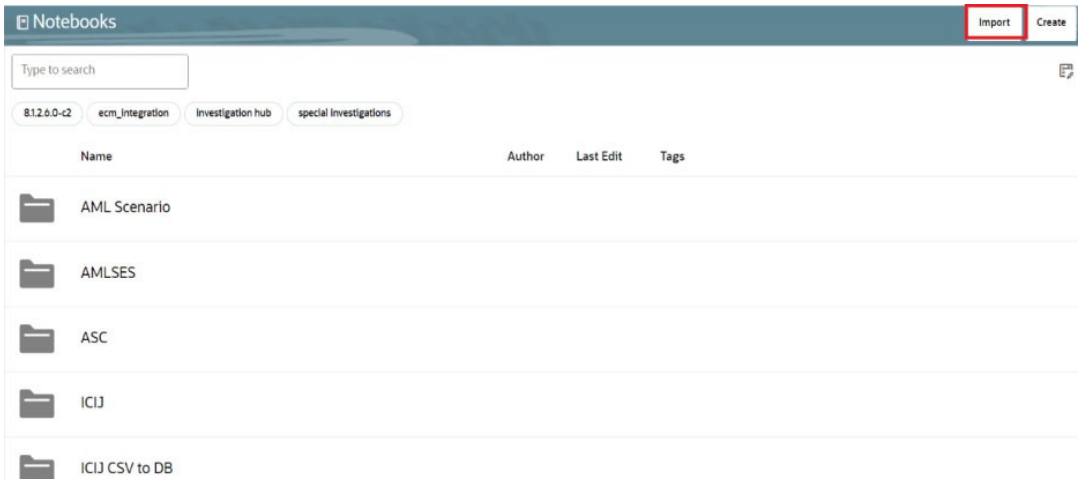
   ```
   https://<Host_Name>:<Port_Number>/cs
   ```

   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.

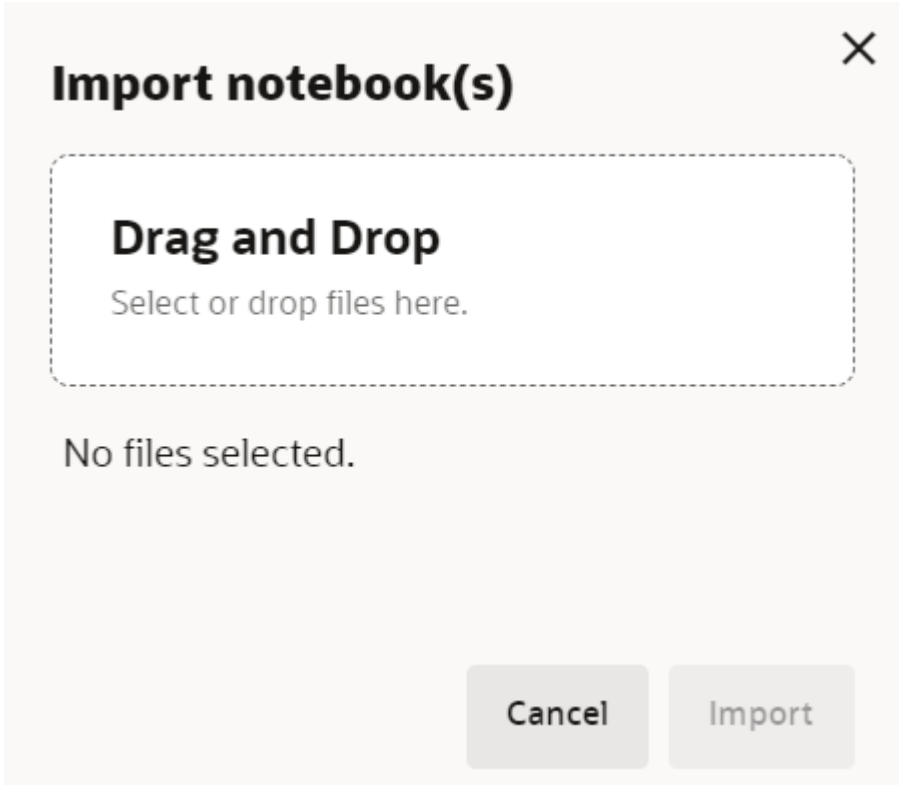
   Once logged in, the Notebooks page is displayed.

Figure A-8 Sample Notebooks



2. Click **Import**. The Import notebook(s) pane is displayed.

Figure A-9 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.

A.21.3 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.

A.21.4 Integrating Notebook with ECM

The notebook is integrated with ECM to enable Case Investigators to investigate cases in the ECM.

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.

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.

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 following table

Table A-14 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
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.

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"
```

A.22 How-To

This section provides a collection of How-To procedures.

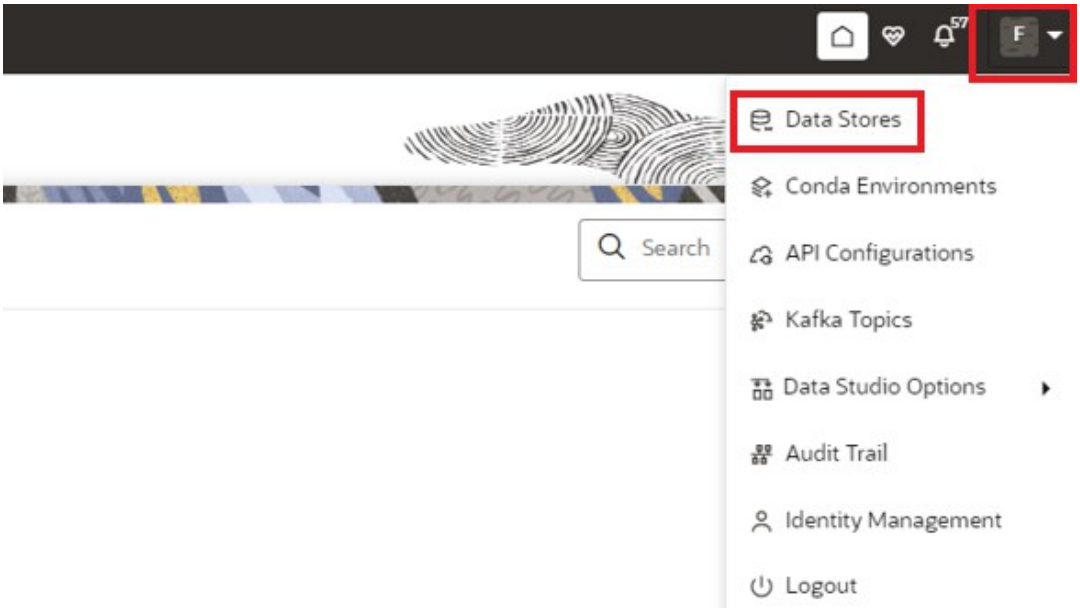
A.22.1 How to Create Data Store

Data store is metadata around the source connection details. Users can register Oracle/Hive data source connection details as data store with Compliance Studio. Data store is added to the workspace to point to a particular source connection to fetch the data.

To create a data store, follow these steps:

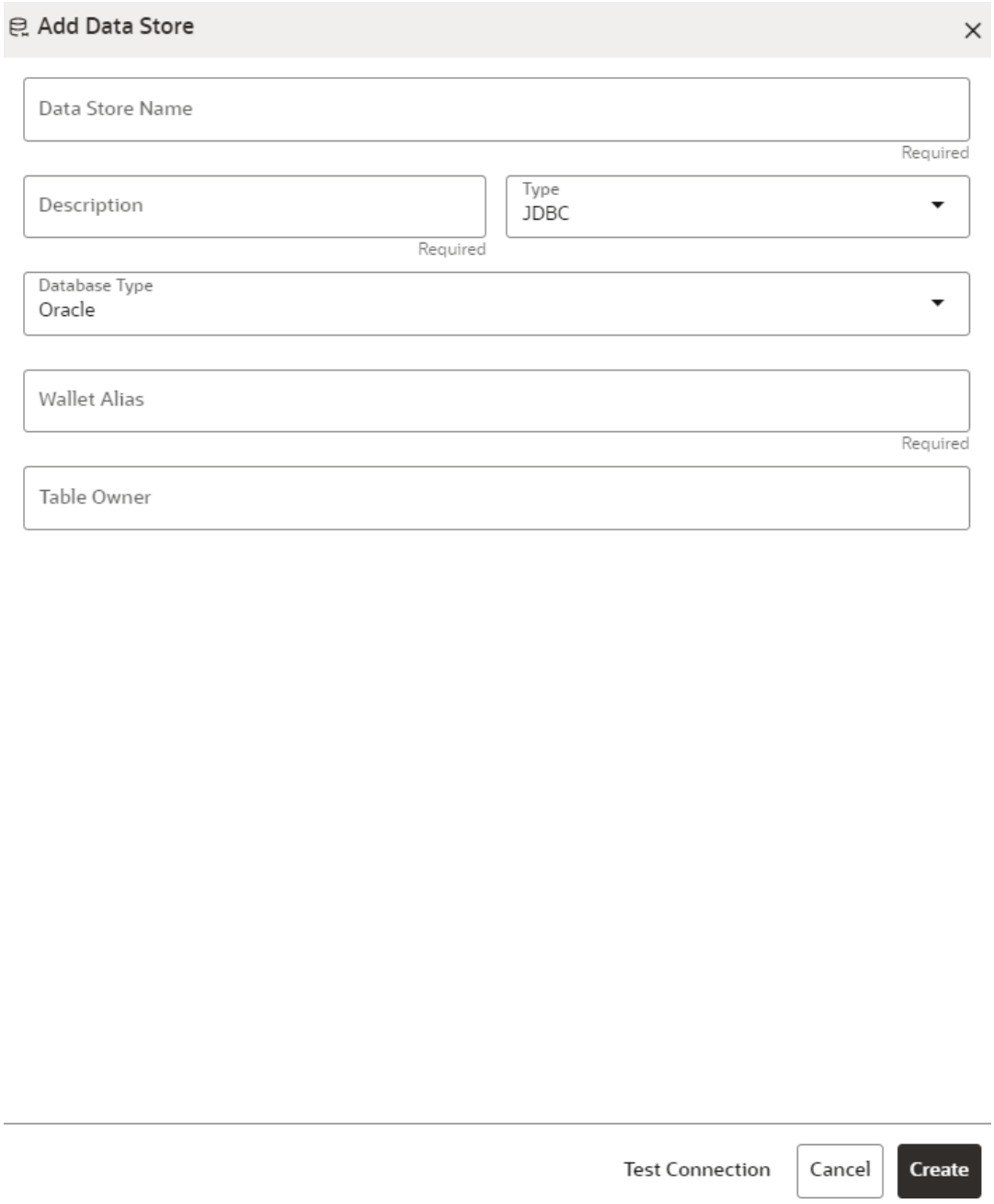
1. Navigate to **Workspace Summary** page.

Figure A-10 Workspace Summary



2. Click the **User Profile** drop-down list and select **Data Store**.
3. Click **Add Data Store**. The Add Data Store page is displayed.

Figure A-11 Add Data Store with Oracle Database



4. Enter the required details as describe in the following table.

Table A-15 Add Data Store

Field	Description
Data Store Name	Enter the connection URL to the database for the data schema.
Description	Enter the description of database connection.
Type	From the Type drop-down list, select the **JDBC**
Database Type	From the Database Type drop-down list, select the **Oracle**.

Table A-15 (Cont.) Add Data Store

Field	Description
Wallet Alias	Enter the Wallet Alias. This value should be same as configured using Oracle Wallet.
Table Owner	Enter the Oracle Database schema name.

5. Click **Create** to create/add a new data store for the sandbox workspace.

A.22.2 How to Register Conda Environment in BD Production Workspace

To register conda environment in the installer, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin` directory.
2. Execute the following command:

```
./compliance-studio.sh -e
```

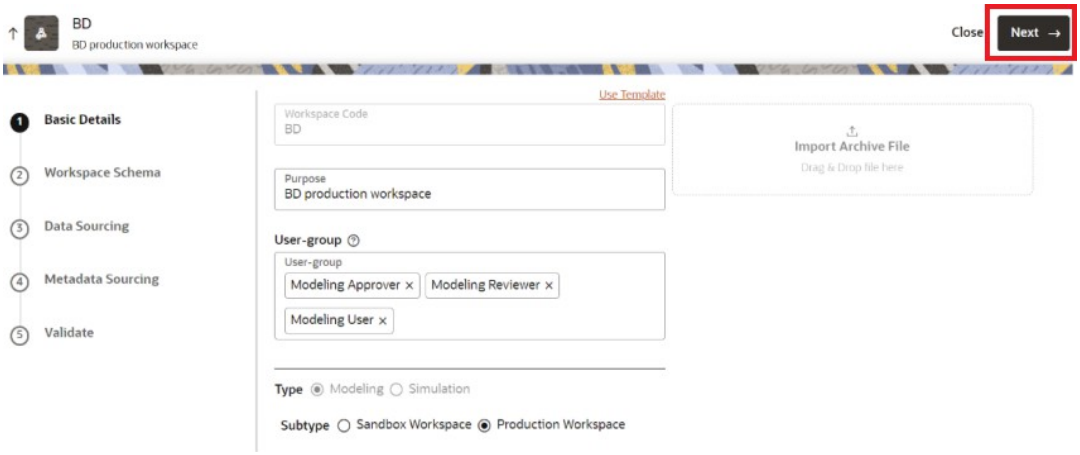
or

```
./compliance-studio --enroll
```

To register conda environment in the BD production workspace, follow these steps:

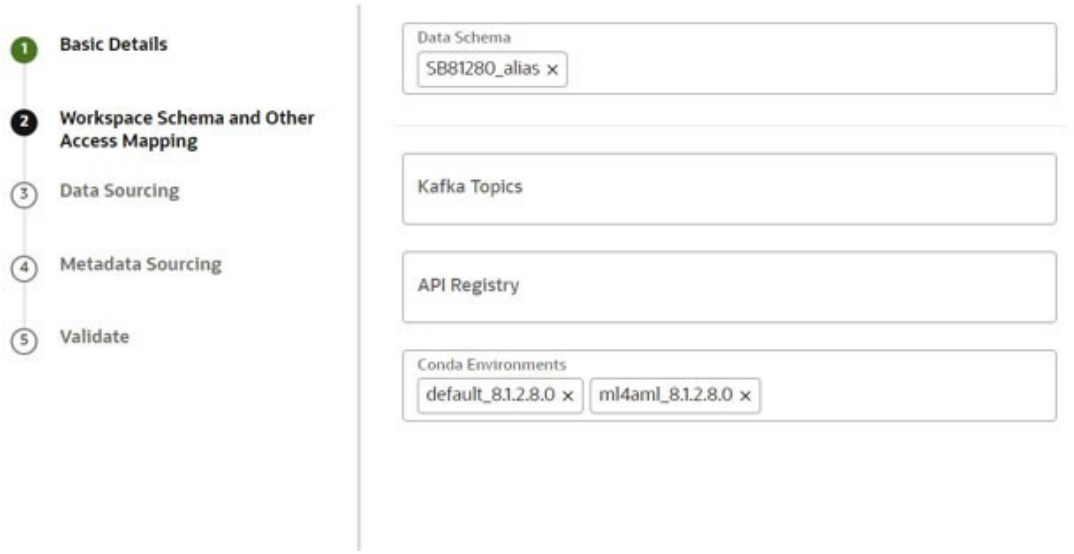
1. Navigate to **BD** workspace.
2. Click the **Action** icon and select **Edit**. The **Basic Details** pane is displayed.

Figure A-12 Basic Details Pane



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

Figure A-13 Workspace Schema



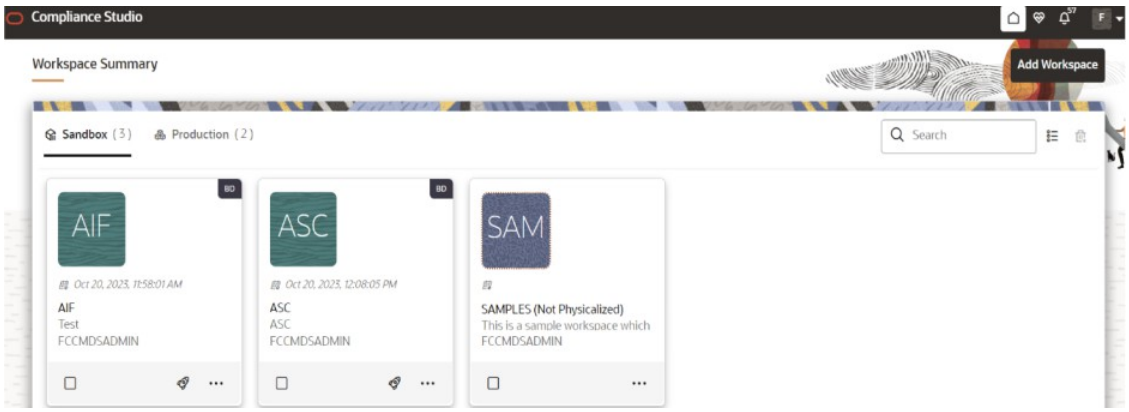
4. From the **Conda Environments**, select **default_8.1.2.8.0** and **ml4aml_8.1.2.8.0**.
5. Click **Next** to navigate to the **Data Sourcing** pane.
6. Click **Next** to navigate to the **Metadata Sourcing** pane.
7. Click **Update**. The conda environments are updated in the BD production workspace.

A.22.3 How to Create Sandbox Workspace

On the Workspace Summary page, click **Add Workspace**. The Workspace Creation window is displayed with the following process:

1. Basic Details
2. Workspace Schema
3. Data Sourcing
4. Metadata Sourcing
5. Validate
6. Summary

Figure A-14 Workspace Summary page

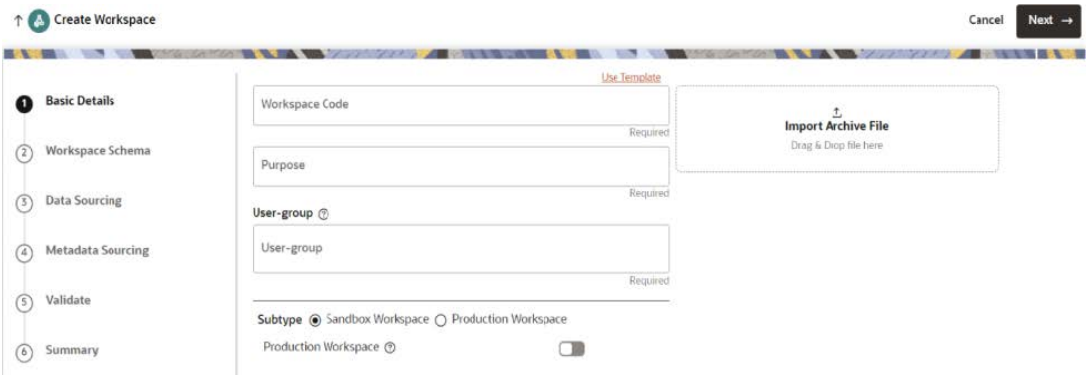


Basic Details

To create a basic details of the workspace, follow these steps:

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).
6. Click **Next**.

Figure A-15 Basic Details



Workspace Schema

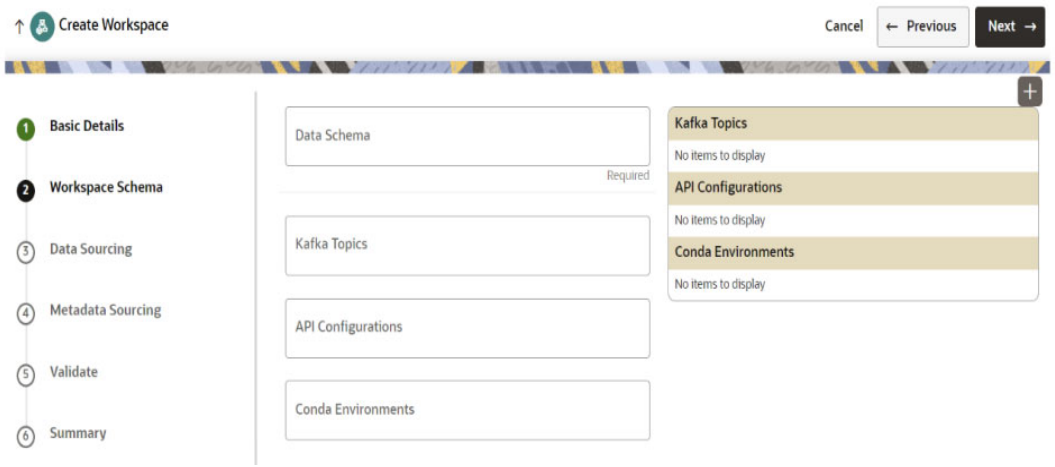
To create the workspace schema, follow these steps:

1. Select the newly created data store as **Data Schema**.

> **Note:**
>
> The Kafka Topics and API Configuration fields should be blank.

2. Select the following **Conda Environments**:
   a. **default_8.1.2.8.0**
   b. **ml4aml_8.1.2.8.0**
3. Click **Next**.

Figure A-16 Workspace Schema



Data Sourcing

> **Note:**
>
> Skip this section for **AML Event Scoring** use case.

Select the following table from the BD production datastore/ any oracle BD schema where it is having sufficient historical data.

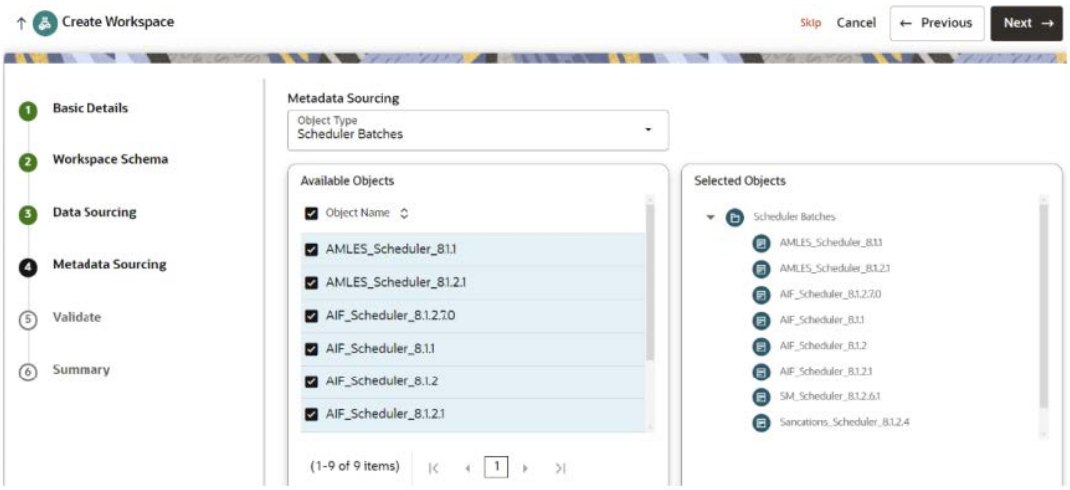
- CUST
- CUST_ACCT
- CUST_SMRY_DAILY
- CUST_SMRY_MNTH
- ACCT
- ACCT_BAL_POSN_SMRY
- ACCT_SMRY_MNTH
- ACCT_POSN
- CASH_TRXN
- WIRE_TRXN
- MI_TRXN
- BACK_OFFICE_TRXN
- TRADE
- TRADE_EXECUTION_EVENT
- SCRTY_MKT_DAILY
- SCRTY
- ORDR
- EXECUTION

- NTCPTRY_PRFL
- DERIVED_ADDRESS
- WATCH_LIST
- WIRE_TRXN_INSTN_LEG
- KDD_SCNRO
- CUST_ACCT_ROLE
- EXTERNAL_ENTITY_ADDR

Metadata Sourcing

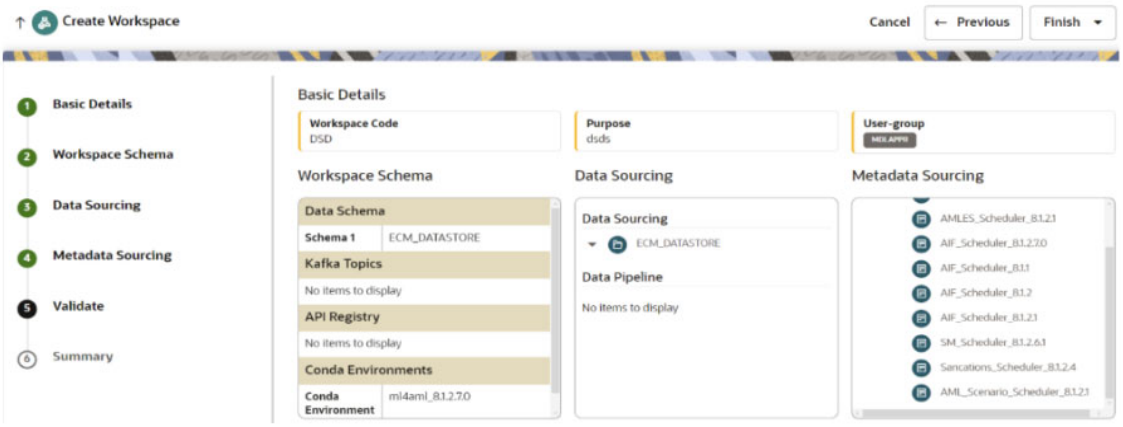
1. From the **Object Type** drop-down list, select **Scheduler Batches**.
2. In the **Available Objects**, select the scheduler based on the use case.
   - For Behavioral Model Use Case
     - **SM_Scheduler_8.1.2.6.1**
   - For Custom Scenario Use Case
     - **Custom_Scenario_Scheduler_8.1.2.8.3**
   - For AML Event Scoring Use Case
     - **AMLES_Scheduler_8.1.1**
     - **AMLES_Scheduler_8.1.2.1**
   - 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**
     - **AIF_Scheduler_8.1.2.7.0**
   - For Shell Account Detection Scenario for AML Use Case
     - **AML_Scenario_Scheduler_8.1.2.1**
   - For Customer Screening Use Case
     - **Sanctions_Scheduler_8.1.2.4**

Figure A-17 Metadata Sourcing



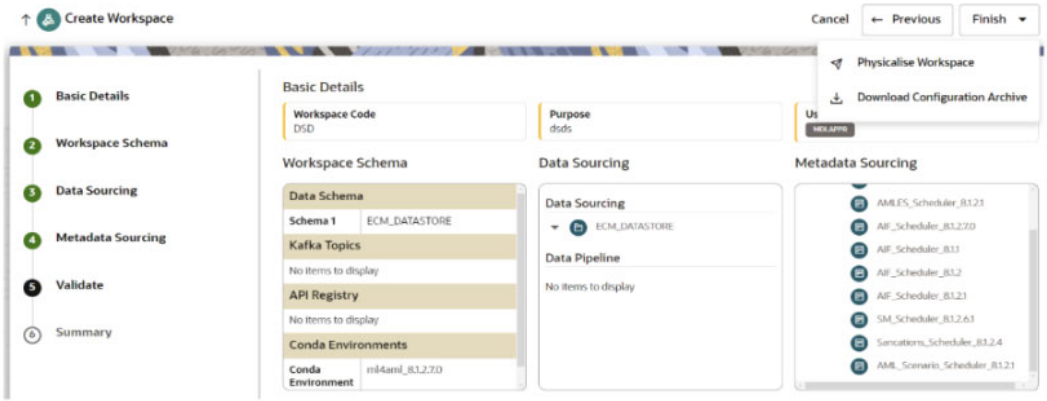
Validate Workspace

Figure A-18 Validate Workspace



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

Figure A-19 Physicalise Workspace



Summary

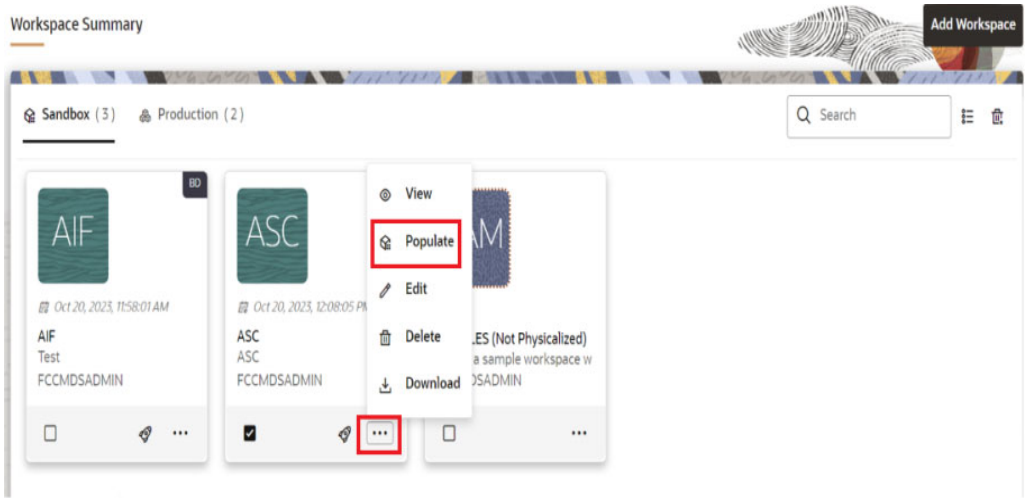
You can view summary of the created workspace.

Figure A-20 Summary

A.22.4 How to Populate the Sandbox Workspace

On the workspace summary screen, select the newly created sandbox workspace.

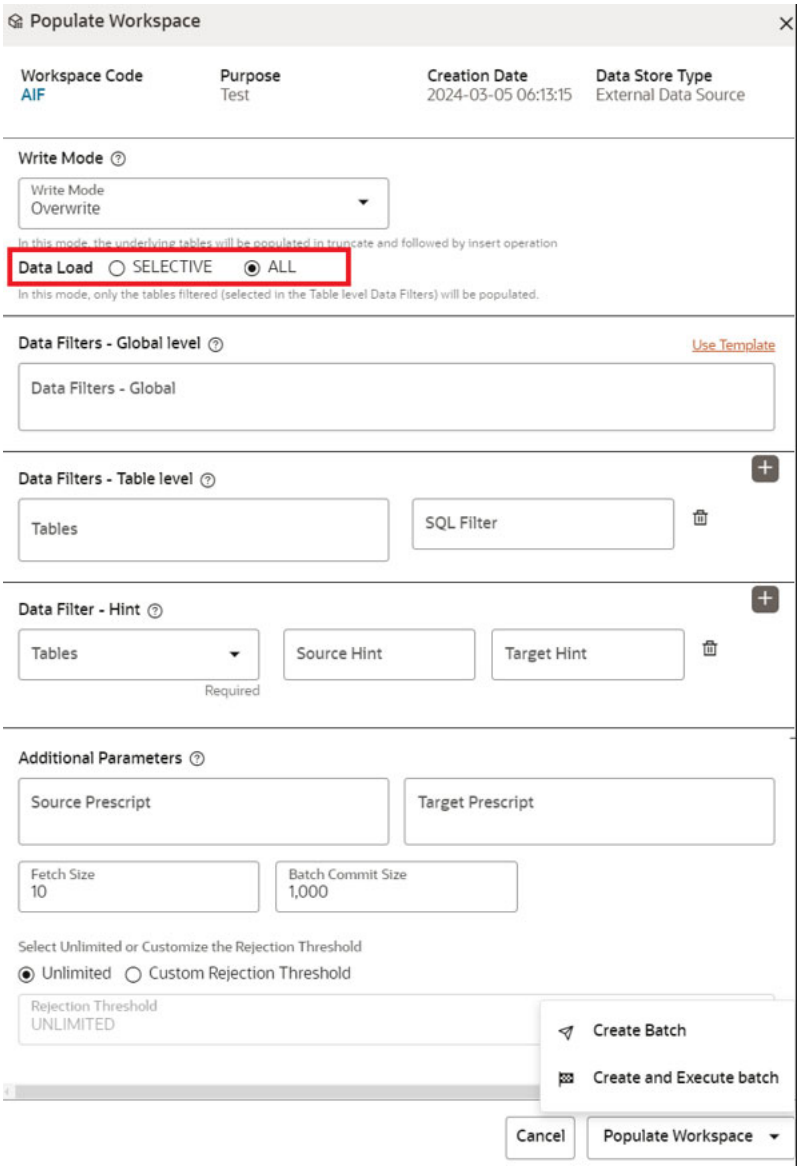
Figure A-21 Sandbox Workspace



To populate the workspace, follow these steps:

1. Click the **Action** icon and select **Populate**. The Populate Workspace window is displayed.

Figure A-22 Default Populate Workspace



2. Select options from the **Write Mode** drop-down list. The available options are:
   - **Overwrite**: In this mode, the underlying tables will be populated in truncate and followed by insert operation.
   - **Append**:In this mode, the underlying tables will be populated in append mode.

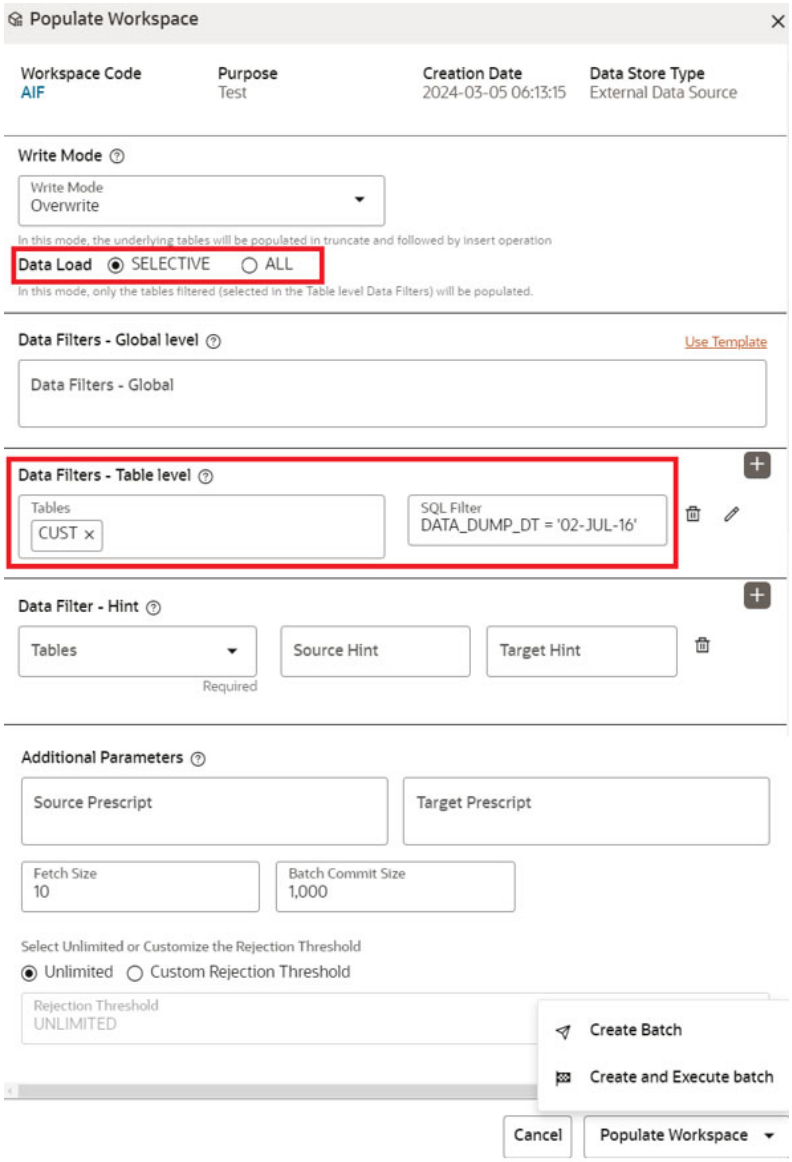
> **Note:**
>
> By default, **Overwrite Mode** is selected.

3. Select the **Data Load** options. The available options are:
   - **All**: In this type, all the underlying tables mapped to the workspace will be populated along with the filters mentioned below for specific tables.

> **Note:**
>
> By default, **ALL** is selected.

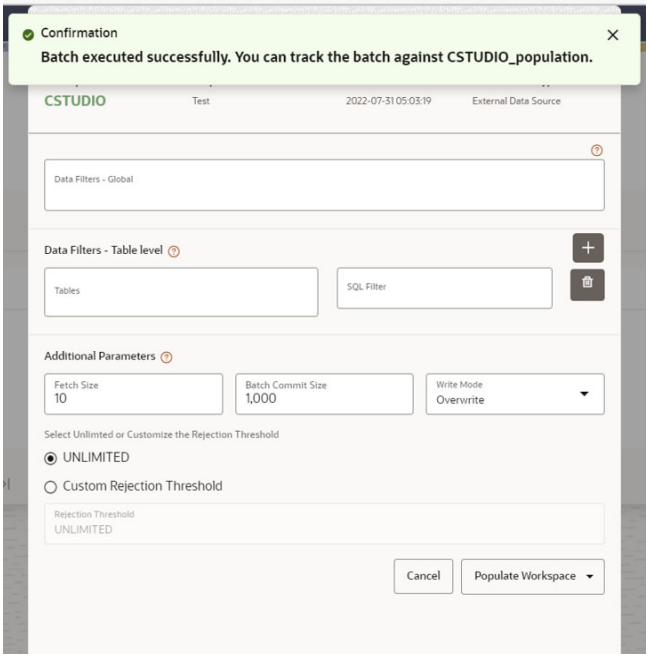
- **Selective**: In this type, only the tables filtered (selected in the Table level Data Filters) will be populated.
  - If **Data Load** is selected as **Selective**, then you need to select table and provide the column name with value in the **Data Filters - Table level** field as shown below.

Figure A-23 Populate Workspace for Selective Type



4. Select **Create and Execute** batch option. It Shows a successful message on successfully triggering the **Workspace Data Population**.

Figure A-24 Workspace Data Population

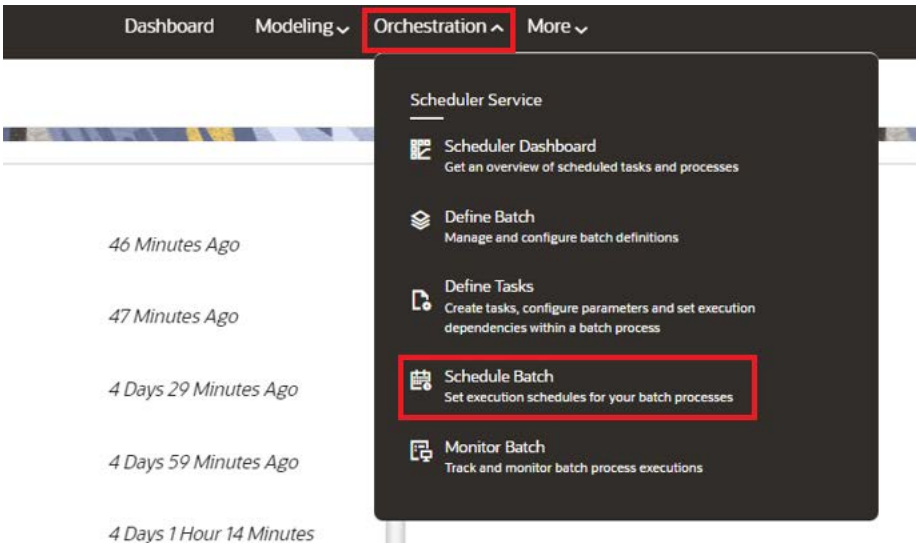


A.22.5 How to Execute Batch

To execute the batch, follow these steps:

1. On the **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 A-25 Schedule Batch

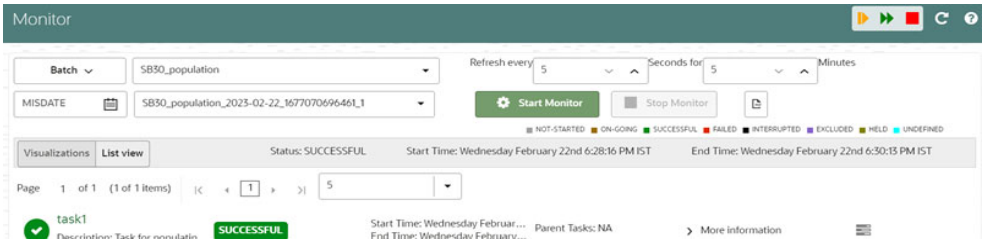


A.22.6 How to Monitor Batch

To monitor the batch, follow these steps:

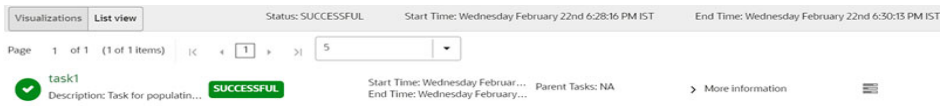
1. On the **Orchestration** menu, click **Monitor Batch**.
2. Select the desired batch name from the drop-down list.
3. Choose the batch ID that has to be monitored.
4. Click **Start Monitor** to start monitoring the batch.

Figure A-26 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 A-27 List View

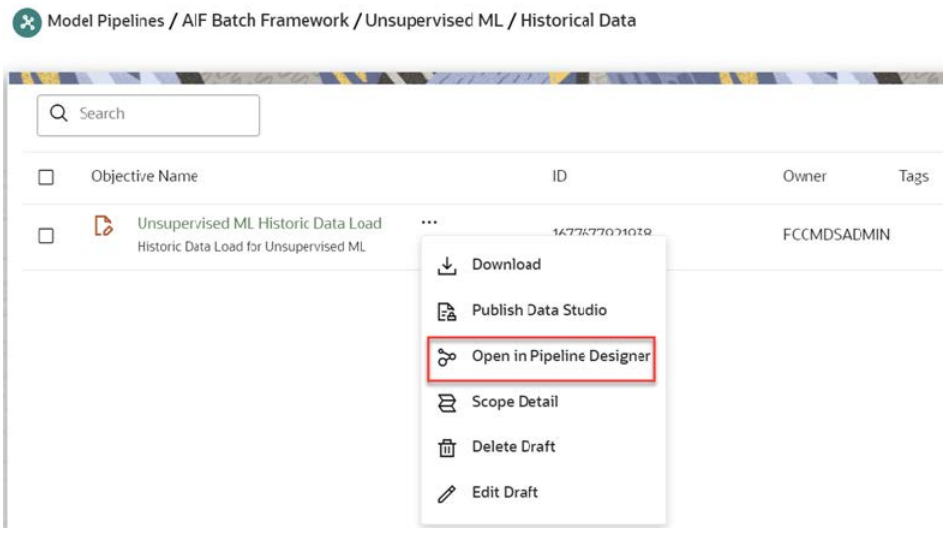


7. For further verification of the successful batch execution, navigate to "`Home > /Modeling / Pipelines/AIF Batch Framework/Unsupervised ML/Historical Data`," where the draft is located.

Figure A-28 Historical Data

8. Click the **Action** icon next to <Objective Name> to view the list of options. The following page is displayed.

Figure A-29 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 A-30 Batch Parameters

> **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/workspacepopulation` directory.

A.22.7 How to Add 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 Behavioral Model** section in the OFS Compliance Studio 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.

A.22.8 How to get Studio Alert Tables into Workspace Schema

> **Note:**
>
> You can skip this section for ML4AML use cases as these steps are taken care internally. This section is applicable only when you are using Scenario Conversion Utility as a stand alone.

To import workspace metadata, follow these steps:

1. Navigate to `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/ficdb/Scenario-Conversion-Utility/bin` directory.
2. Identify the utility and execute command as mentioned in the following table.

Table A-16 Utility for Workspace

Utility	Sandbox Workspace	Production Workspace	Command
importWorkspaceSQLCom mon.sh	Yes	Yes	./importWorkspaceSQLCommon.sh -w <workspace_wallet_alias> NOTE If the accounts belonging to a customer do not belong to the same jurisdiction as

A.23 Advanced Feature for ASC Use Case

This section explains about ASC Use case Advanced features.

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.

Optimizing SQL performance

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.

Periodic Workspace Schema Cleanup

> **Note:**
>
> This section will be performed only during end of the tuning cycle.

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;

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.

A.24 Incremental Workspace Refresh

Incremental workspace refresh helps to get the incremental data for new date from the source or adding additional partition to an existing table with respect to changes in the source.

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>
```

A.25 Data Model Support for AAI Applications

Oracle Data Model (ODM) data model support is added for the Customer Segmentation and Anomaly Detection use case only.

> **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 **Data Model Management** section in the OFS Analytical Applications Infrastructure User Guide.

A.26 Enable Additional Spark or PySpark interpreter

An additional Spark or PySpark interpreter is required to connect to two different external clusters at the same time.

To set up an additional Spark or PySpark interpreter, follow these steps:

1. Create a start-script for the second Spark interpreter.

   > **Note:**
   >
   > This is an optional step.

   a. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/interpreters/bin` directory and create a new start-script called `start-spark2-interpreter.sh` using the following command:

      ```
      cp start-spark-interpreter.sh start-spark2-interpreter.sh
      ```

   b. Edit the start-spark2-interpreter.sh file in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/interpreters/bin/` directory to update:

      i. Port number to a new port number that is not in use (for example, 7030)

      ii. Rename the log file, search for the text, .log and give a new name to the log (for example, from spark.log to spark2.log).

   c. Edit the `start-all-interpreters.sh` file in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/interpreters/bin/` directory as follows:

      i. Search for the `text sh "$DEPLOY_APP_HOME"/interpreters/bin/startspark-interpreter.sh &`

ii. Add an additional entry with `sh "$DEPLOY_APP_HOME"/interpreters/bin/ start-spark2-interpreter.sh &`

> **Note:**
>
> For the **2nd Spark** interpreter variant, use `start-spark2- interpreter.sh`, when configuring for a 3rd variant, use as `startspark3- interpreter.sh` etc.

2. Create the interpreter JSON for the additional Spark interpreter.

   a. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/interpreters/conf` directory and create the new interpreter JSON called `spark2.json` using the following command:

   ```
   cp spark.json spark2.json
   ```

   b. Edit the `spark2.json` file in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/interpreters/conf/` directory as follows:

      i. Update the following parameter values:

      ```
      group: <new-spark-interpreter-name>,
      name: <new-spark-interpreter-name>,
      groupSettings.initialCodeCapability: <new-spark-interpreter-name>,
      port: 7030 (the port chosen in the step 1),
      capabilities.name: <new-spark-interpreter-name>,
      capabilities.button.label: <new-spark-interpreter-name>,
      ```

3. After the update, the file will look like the following:

```
[
{
"group": "spark",
"name": "spark",
"className": "org.apache.zeppelin.spark.SparkInterpreter",
"groupSettings": {
"initialCode": "1+1",
"initialCodeCapability": "spark"
},
"host": "localhost",
"port": 7017,
"capabilities": [
{
"name": "spark",
"highlightLanguage": "scala",
"formEscapeCharacter": "@",
"button": {
"defaultCode": "println(\"Hello, world\")",
"icon": "fa fa-fw fa-building-o",
"label": "Spark"
}
}
],
"defaultInterpreter": true,
```

```
"properties": {
"spark.executor.memory": {
"envName": null,
"propertyName": "spark.executor.memory",
"defaultValue": "",
"description": "Executor memory per worker instance. ex) 512m,
32g",
"type": "string"
},
"args": {
"envName": null,
"propertyName": null,
"defaultValue": "",
"description": "spark commandline args",
"type": "textarea"
},
"zeppelin.spark.useHiveContext": {
"envName": "ZEPPELIN_SPARK_USEHIVECONTEXT",
"propertyName": "zeppelin.spark.useHiveContext",
"defaultValue": true,
"description": "Use HiveContext instead of SQLContext if it is
true.",
"type": "checkbox"
},
"spark.app.name": {
"envName": "SPARK_APP_NAME",
"propertyName": "spark.app.name",
"defaultValue": "Zeppelin",
"description": "The name of spark application.",
"type": "string"
},
"spark.pyspark.python": {
"envName": null,
"propertyName": "spark.pyspark.python",
"defaultValue": "python3",
"description": "Python command to run pyspark workers with",
"type": "string"
},
"zeppelin.spark.printREPLOutput": {
"envName": null,
"propertyName": "zeppelin.spark.printREPLOutput",
"defaultValue": true,
"description": "Print REPL output",
"type": "checkbox"
},
"spark.cores.max": {
"envName": null,
"propertyName": "spark.cores.max",
"defaultValue": "",
"description": "Total number of cores to use. Empty value uses
all available core.",
"type": "number"
},
"zeppelin.spark.maxResult": {
"envName": "ZEPPELIN_SPARK_MAXRESULT",
"propertyName": "zeppelin.spark.maxResult",
```

```
"defaultValue": "1000",
"description": "Max number of Spark SQL result to display.",
"type": "number"
},
"spark.master": {
"envName": "MASTER",
"propertyName": "spark.master",
"defaultValue": "yarn",
"description": "Spark master uri. ex) spark://masterhost:7077",
"type": "string"
},
"spark.yarn.archive": {
"envName": null,
"propertyName": "spark.yarn.archive",
"defaultValue": "",
"description": "An archive containing needed Spark jars for
distribution to the YARN cache",
"type": "string"
},
"spark.driver.bindAddress": {
"envName": "DRIVER_BIND_ADDRESS",
"propertyName": "spark.driver.bindAddress",
"defaultValue": "0.0.0.0",
"description": "Hostname or IP address where to bind listening
sockets.",
"type": "string"
},
"zeppelin.spark.enableSupportedVersionCheck": {
"envName": null,
"propertyName": "zeppelin.spark.enableSupportedVersionCheck",
"defaultValue": true,
"description": "Do not change - developer only setting, not for
production use",
"type": "checkbox"
},
"zeppelin.spark.uiWebUrl": {
"envName": null,
"propertyName": "zeppelin.spark.uiWebUrl",
"defaultValue": "",
"description": "Override Spark UI default URL",
"type": "string"
},
"zeppelin.spark.useNew": {
"envName": null,
"propertyName": "zeppelin.spark.useNew",
"defaultValue": true,
"description": "Whether use new spark interpreter
implementation",
"type": "checkbox"
},
"zeppelin.spark.ui.hidden": {
"envName": null,
"propertyName": "zeppelin.spark.ui.hidden",
"defaultValue": false,
"description": "Whether to hide spark ui in zeppelin ui",
"type": "checkbox"
```

```
},
"zeppelin.interpreter.output.limit": {
"envName": null,
"propertyName": "zeppelin.interpreter.output.limit",
"defaultValue": "102400",
"description": "Output message from interpreter exceeding the
limit will be truncated",
"type": "number"
}
},
"initialCode": [],
"editor": {
"language": "scala",
"editOnDblClick": false
}
}
]
```

4. Create the interpreter JSON for the second PySpark interpreter.
   a. Navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/interpreters/conf` directory and create the new interpreter JSON called `pyspark2.json` using the following command:

      ```
      cp pyspark.json pyspark2.json
      ```

   b. Edit the `pyspark2.json` file in the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/interpreters/conf/` directory as follows:
      i. Update the following parameter values:

         ```
         group: <new-spark-interpreter-name>,
         name: <new-spark-interpreter-name>,
         groupSettings.initialCodeCapability: <new-spark-interpreter-name>,
         port: 7030 (the port chosen in the step 1),
         capabilities.name: <new-spark-interpreter-name>,
         capabilities.button.label: <new-spark-interpreter-name>,
         ```

5. After the update, the file will look like the following:

```
[
{
"group": "spark",
"name": "pyspark",
"className": "org.apache.zeppelin.spark.PySparkInterpreter",
"host": "localhost",
"port": 7017,
"capabilities": [
{
"name": "pyspark",
"highlightLanguage": "python",
"button": {
"defaultCode": "print('Hello World')",
"icon": "icon-python",
"label": "PySpark"
},
```

```
"formEscapeCharacter": "$"
}
],
"properties": {
"zeppelin.pyspark.python": {
"envName": "PYSPARK_PYTHON",
"propertyName": null,
"defaultValue": "python3",
"description": "Python executable to run pyspark with",
"type": "string"
},
"zeppelin.pyspark.useIPython": {
"envName": null,
"propertyName": "zeppelin.pyspark.useIPython",
"defaultValue": false,
"description": "whether use IPython when it is available",
"type": "checkbox"
},
"zeppelin.interpreter.output.limit": {
"envName": null,
"propertyName": "zeppelin.interpreter.output.limit",
"defaultValue": "102400",
"description": "Output message from interpreter exceeding the
limit will be truncated",
"type": "number"
}
},
"initialCode": []
}
]
```

> **Note:**
>
> If you try to connect two interpreters to different external clusters when setting the environment variables, `SPARK_HOME` and `HADOOP_CONF_DIR`, as part of providing custom Spark libraries in Yarn Mode, ensure that you append the environment variables to the respective Spark interpreter start-scripts.

6. Restart Compliance Studio. To do this, navigate to the `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/bin/` directory and run the `./compliancestudio. sh -restart or ./compliance-studio.sh -r script`

A.26.1 Spark Interpreter User Impersonation

Configure the Spark cluster and Studio to allow proxy users.

Add the below properties and values in `core-site.xml` in the Spark cluster as well as Studio and restart the Spark cluster and Studio:

```
<property>
<name>hadoop.proxyuser.zeppelin.groups</name>
<value>*</value>
</property>
<property>
```

```
<name>hadoop.proxyuser.zeppelin.hosts</name>
<value>*</value>
</property>
```

Configure the Spark interpreter to run the spark-submit job as the currently logged-in user.

Add the below property in `spark.json`:

```
"zeppelin.spark.run.asLoginUser": {
"envName": null,
"propertyName": "zeppelin.spark.run.asLoginUser",
"defaultValue": true,
"description": "Whether run spark job as the zeppelin login user, it is
only applied when running spark job in hadoop yarn cluster and shiro is
enabled",
"type": "checkbox"
}
```

> **Note:**
>
> There will be only a single keytab used by all Spark interpreter runs.

A.26.2 Sample spark-default.conf Configuration File

Here is the sample code block for creating `spark-default.conf` file.

```
spark.driver.port 30303
spark.blockManager.port 31313
spark.driver.bindAddress 0.0.0.0
spark.yarn.dist.files <COMPLIANCE STUDIO INSTALLTION PATH>/deployed/mmg-home/
mmg-studio/interpreter-server/spark-interpreter-<version>/extralibs/spark-
<version>-bin-hadoop<version>/python/lib/pyspark.zip,<COMPLIANCE STUDIO
INSTALLTION PATH>/deployed/mmg-home/mmg-studio/interpreter-server/
sparkinterpreter-<
version>/extralibs/spark-<version>-bin-hadoop<version>/python/
lib/py4j-0.10.7-src.zip
spark.executorEnv.PYTHONPATH pyspark.zip:py4j-0.10.7-src.zip
spark.driver.defaultJavaOptions "-Dsun.security.krb5.debug=false -
Djavax.security.auth.useSubjectCredsOnly=false -
Djava.security.krb5.conf=<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/
batchservice/user/conf/krb5.conf"
spark.driver.host <FQDN_HOSTNAME>
spark.yarn.keytab <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/
batchservice/user/conf/fccstudio.keytab
spark.yarn.principal <KRBS_PRINCIPAL>
spark.yarn.kerberos.relogin.period 1m
```

> **Note:**
>
> - **FQDN_HOSTNAME** stands for compliance Studio Fully Qualified hostname, and **KRBS_PRINCIPAL** stands for Kerberos principal.
> - For example, the Spark version is **spark-2.4.0-bin-hadoop2.7**.

A.27 Enable Data Studio Options in Compliance Studio

In order to see Data Studio options for a particular user, make sure that the following groups are assigned:

- IDNTYAUTH
- IDNTYADMN
- DSUSRGRP

A.28 Rebuilding Indices in OpenSearch

Indices rebuild is required when there is mismatch between the records in the database and OpenSearch indexes. The following steps are also applicable even if there is a mismatch between any of the indexes.

To rebuild indices in the OpenSearch, follow these steps:

1. Execute the following curl command to delete the index:

   ```
   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
   ```

   > **Note:**
   >
   > The curl command should be executed in the Compliance Studio server.

2. Execute the following to Load data for __**prev index** for the runskey LESS THAN the last successful ER batch runskey.

   a. URL: `http://<hostname>:<port>/load-to-open-search/idx/createIndex`
   For example: `http://testserver.com:7053/load-to-open-search/idx/createIndex`

   b. **Request Body**: The JSON request body can be obtained using below query:

   ```
   SELECT V_IDX_JSON FROM FCC_IDX_M_JSON_MAP WHERE
   V_PIPELINE_ID='<PIPELINE_ID>';
   ```

> **Note:**
>
> In the below request, ##LATEST_RUN_SKEY_OF_LAST_SUCCESSFULL_ERJOB## is the latest runskey for which all 4 ER Jobs were executed successfully.

i. Make the following changes in the json keys:

```
"loadType": "DeltaLoad"
"tableName" :"FCC_ER_FULL"
"filterCondition": "N_RUN_SKEY <
##LATEST_RUN_SKEY_OF_LAST_SUCCESSFULL_ERJOB##"
```

For example,

```
"loadType": "DeltaLoad"
"tableName" :"FCC_ER_FULL"
"filterCondition": "N_RUN_SKEY <196"
```

> **Note:**
>
> Here, 196 is the latest runskey for which all 4 ER Jobs were executed successfully.

ii. Ensure that the "deletedProfilesTableName" key and its value are not in the request body.

iii. Provide the ER schema alias name as "schemaName".
For example, "schemaName": "ER_SCHEMA_ALIAS"

iv. Provide the wallet path in the "walletFilePath" key.
For example, "walletFilePath": "/scratch/test/testpath/compStudio_ 31010949/OFS_- COMPLIANCE_STUDIO/wallet"

v. Provide tnsnames.ora file path in the "walletFilePath" key.
For example, "tnsOraFilePath": "/scratch/test/testpath/compStudio_31010949/OFS_- COMPLIANCE_STUDIO/wallet"

The sample Request body for __prev index is as follows:

```
{
"schemaName": "ER_SCHEMA_ALIAS",
"walletFilePath":"/scratch/test/testPath/OFS_COMPLIANCE_STUDIO/
wallet",
"tnsOraFilePath":"/scratch/test/testPath/OFS_COMPLIANCE_
STUDIO/wallet",
"tableName": "FCC_ER_FULL",
"filterCondition": "N_RUN_SKEY < 196",
"indexName": "stg_party_812",
"indexAlias": "csa_812_alias",
"indexLogicalName": "csa_stg_party_812",
"indexBusinessName": "csa_stg_party_812",
"indexKeyAttribute": "original_id",
```

```
"deleteProfilesIdxKeyAttribute":"v_party_id",
"loadType": "DeltaLoad",
"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": "mdm_id",
"type": "text",
"similarity": "boolean",
"analyzerType": "pipe_delimiter",
"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"
],
"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"]
}
```

CURL COMMAND:

```
curl -XPOST http://hostname:port/load-to-open-search/idx/
createIndex -H 'Content-Type: application/json' -d'<request_body>'
```

For example,

```
curl -XPOST http://testserver:7053/load-to-open-search/idx/
createIndex -H 'Content-Type: application/json' -d'<request_body>'
```

3. Execute the following to load data for **__delta index** for the runskey EQUAL TO last successful ER batch runskey.

a. URL: `http://<hostname>:<port>/load-to-open-search/idx/createIndex`
For example: `http://testserver.com:7053/load-to-open-search/idx/createIndex`

b. **Request Body**: The JSON request body can be obtained using below query:

```
SELECT V_IDX_JSON FROM FCC_IDX_M_JSON_MAP WHERE
V_PIPELINE_ID='<PIPELINE_ID>';
```

> **Note:**
>
> In the below request, ##LATEST_RUN_SKEY_OF_LAST_SUCCESSFULL_ERJOB## is the latest runskey for which all 4 ER Jobs were executed successfully.

i. Make the following changes in the json keys:

```
"loadType": "DeltaLoad"
"tableName" :"FCC_ER_FULL"
"filterCondition": "N_RUN_SKEY =
##LATEST_RUN_SKEY_OF_LAST_SUCCESSFULL_ERJOB##>"
```

For example,

```
"loadType": "DeltaLoad"
"tableName" :"FCC_ER_FULL"
"filterCondition": "N_RUN_SKEY = 196"
```

> **Note:**
>
> Here, 196 is the latest runskey for which all 4 ER Jobs were executed successfully.

ii. Ensure that the "deletedProfilesTableName" key and its value are not in the request body.

iii. Provide the ER schema alias name as "schemaName".
For example, "schemaName": "ER_SCHEMA_ALIAS"

iv. Provide the wallet path in the "walletFilePath" key.
For example, "walletFilePath": "/scratch/test/testpath/compStudio_ 31010949/OFS_- COMPLIANCE_STUDIO/wallet"

v. Provide tnsnames.ora file path in the "walletFilePath" key.
For example, "tnsOraFilePath": "/scratch/test/testpath/compStudio_31010949/OFS_- COMPLIANCE_STUDIO/wallet"

The sample Request body for __delta index is as follows:

```
{
"schemaName": "ER_SCHEMA_ALIAS",
"walletFilePath":"/scratch/test/testPath/OFS_COMPLIANCE_STUDIO/
wallet",
```

```
"tnsOraFilePath":"/scratch/test/testPath/OFS_COMPLIANCE_
STUDIO/wallet",
"tableName": "FCC_ER_FULL",
"filterCondition": "N_RUN_SKEY = 196",
"indexName": "stg_party_812",
"indexAlias": "csa_812_alias",
"indexLogicalName": "csa_stg_party_812",
"indexBusinessName": "csa_stg_party_812",
"indexKeyAttribute": "original_id",
"deleteProfilesIdxKeyAttribute":"v_party_id",
"loadType": "DeltaLoad",
"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": "mdm_id",
"type": "text",
"similarity": "boolean",
"analyzerType": "pipe_delimiter",
"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"
],
"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"]
}
```

CURL COMMAND:

```
curl -XPOST http://hostname:port/load-to-open-search/idx/
createIndex -H 'Content-Type: application/json' -d'<request_body>'
```

For example,

```
curl -XPOST http://testserver:7053/load-to-open-search/idx/
createIndex
-H 'Content-Type: application/json' -d'<request_body>'
```