

**Oracle Financial Services Crime and
Compliance Studio**

Administration Guide

Release 8.0.7.4.0

April 2020

E91246-01

ORACLE®

Financial Services

OFS Crime and Compliance Studio Administration and Configuration Guide

Copyright © 2020 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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

For information on third party licenses, click [here](#).

Document Control

Version Number	Revision Date	Changes Done
8.0.7.4.0	Updated: April 2020	<ul style="list-style-type: none">• FCC Studio is enhanced to include a new menu item in the UI to configure the Entity Resolution rules within the application. For more information, see Managing Rulesets.• The Financial Crime Graph Model has been enhanced with the following:<ul style="list-style-type: none">■ FCC Studio can use either the BD or ECM schema as the source of FCDM data for the graph. When data is loaded from ECM, the graph includes CASE nodes and event edges. When ECM integration is required, ECM must be used as the source to include the case information.■ The datatype of Date attributes to local_date from string to enable network evolutionFor more information, see Oracle Financial Crime Graph Model.• Integrate FCC Studio with IDCS, as the SSO provider. For more information, see Managing User Administration for IDCS.• fcc-jdbc and jdbc interpreters can use Oracle Wallet credentials to enable secure data access. For more information, see the fcc-jdbc Interpreter and jdbc Interpreter.• You can change Python Version in the fcc-python Interpreter. For more information, see Changing Python Version in the fcc-python Interpreter.• You can add desired Python packages to the fcc-python Interpreter. For more information, see Adding Python Packages to the fcc-python Interpreter.

Version Number	Revision Date	Changes Done
8.0.7.3.0	Updated: March 2020	<ul style="list-style-type: none"> • The Financial Crime Graph Model has been modified from Homogenous to Heterogenous format, that is, the node and edge types can have specific properties that are relevant to their type. • The following changes have been introduced to the graph model to make the graph more suitable for users initiating investigations. <ul style="list-style-type: none"> ■ The intermediaries in the Out-of-the-box graphs have been removed and are not loaded by default for ease of investigation. ■ The derived entity node logic has been changed such that the Name and ID information are combined in the derived entity nodes to prevent multiple transaction edges to entities with the same name. <p>For more information, see Oracle Financial Crime Graph Model.</p> • The ETL process has been changed as there is no longer a need to homogenize the nodes and edges before loading into the graph. For more information, see Appendix - Creating and Executing Run Executable.
8.0.7.2.0	Updated: February 2020	Updated the guide for 8.0.7.2.0 release.
8.0.7.1.0	Updated: October 2019	Updated the guide for 8.0.7.1.0 release.
8.0.7.0.0	Created: Feb 2019	Created the first version of the Crime and Compliance Studio Administration Guide for 8.0.7.0.0 Release.

Table of Contents

1	About this Guide.....	8
1.1	Summary.....	8
1.2	Audience	8
1.3	Related Documents.....	8
1.4	Abbreviations	8
2	About Oracle Financial Services Crime and Compliance Studio	10
2.1	Introduction to FCC Studio.....	10
2.2	The Architecture of FCC Studio.....	11
2.3	Oracle Financial Crime Graph Model	11
3	Managing User Administration	13
3.1	Realm-Based Authorization for FCC Studio	13
3.1.1	<i>Accessing FCC Studio using FCCMRealm</i>	<i>13</i>
3.1.2	<i>Accessing FCC Studio using DemoRealm</i>	<i>13</i>
3.1.3	<i>Accessing FCC Studio using IdcsRealm</i>	<i>14</i>
4	Accessing Crime and Compliance Studio	15
5	Managing Permissions	17
5.1	Permissions Overview	17
5.2	Accessing Permissions Page.....	19
5.2.1	<i>Users</i>	<i>20</i>
5.2.2	<i>Groups</i>	<i>20</i>
5.2.3	<i>Roles</i>	<i>20</i>
5.2.4	<i>Permission Templates</i>	<i>21</i>
6	Managing Tasks	22
6.1	Accessing Tasks.....	22
6.2	Task Statuses	22
6.3	Table Columns.....	23
6.4	Table Filters	23
7	Managing Credentials.....	25
7.1	Accessing Credentials	25
7.2	Using Credentials	27

7.2.1	<i>Linking Credentials to Interpreter Variants</i>	27
8	Managing Rulesets	29
8.1	Accessing the Rulesets.....	29
8.2	Creating Rulesets	29
8.3	Creating Rules in a Ruleset.....	30
8.3.1	<i>Scoring Method</i>	31
9	Configuring Interpreters	33
9.1	Accessing Interpreters.....	33
9.2	Creating a New Interpreter Variant	33
9.2.1	<i>Creating a New fcc-jdbc Interpreter Variant</i>	34
9.3	Configure Interpreters.....	34
9.3.1	<i>fcc-jdbc Interpreter</i>	35
9.3.2	<i>fcc-ore Interpreter</i>	37
9.3.3	<i>fcc-pyspark Interpreter</i>	38
9.3.4	<i>fcc-python Interpreter</i>	40
9.3.5	<i>fcc-spark-scala Interpreter</i>	41
9.3.6	<i>fcc-spark-sql Interpreter</i>	42
9.3.7	<i>jdbc Interpreter</i>	44
9.3.8	<i>md Interpreter</i>	45
9.3.9	<i>pgql Interpreter</i>	46
9.3.10	<i>pgx-algorithm Interpreter</i>	46
9.3.11	<i>pgx-java Interpreter</i>	47
9.3.12	<i>pyspark Interpreter</i>	47
9.3.13	<i>spark Interpreter</i>	47
10	Configuring ETL	49
10.1	Performing Data Source Configuration.....	49
11	Managing FCC Studio Batches	52
11.1	Preparing for Batches.....	52
11.2	Performing Batches	52
11.2.1	<i>Data Movement and Graph Loading for Big Data Environment</i>	52
11.2.2	<i>Executing Published Notebook</i>	54

- 12 Appendix - Creating and Executing Run Executable..... 56**
- 13 Appendix - Modifying the Python Images for the Python Interpreter..... 62**
 - 13.1 Preparing for Building a Python Image 62
 - 13.1.1 *Modifying the Python Packages in the Python Interpreter* 62
 - 13.1.2 *Adding Different Version of Python to Python Interpreter* 63
 - 13.2 Building and Pushing an Image 63
 - 13.3 Replacing Python Image in FCC Studio 64
- 14 Appendix - Spark Query Parameters 65**
- 15 Appendix - Performing Graph Configurations..... 66**
 - 15.1 Attributes Case in Graph..... 66
 - 15.2 Extra Empty Nodes and Edges Providers 67
 - 15.3 Additional Configuration 68
- 16 OFSAA Support Contact Details 69**
- 17 Send Us Your Comments..... 70**

1 About this Guide

Topics:

- [Summary](#)
- [Audience](#)
- [Related Documents](#)
- [Abbreviations](#)

1.1 Summary

This guide provides instructions for maintaining and configuring the OFS Crime and Compliance Studio (FCC Studio) application, its subsystem components, and any third-party software required for operation.

FCC Studio provides an open and scalable infrastructure that supports end-to-end functionality across all Oracle Financial Services solution sets. FCC Studio’s extensible and modular architecture enables a customer to deploy new solution sets readily as the need arises.

1.2 Audience

This guide is intended for administrators and implementation consultants. Their roles and responsibilities, as they operate within FCC Studio, include the following:

- **System Administrator:** Configures and maintains the system, user accounts, and roles, monitors data management, archives data, loads data feeds, reloads cache, and performs post-processing tasks.

1.3 Related Documents

You can access the following additional documents related to the OFS Crime and Compliance Studio application from the [Oracle Help Center \(OHC\)](#) Documentation Library:

- *Oracle Financial Services Crime and Compliance Studio Installation Guide*
- *Oracle Financial Services Crime and Compliance Studio Deployment Guide (Using Kubernetes)*
- *Oracle Financial Services Crime and Compliance Studio User Guide*
- *Oracle Financial Services Crime and Compliance Studio Data Model Guides*
- *Oracle Financial Services Crime and Compliance Studio Release Notes and Readme*

1.4 Abbreviations

The following table lists the abbreviations used in this document:

Table 1: Abbreviations

Abbreviation	Meaning
OFS	Oracle Financial Services
FCC Studio	Financial Crime and Compliance Studio
OFSAA	Oracle Financial Services Analytical Application

Table 1: Abbreviations

Abbreviation	Meaning
PGX	Parallel Graph AnalytiX
AML	Anti-money Laundering
FCDM	Financial Crime Data Model
BD	Behavior Detection
SSO	Single Sign-On
IDCS	Oracle Identity Cloud Service
ECM	Enterprise Case Management
ETL	Extract, Transform, Load
K8s	Kubernetes

2 About Oracle Financial Services Crime and Compliance Studio

This chapter provides functional details about Oracle Financial Services (OFS) Crime and Compliance Studio (FCC Studio) application.

Topics:

- [Introduction to FCC Studio](#)
- [The Architecture of FCC Studio](#)
- [Oracle Financial Crime Graph Model](#)

2.1 Introduction to FCC Studio

To effectively monitor anti-money laundering and anti-fraud programs in financial institutions, the most challenging requirement is to quickly identify and adapt to the changing patterns of financial crime. This ability to discover new and emerging criminal behavioral patterns, coupled with the facility to rapidly deploy as models, is a critical requirement.

Data scientists and analysts can use FCC Studio to interactively explore financial crime data and gain insights into new and emerging financial crime patterns and trends.

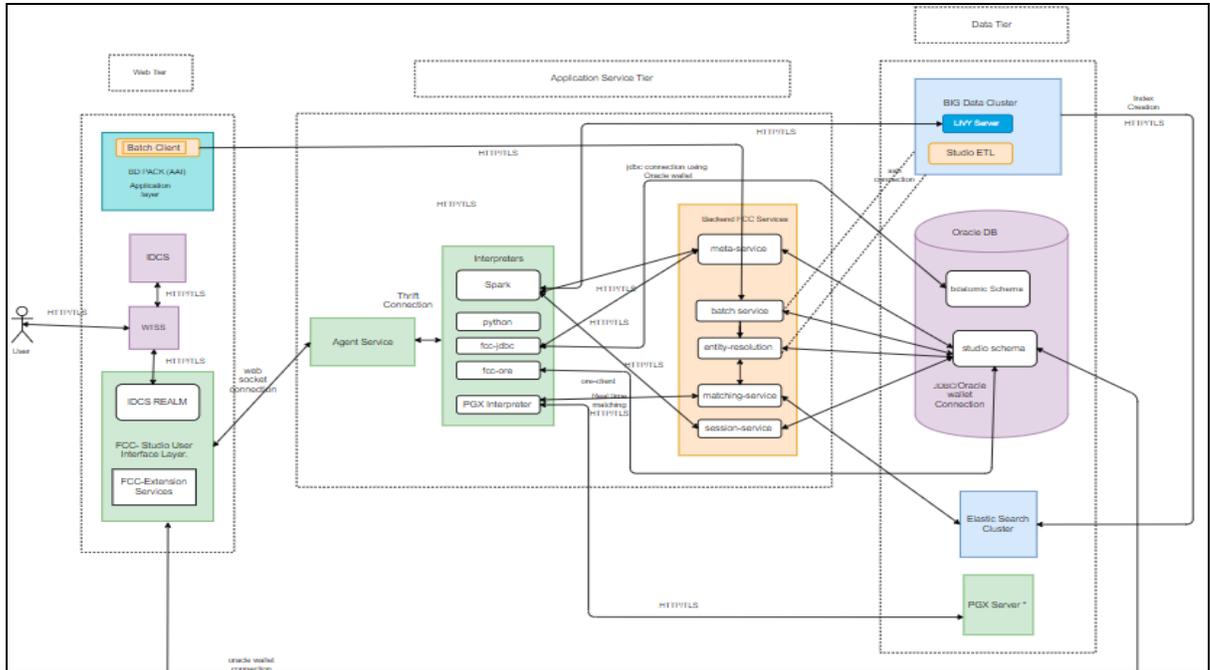
The key features of FCC Studio include the following:

- Provides an integrated and comprehensive analytics toolkit designed to rapidly discover and model new financial crime patterns.
- Interacts with the database, process the data, and generate patterns in various formats using interpreters.
- Provides secure access to an institution's financial crime data with predefined scenarios, out-of-the-box graph queries, and visualizations.
- Uses Graph Analytics and Graph Query methods to analyze historic data available in the database, and forecast the generated patterns using various interpreters.
- Uses Machine Learning Algorithms to gain insights from historical alert data to prioritize the alerts generated by the detection engines.
- Offers a unified tool for Graph Analytics, Data Visualization, Machine Learning, Scenario Authoring, Pattern Discovery, Data Mashups, and testing for financial crime data.
- Works with Apache Spark, the most prevalent analytics engine on Big Data.
- Works with Apache Zeppelin, a web-based notebook that enables interactive data analysis.
- Supports Polyglot Scenario Authoring to author new scenarios in SQL, Scala, Python, or R language.
- Embedded with a highly scalable in-memory Graph Analytics Engine (PGX).
- Enterprise-ready with underlying OFSAA framework.
- Works with earlier 8.x releases of Oracle Financial Crime and Compliance Management Anti Money Laundering (AML) and Fraud applications.
- Integrated with Oracle Financial Crime Application Data and readily usable across the enterprise financial crime data lake. This can automatically load Oracle AML and Fraud data into the data lake and mashup FCC Studio data with third-party data for discovery and modeling.

2.2 The Architecture of FCC Studio

The following figure illustrates the architecture of the FCC Studio application.

Figure 1: FCC Studio Architecture

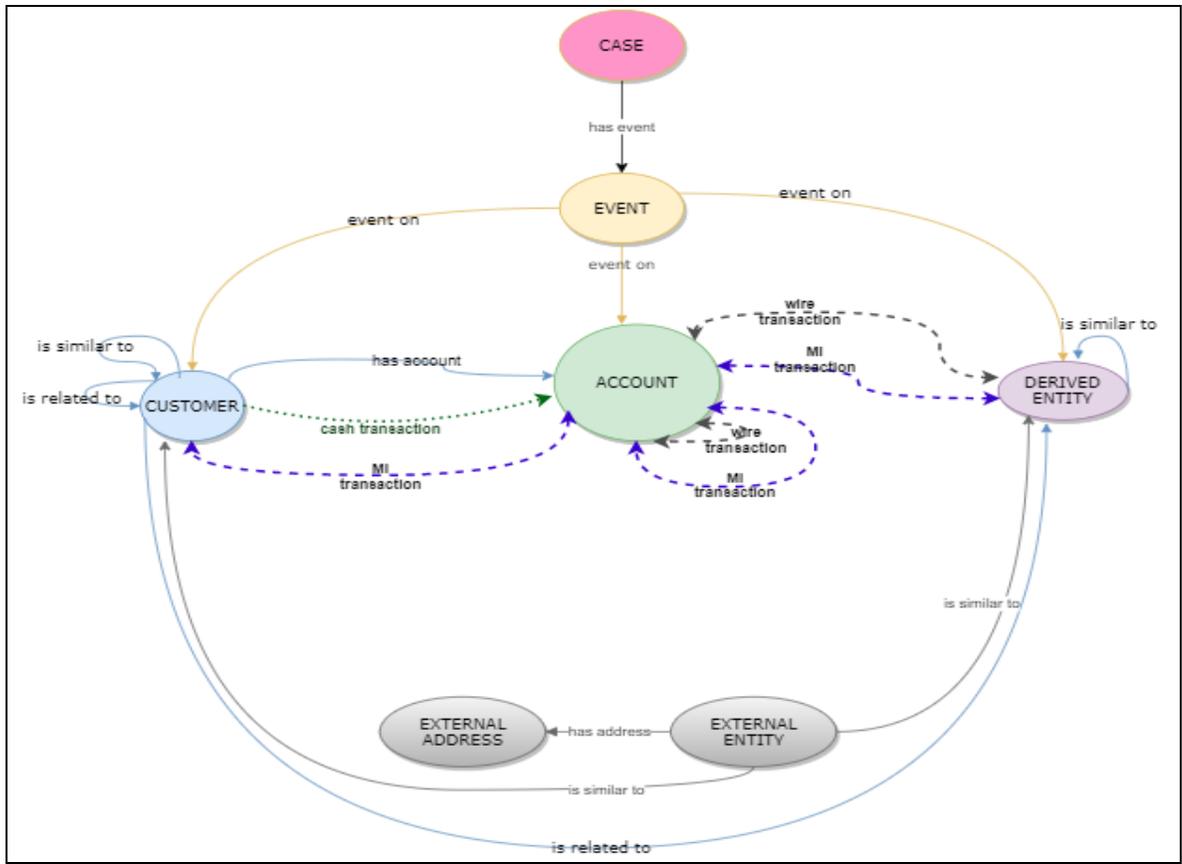


2.3 Oracle Financial Crime Graph Model

The Oracle Financial Crime Graph Model serves as a window into the financial crimes data lake. It collates disparate data sets into an enterprise-wide global graph, enabling a whole new set of financial crime use cases. The Graph model enables to accelerate financial crime investigation use cases.

For information on Graph Data Model, see [Graph Data Model](#).

For information on the node and edge properties of the Oracle Financial Crime Graph Model, see the [Data Model Guide](#).



NOTE

The Case node in this Financial Crime Graph Model is loaded only when you load the FCDM data from Enterprise Case Management (ECM). When data is loaded from ECM, the graph includes “CASE” nodes and “has event” edges.

3 Managing User Administration

User Administration refers to the process of controlling the user privileges in accessing the application resources and is based on business requirements to provide access to view, create, edit, or delete confidential data.

User Administration involves administrator tasks to create user definitions, user groups, maintain profiles, authorize users and user groups, and map users to groups, domains and roles, grant permissions based on user roles and requirements, and so on.

3.1 Realm-Based Authorization for FCC Studio

Realm indicates the functional grouping of database schemas and roles that must be secured for an application. Realms protect data from access through system privileges and do not provide additional privileges to its owner or participants.

User administration for FCC Studio is performed using realm-based authorization and authentication. Realm-based authorization establishes a set of database accounts and roles, that can manage or access objects protected in realms, and are authorized to use its system privileges. It provides a runtime mechanism to logically check if a user's command is allowed to access objects specified in the command and to proceed with its execution.

After you create a realm, you can register a set of schema objects or roles (secured objects) for realm protection and authorize a set of users or roles to access the secured objects.

The FCC Studio application can be accessed using the following realms:

- FCCMRealm. For more information, see [Accessing FCC Studio using FCCMRealm](#).
- DemoRealm. For more information, see [Accessing FCC Studio using DemoRealm](#).
- IdcsRealm. For more information, see [Accessing FCC Studio using IdcsRealm](#).

3.1.1 Accessing FCC Studio using FCCMRealm

This section provides information on the creation of users who can access FCC Studio using the FCCMRealm method of authentication and authorization through OFSAAI. The users with SYSADMIN and SYSAUTH roles in OFSAAI can create and authorize users respectively. For more information on creating and authorizing users in OFSAAI, see the [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

The following table describes the roles and the corresponding user groups who can access FCC Studio using FCCMRealm.

Table 1: Roles and User Groups in FCC Studio

Role	User Groups
DSADMIN	DSADMINGRP
DSINTER	DSINTERGRP
DSUSER	DSUSERGRP
DSBATCH	DSBATCHGRP

3.1.2 Accessing FCC Studio using DemoRealm

The following table describes the roles and the corresponding users who can access FCC Studio using DemoRealm.

NOTE This realm is used to access FCC Studio only for demo purpose and must not be used in the production environment.

Table 2: Roles and User Groups in FCC Studio

Role	Users
DSADMIN	DSADMIN
DSINTER	DSINTER
DSUSER	DSUSER
DSBATCH	DSBATCH

3.1.3 Accessing FCC Studio using IdcsRealm

This section provides information on managing users who can access FCC Studio with Oracle Identity Cloud Service (IDCS). IDCS acts as the Single Sign-On (SSO) service provider for implementations between FCC Studio, Investigation Hub, and Enterprise Case Management. This configuration prevents separate logins for each application.

To integrate FCC Studio with IDCS as the SSO provider, follow these steps:

1. Create the following user groups in the IDCS application:
 - DSADMINGRP
 - DSINTERGRP
 - DSUSERGRP
 - DSBATCHGRP
2. Map the user groups to the respective user based on the user roles.

For more information on Managing Users with Oracle Identity Cloud Service Accounts, see <https://docs.oracle.com/en/cloud/get-started/subscriptions-cloud/mmocs/managing-users-identity-cloud-service-accounts-1.html>.

4 Accessing Crime and Compliance Studio

To access the Crime and Compliance Studio application as a system administrator, follow these steps:

1. Enter the URL in the following format into the browser:

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

Here <Port_Number> is,

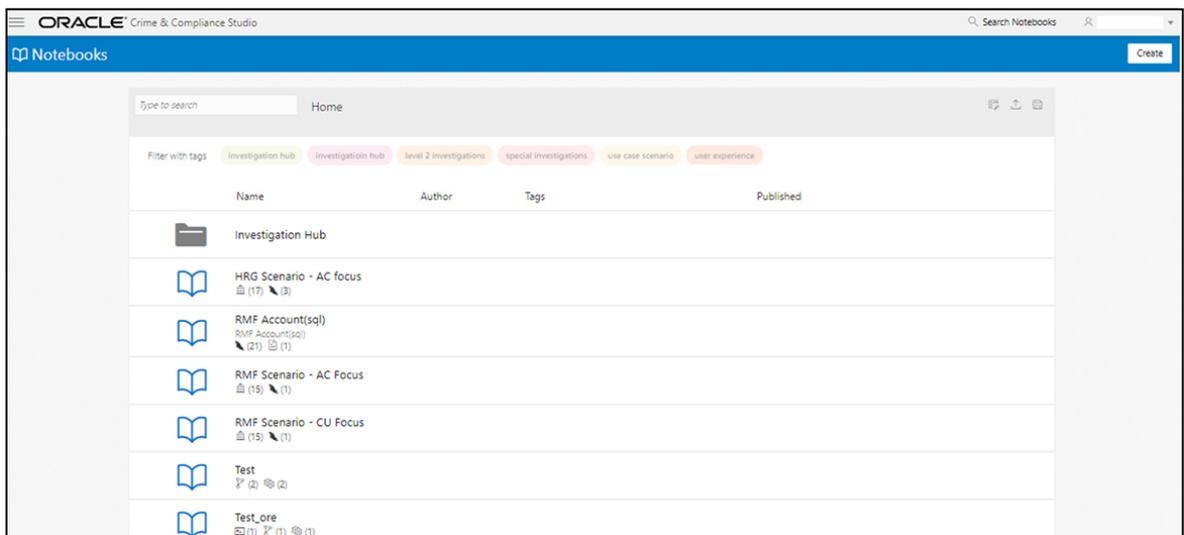
- 7008 for FCC Studio installed on-premise.
- 30078 for FCC Studio deployed on the Kubernetes cluster.

The Crime and Compliance Studio login page is displayed.

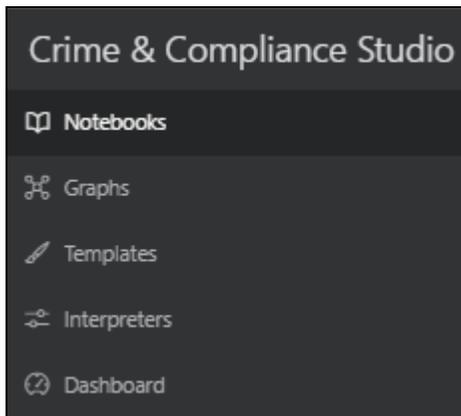


2. Login with the **Username** and **Password** of the System Administrator.
3. Click **Login**.

The Crime and Compliance Studio application's landing page is displayed.



4. Click the Navigation Menu  on the upper-left corner.
The menu items applicable to the logged-in user are displayed.



For information on Notebooks, Graphs, and Templates, see the *Oracle Financial Services Crime and Compliance Studio User Guide*.

5 Managing Permissions

This section provides information on users, groups, roles, and other permissions.

Topics:

- [Permissions Overview](#)
- [Accessing Permissions Page](#)

5.1 Permissions Overview

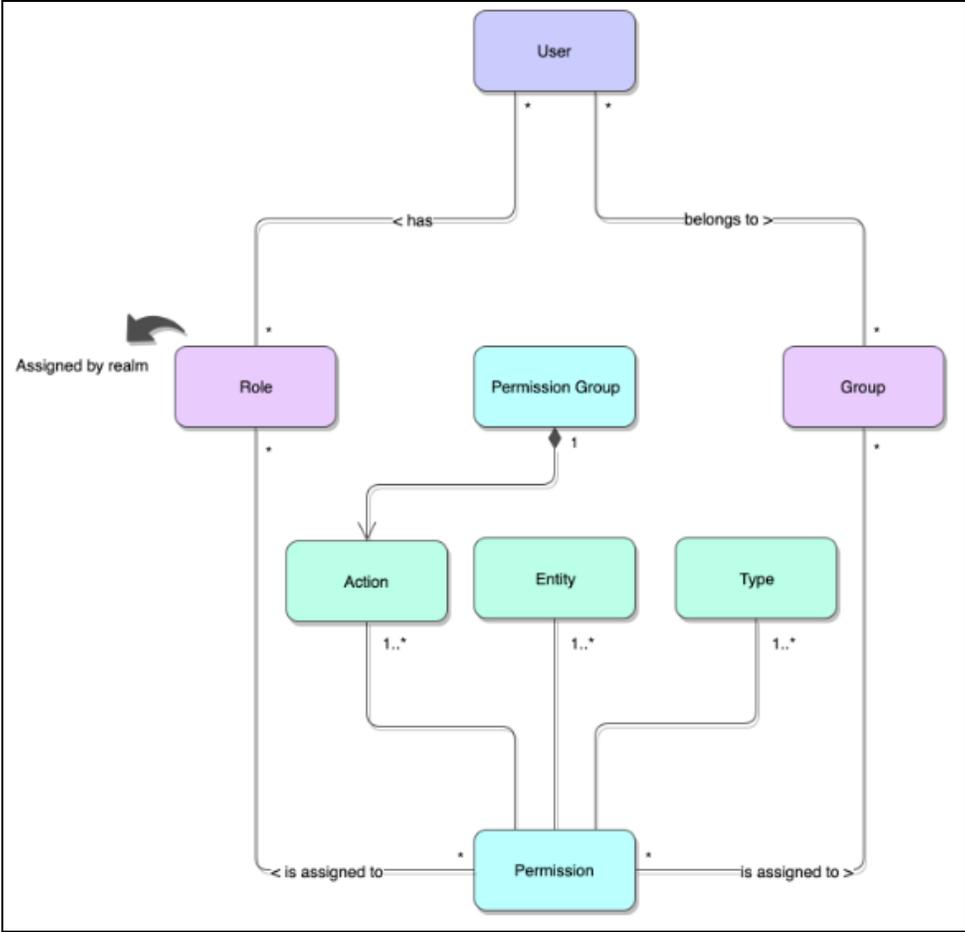
FCC Studio offers a rich permission system. The four key components are permissions, permission templates, roles, and groups.

A permission is an action that applies to entities and types. A permission can be a general action, an action on all entities of a particular type, or an action on a specific entity of a particular type.

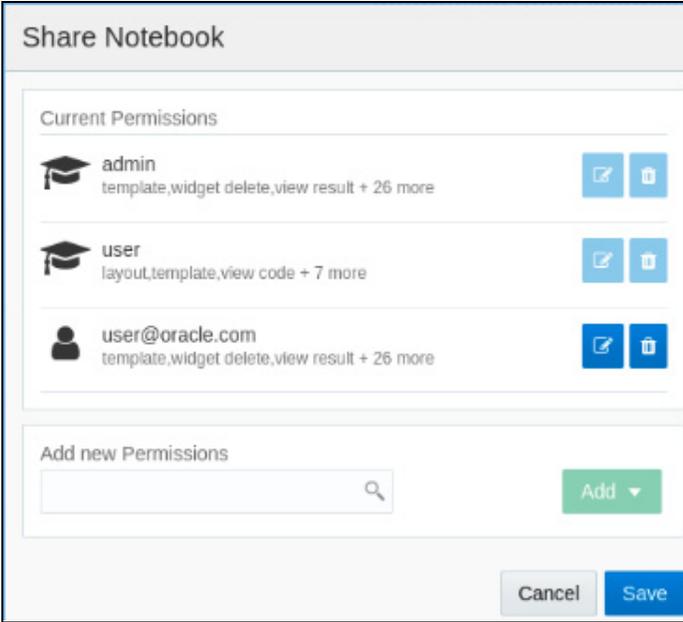
For example:

- `general:create_notebook` - The general action to create a notebook.
- `graph:graph_update` - The action to update all entities of type graph, that is, the action to update all the graphs in the **Graphs** page.
- `notebook:export` - The action to export a specific entity of type notebook, that is, the action to export a particular notebook.

To evaluate whether a user can perform an action on an entity of a type, the permission must be assigned to the user, to one of their roles, or groups.



When sharing a notebook, you can select the user(s)/member(s) to share a notebook and grant the required permission.



Select Permissions

Permission-Group

read write own

limited_read

▲ Advanced Mode

<input type="checkbox"/> *	<input checked="" type="checkbox"/> attach	<input checked="" type="checkbox"/> clear
<input checked="" type="checkbox"/> clone	<input checked="" type="checkbox"/> detach	<input checked="" type="checkbox"/> export
<input checked="" type="checkbox"/> iframe	<input checked="" type="checkbox"/> layout	<input checked="" type="checkbox"/> restart_cluster
<input checked="" type="checkbox"/> run_all	<input checked="" type="checkbox"/> snapshot	<input checked="" type="checkbox"/> template
<input checked="" type="checkbox"/> toggle_show_code	<input checked="" type="checkbox"/> update	<input checked="" type="checkbox"/> view
<input checked="" type="checkbox"/> view_code	<input checked="" type="checkbox"/> view_result	<input checked="" type="checkbox"/> widget_create
<input checked="" type="checkbox"/> widget_delete	<input checked="" type="checkbox"/> widget_execute	<input checked="" type="checkbox"/> widget_modify
<input checked="" type="checkbox"/> widget_move	<input checked="" type="checkbox"/> widget_view	

Save Cancel

5.2 Accessing Permissions Page

To access permissions, follow these steps:

1. Navigate to the FCC **Studio** workspace.
2. Click the Navigation Menu  on the upper-left corner.
The menu items are listed.
3. Click **Permissions**.

The **Permissions** page lists the following:

- [Users](#)
- [Groups](#)
- [Roles](#)
- [Permission Templates](#)

The screenshot shows the 'Permissions' page with three main sections: Users, Groups, and Roles. Each section has a table listing items and their associated permissions.

Username	Last Login	Roles	Groups	Permissions	Creation Time	Actions
admin@oracle.com	05 Sep 2019 05:21 pm	admin	-		05 Sep 2019 05:08 pm	
guest@oracle.com	-	-	-		05 Sep 2019 02:23 pm	
user10@oracle.com	-	-	-	view_sessions, view, paragraph_comment, export, iframe, paragraph_view, view_code, view_result, graph_view, export_all	05 Sep 2019 02:23 pm	

Name	Permissions	Creation Time	Actions
group1		05 Sep 2019 02:23 pm	
group2	view_result, clone, iframe, view, paragraph_comment, view_code, snapshot, paragraph_view, view_sessions, export, Show More ...	05 Sep 2019 02:23 pm	
group3	view_sessions, view, paragraph_comment, view_result, paragraph_view, export, view_code, iframe, graph_view, export_all	05 Sep 2019 02:23 pm	

Name	Permissions	Creation Time	Actions
admin		05 Sep 2019 02:23 pm	
user	view_result, view_sessions, paragraph_view, snapshot, iframe, clone, view, export, view_code, paragraph_comment, Show More ...	05 Sep 2019 02:23 pm	

5.2.1 Users

The Users section lists all the users, the date of their last login, and their roles, groups, other permissions. Users cannot be added or deleted in this section, but the groups they belong to can be updated.

This screenshot shows the 'Users' section with a table listing user details. The permissions for 'user10@oracle.com' are highlighted.

Username	Last Login	Roles	Groups	Permissions	Creation Time	Actions
admin@oracle.com	05 Sep 2019 05:21 pm	admin	-		05 Sep 2019 05:08 pm	
guest@oracle.com	-	-	-		05 Sep 2019 02:23 pm	
user10@oracle.com	-	-	-	view_sessions, view, paragraph_comment, export, iframe, paragraph_view, view_code, view_result, graph_view, export_all	05 Sep 2019 02:23 pm	

5.2.2 Groups

A Group consists of one or more permissions. A user can belong to multiple groups. For example, Oracle Labs, General, and External. The **Groups** section allows users to view and manage all groups. Groups can be added, updated, and deleted.

This screenshot shows the 'Groups' section with a table listing group details. The permissions for 'group3' are highlighted.

Name	Permissions	Creation Time	Actions
group1		05 Sep 2019 02:23 pm	
group2	view_result, clone, iframe, view, paragraph_comment, view_code, snapshot, paragraph_view, view_sessions, export, Show More ...	05 Sep 2019 02:23 pm	
group3	view_sessions, view, paragraph_comment, view_result, paragraph_view, export, view_code, iframe, graph_view, export_all	05 Sep 2019 02:23 pm	

5.2.3 Roles

A Role consists of one or more permissions. A user can have multiple roles that are typically assigned by the realm. For example, admin, user, and guest. All the listed roles can be added, updated, and deleted.

This screenshot shows the 'Roles' section with a table listing role details. The permissions for 'user' are highlighted.

Name	Permissions	Creation Time	Actions
admin		05 Sep 2019 02:23 pm	
user	view_result, view_sessions, paragraph_view, snapshot, iframe, clone, view, export, view_code, paragraph_comment, Show More ...	05 Sep 2019 02:23 pm	

5.2.4 Permission Templates

A Permission Template is a set of actions. For example, limited_read, read, and write. All the listed Permission Templates can be added, updated, and deleted.

The screenshot shows a web interface titled "Permission Templates" with a subtitle: "Set of (often logically related) actions to simplify applying multiple permissions (e.g. when assigning permissions in the 'Share Notebook' dialog)". A blue button in the top right corner says "+ Create Permission Template".

Name	Permissions	Creation Time	Actions
read	#name view export paragraph_view view_code clone paragraph_comment view_sessions snapshot view_result Show More ...	05 Sep 2019 02:23 pm	
write	view_sessions toggle_show_code update paragraph_execute paragraph_create template clone paragraph_delete attach clear Show More ...	05 Sep 2019 02:23 pm	
own	•	05 Sep 2019 02:23 pm	
limited_read	#name paragraph_comment view_code view_result view_sessions view export paragraph_view graph_view export_all	05 Sep 2019 02:23 pm	

6 Managing Tasks

This chapter provides information on the tasks in FCC Studio and the user associated with each task.

Topics:

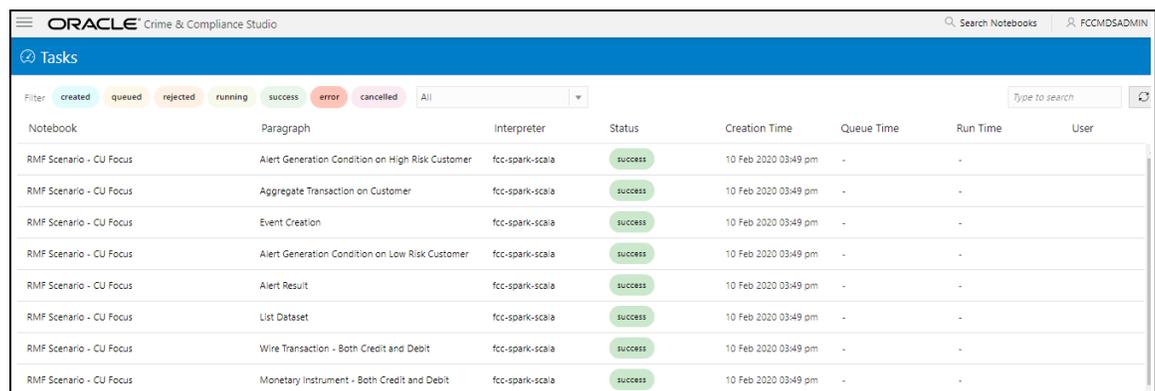
- [Accessing Tasks](#)
- [Task Statuses](#)
- [Table Columns](#)
- [Table Filters](#)

6.1 Accessing Tasks

To access Tasks, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Click the Navigation Menu  on the upper-left corner.
The menu items are listed.
3. Click **Tasks**.

The **Tasks** page lists all tasks of FCC Studio and displays the notebook, paragraph, interpreter, and user associated with each task.



Notebook	Paragraph	Interpreter	Status	Creation Time	Queue Time	Run Time	User
RMF Scenario - CU Focus	Alert Generation Condition on High Risk Customer	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	Aggregate Transaction on Customer	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	Event Creation	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	Alert Generation Condition on Low Risk Customer	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	Alert Result	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	List Dataset	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	Wire Transaction - Both Credit and Debit	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	
RMF Scenario - CU Focus	Monetary Instrument - Both Credit and Debit	fcc-spark-scala	success	10 Feb 2020 03:49 pm	-	-	

6.2 Task Statuses

The tasks can have the following possible statuses:

Table 1: Task Statuses

Field	Description
created	Indicates that the task is just created.
queued	Indicates that the task is in a queue, waiting to be run. This can happen when the same user runs multiple paragraphs of the same interpreter in the same notebook - the interpreter will first finish executing the first paragraph (i.e., task) and then moves onto the second one which will have status queued until then.
running	Indicates that the tasks are being executed.

Table 1: Task Statuses

Field	Description
rejected	Indicates that the task is rejected.
success	Indicates that the task is successfully completed.
canceled	Indicate that the execution of the task is canceled (For example, by clicking the 'Cancel Execution' ("Stop") button on a paragraph).
error	Indicates that an error occurred during the execution of a task. The error can be one of the following: <ul style="list-style-type: none"> • The concerned interpreter is unsupported • The interpreterClient is disconnected • The task is not found • The status of the task cannot be changed to running or success

6.3 Table Columns

Table 2: Table Columns

Field	Description
Notebook	Indicates the name of the notebook for which the task was created for.
Paragraph	Indicates the title of the paragraph associated with the task, or if there is no title, the first line of code of the paragraph
Interpreter	Indicates the name of the interpreter that the task was created to run against
Status	Indicates the status of the task
Creation Time	Indicates the time (in device-local time) when the task was created
Queue Time	Indicates the total time spent by the task in the queue, that is, the time between the creation of the task and the beginning of the task execution
Run Time	Indicates the time taken for the task to run, that is, the time between the beginning and the end of the task execution
User	Indicates the username of the user who created the task

6.4 Table Filters

The list of tasks can be filtered in many ways to make it easy to search for specific tasks. You can filter a task by the following categories:

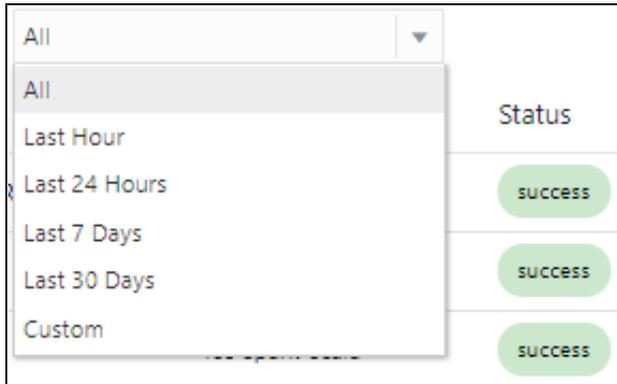
- **Task status:** Filters the tasks based on the statuses that match the selected statuses.



- **Text:** The search box allows the user to filter the tasks based on the notebook name, paragraph title, paragraph code, interpreter, and user.



- **Date and Time:** Filters the tasks based on task creation time. The following options are available:



- Last Hour
- Last 24 Hours
- Last 7 days
- Last 30 days
- Custom: Allows the user to enter a custom from and/or to date-time. If empty, it assumes infinite past/future.

7 Managing Credentials

FCC Studio provides secure and safe credential management. For example, passwords, Oracle Wallets, or KeyStores. The credentials can be used to connect to data sources.

Topics:

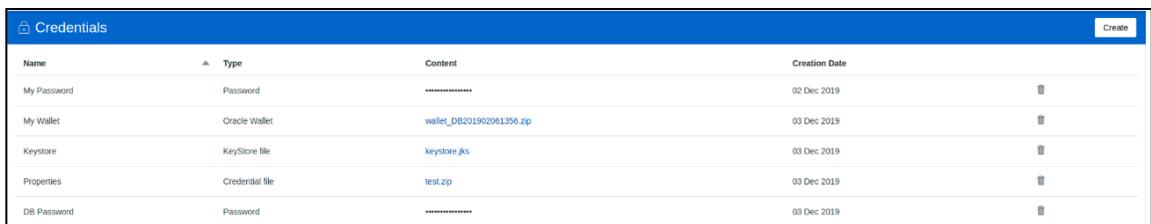
- [Accessing Credentials](#)
- [Using Credentials](#)

7.1 Accessing Credentials

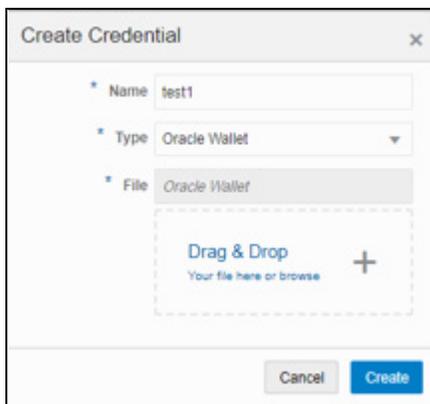
To access the Credentials page, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Click the Navigation Menu  on the upper-left corner.
The menu items are listed.
3. Click **Credentials**.

The *Credentials* page is displayed.



4. Create the wallet credential.
 - a. Click **Create** on the top-right corner to create a new wallet credential.
The **Create Credential** dialog box is displayed.



- b. Enter the following details:

Table 1: Wallet Credential Details

Field	Description
Name	Enter the name for the wallet credential.

Table 1: Wallet Credential Details

Field	Description
Type	Select Oracle Wallet.
File	<p>Upload the wallet zip file that includes the following files:</p> <ul style="list-style-type: none"> • <code>cwallet.sso</code> • <code>ewallet.p12</code> • <code>tnsnames.ora</code> <p>NOTE:</p> <ul style="list-style-type: none"> • The wallet file must be in .zip format. • The maximum file size allowed for the credential file is 128Kb.

- c. Click **Create**.

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

5. Create the password credential for the wallet.

- a. Click **Create** on the top-right corner to create a new password credential for the wallet.

The **Create Credential** dialog box is displayed.

- b. Enter the following details:

Table 2: Wallet Credential Details

Field	Description
Name	Enter the name for the password credential.
Type	Select Password.
File	Enter the wallet password for the password credential.

- c. Click **Create**.

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

6. To download the credential files, click the credential file name on the *Credentials* page.

7. To delete a credential, click the  Delete in the last column of the row of the credential.

7.2 Using Credentials

You can use the credentials by linking them to certain interpreter variants to enable secure data access.

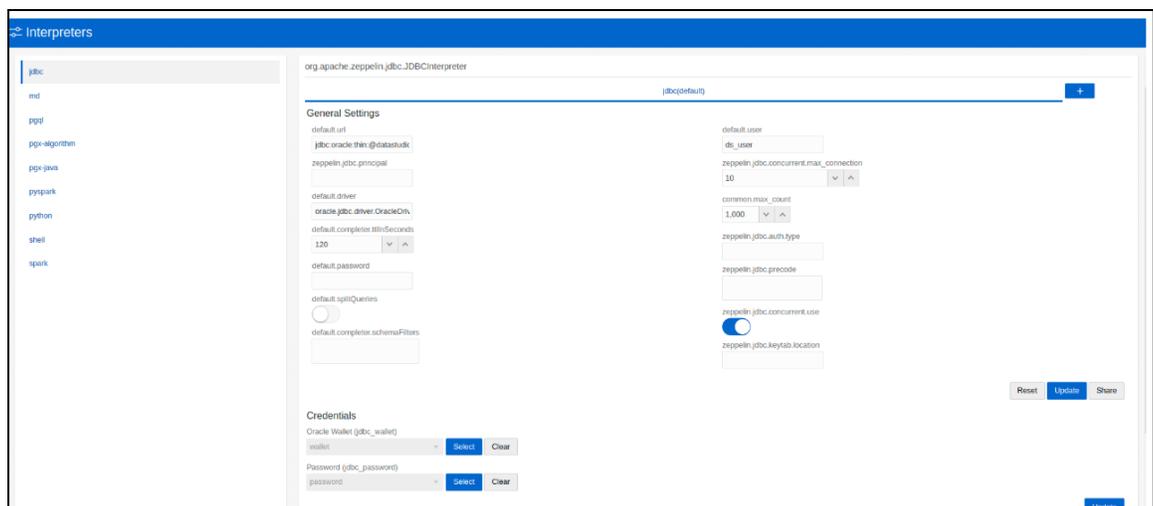
7.2.1 Linking Credentials to Interpreter Variants

You can use the credentials by linking them to certain interpreter variants to enable secure data access. If an interpreter variant is enabled to accept credentials, the *Credentials* section is displayed in the *Interpreters* page.

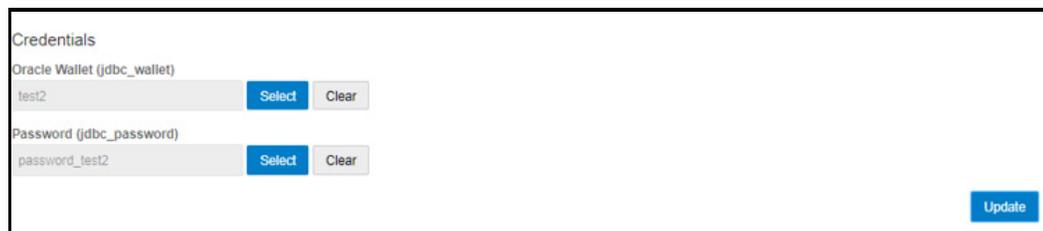
To link a credential to an interpreter variant, follow these steps:

1. Navigate to the *Interpreters* page.
2. Click the interpreter from the list displayed on the LHS.

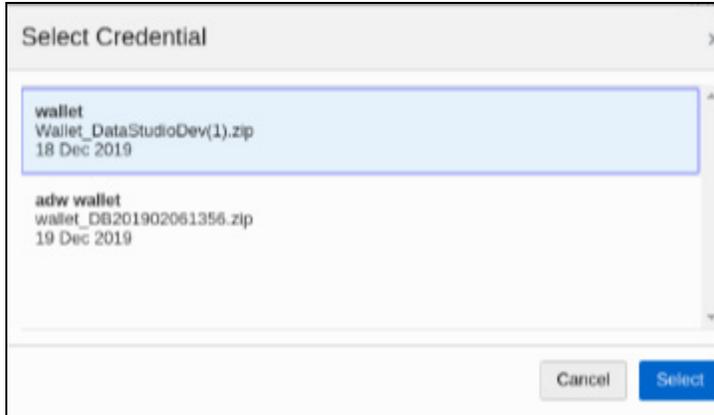
The interpreter variant is displayed on the RHS.



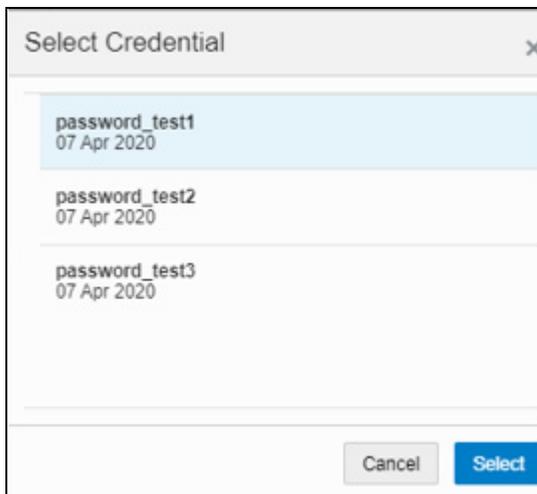
3. Go to the *Credentials* section.



4. Select the wallet credential that you want to link to the Interpreter variant from the displayed popup and Click **Select**.

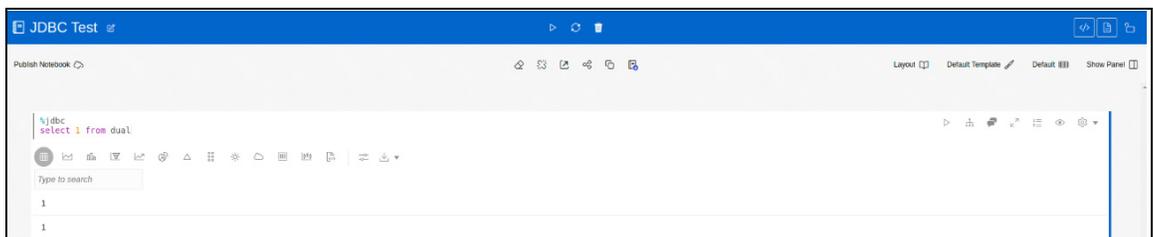


5. Select the password credential that you want to link to the Interpreter variant from the displayed popup and Click **Select**.



6. After you have linked your credentials, click **Update** to save the changes.

For example, you can link the credentials (a wallet and a password) to JDBC interpreter variants to enable secure data access. This linking enables the JDBC interpreter to securely connect to the specified Oracle DB.



8 Managing Rulesets

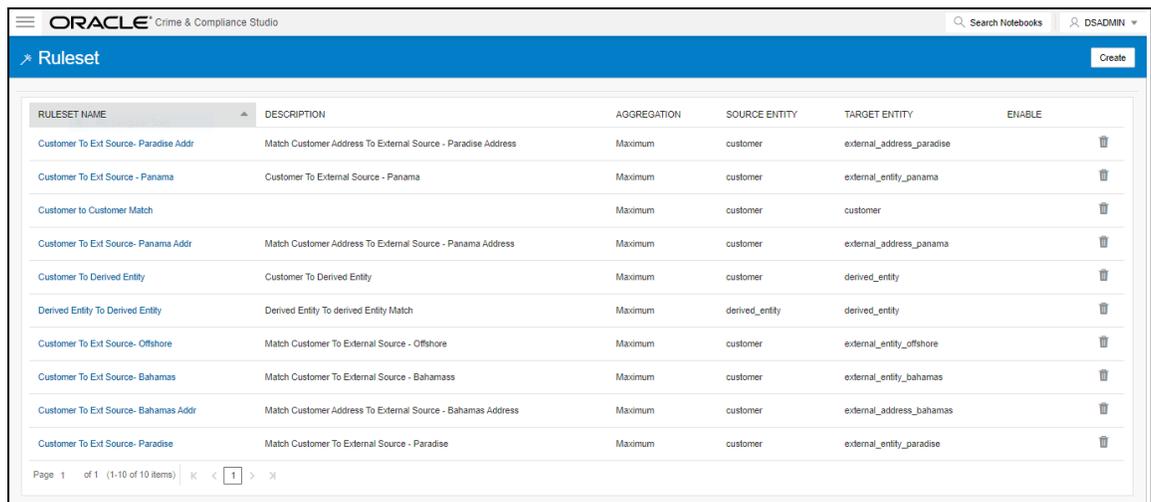
A Ruleset is a set of rules that are applied to the defined source and target entities, compares the attributes of the entities to derive a match. For information on matching rulesets, see the *Financial Crime Graph Model Matching Guide*.

8.1 Accessing the Rulesets

To access the Rulesets page, follow these steps:

1. Navigate to the FCC Studio workspace.
2. Click the Navigation Menu  on the upper-left corner.
The menu items are listed.
3. Click **Ruleset**.
The *Ruleset* page is displayed with all the out-of-the-box rulesets.

Figure 1: Ruleset Page

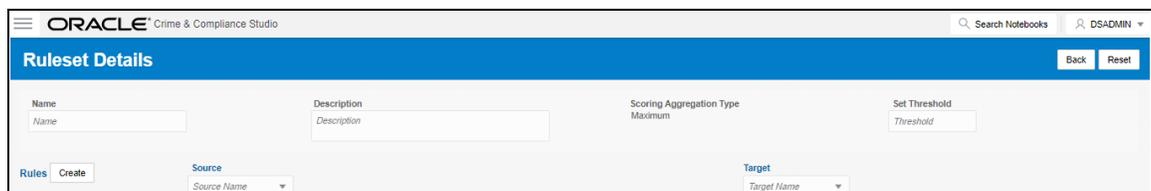


RULESET NAME	DESCRIPTION	AGGREGATION	SOURCE ENTITY	TARGET ENTITY	ENABLE
Customer To Ext Source- Paradise Addr	Match Customer Address To External Source - Paradise Address	Maximum	customer	external_address_paradise	
Customer To Ext Source - Panama	Customer To External Source - Panama	Maximum	customer	external_entity_panama	
Customer to Customer Match		Maximum	customer	customer	
Customer To Ext Source- Panama Addr	Match Customer Address To External Source - Panama Address	Maximum	customer	external_address_panama	
Customer To Derived Entity	Customer To Derived Entity	Maximum	customer	derived_entity	
Derived Entity To Derived Entity	Derived Entity To derived Entity Match	Maximum	derived_entity	derived_entity	
Customer To Ext Source- Offshore	Match Customer To External Source - Offshore	Maximum	customer	external_entity_offshore	
Customer To Ext Source- Bahamas	Match Customer To External Source - Bahamass	Maximum	customer	external_entity_bahamas	
Customer To Ext Source- Bahamas Addr	Match Customer Address To External Source - Bahamas Address	Maximum	customer	external_address_bahamas	
Customer To Ext Source- Paradise	Match Customer To External Source - Paradise	Maximum	customer	external_entity_paradise	

8.2 Creating Rulesets

To create a ruleset, follow these steps:

1. Navigate to the *Ruleset* page.
2. Click **Create**.
The *Ruleset Details* page is displayed.



Ruleset Details Back Reset

Name:

Description:

Scoring Aggregation Type: Maximum

Set Threshold:

Rules: Create

Source: Source Name

Target: Target Name

3. Enter the following details.

Field	Description
Name	Indicates the name of the ruleset.
Description	Indicates the additional description given for the ruleset.
Scoring Aggregation Type	Indicates the scoring aggregation method. Select one of the following options: <ul style="list-style-type: none"> Maximum: Considers the highest score obtained out of all the rules created for a ruleset. Minimum: Considers the lowest score obtained out of all the rules created for a ruleset.
Set Threshold	Indicates the threshold value set for a ruleset. A Similarity Edge is generated only when the maximum score obtained for a ruleset is equal to or higher than the threshold value.
Source	Indicates the source entity (node). The values are auto-populated from the metadata table that contains the elastic search index names generated as a result of running the Sqoop job.
Target	Indicates the target entity (node). The values are auto-populated from the metadata table that contains the elastic search index names generated as a result of running the Sqoop job.

8.3 Creating Rules in a Ruleset

To create rules in a ruleset, follow these steps:

1. Navigate to a *Ruleset Details* page.
2. Click **Create** to add a new rule.
A *New Rule* section is displayed.
3. Enter the following details:

Field	Description
Name	Indicates the name of the rule.
Description	Indicates the description of the rule.
Rule Threshold	Indicates the threshold value set for a rule. This rule contributes to the matching, only when the maximum score obtained for a rule is equal to or higher than the threshold value.

4. Click **Create** to add new Mappings:

Field	Description
Source Attribute	Indicates the source attribute.
Target Attribute	Indicates the target attribute.
Match Type	Indicates the match type. Select one of the following options: <ul style="list-style-type: none"> Exact: To obtain the matches that are 100% perfect when finding the entities in a database. Fuzzy: To obtain the matches that are less than 100% perfect when finding the entities in a database.
Scoring Method	The scoring methods used are as follows: <ul style="list-style-type: none"> Default Jaro Winkler For more information, see Scoring Method .
Threshold	Indicates that a score below the mentioned value does not generate a result from the elastic search.
Weightage	Indicates the weightage given for the attributes in the rule.
Condition	Indicates that this attribute cannot have a null value. This attribute must be populated and must return a value for the matching.

8.3.1 Scoring Method

The scoring methods used in the entity resolution component are as follows:

- Default Method

The distance is computed by finding the number of edits which transforms one string to another. The transformations allowed are as follows:

- Insertion: Adding a new character
- Deletion: Deleting a character
- Substitution: Replace one character with another

By performing these operations, the algorithm attempts to modify the first string to match the second one. The final result obtained is the edit distance.

For example:

```
a. textdistance.levenshtein('arrow', 'arow')
1
b. >> textdistance.levenshtein.normalized_similarity('arrow', 'arow')
0.8
```

Here, if you insert single 'r' in string 2, that is, 'arow', it becomes same as the string 1. Hence, the edit distance is 1. Similar with Hamming distance, you can generate a bounded similarity score between 0 and 1. The similarity score obtained is 80%.

- Jaro Winkler

This algorithm gives high scores for the following strings:

- a. The strings that contain same characters, but within a certain distance from one another.
- b. The order of the matching characters is same.

To be precise, the distance of finding similar character is one character less than half of the length of the longest string. So if the longest string has a length of five, a character at the start of the string 1 must be found before or on $((5/2)-1) \sim 2$ nd position in the string 2. This is considered a valid match. Hence, the algorithm is directional and gives high score if matching is from the beginning of the strings.

For example:

- a. `textdistance.jaro_winkler("mes", "messi")`
0.86
- b. `textdistance.jaro_winkler("crate", "crat")`
0.96
- c. `textdistance.jaro_winkler("crate", "atcr")`
0.0

In first case, as the strings are matching from the beginning, high score is given. Similarly, in the second case, only one character was missing and that too at the end of the string 2, hence a very high score is given. In third case, the last two character of string 2 are rearranged by bringing them at front and hence results in 0% similarity.

9 Configuring Interpreters

An interpreter reads and executes the instructions written in a programming or scripting language without compiling the high-level language code into a machine language program.

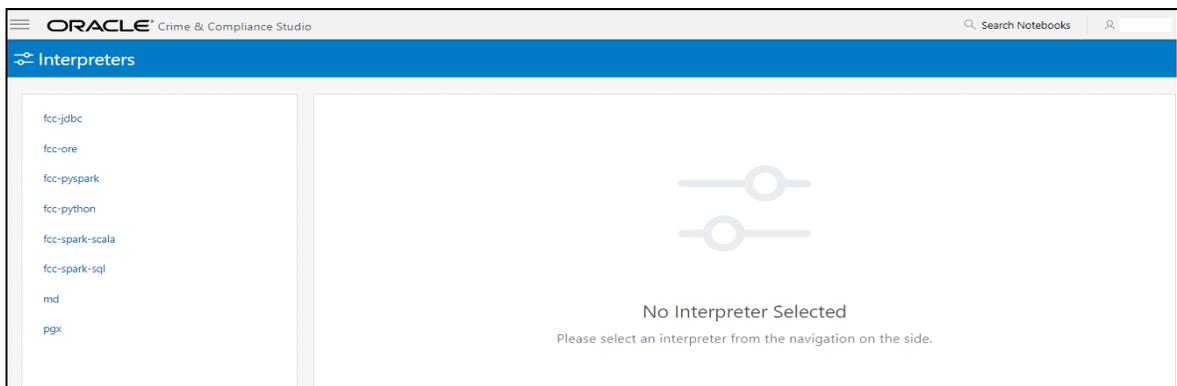
Topics:

- [Accessing Interpreters](#)
- [Creating a New Interpreter Variant](#)
- [Configure Interpreters](#)

9.1 Accessing Interpreters

To access interpreters, follow these steps:

1. Click the Navigation Menu  on the upper-left corner on the FCC Studio landing page.
The menu items are listed.
2. Click **Interpreters**.
The *Interpreters* page is displayed.



3. Click the interpreter that you want to access from the list displayed on the LHS.
The default interpreter variant configured is displayed on the RHS.
4. Modify the required values.
5. Click **Update**.
The modified values are updated in the interpreter.

9.2 Creating a New Interpreter Variant

In FCC 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.

- For a sample on creating a new interpreter variant, see [Creating a New fcc-jdbc Interpreter Variant](#)
- To enable a second Spark/PySpark interpreter, see [Enabling a Second Spark/PySpark Interpreter](#) chapter in the *OFS Crime and Compliance Studio Installation Guide (On-Premise)*.

9.2.1 Creating a New fcc-jdbc Interpreter Variant

To create a new fcc-jdbc interpreter variant, follow these steps:

1. Navigate to the *Interpreters* page.
2. Click the **fcc-jdbc** interpreter from the list displayed on the LHS.
The default interpreter variant is displayed on the RHS.
3. Click  **Add** to create a new variant for the selected interpreter:
The **Create Interpreter Variant** dialog box is displayed.
4. Enter the **Name** for the new interpreter variant.
5. Click **Create**.
A new variant is created with name, **<Interpreter Type>.<Variant Name>**.
6. Provide the new schema details such as the **default.url**, **default.user**, and **default.password**.
7. Click **Update**.
A new variant is created for the jdbc interpreter.
8. The Oracle Database schema that you have created must be granted with the same permissions that are granted to the BD or ECM atomic schema.
For more information, see the *Prerequisite Environmental Settings* section in the *OFS Crime and Compliance Studio Installation Guide*.
9. Run the following script after modifying the schema name with the newly created schema:


```
../OFS_FCCM_STUDIO/metaservice/model/SQLScripts/Atomic_Schema/FCC_JRS-DCN_CONTEXT_ATOMIC.sql
../OFS_FCCM_STUDIO/metaservice/model/SQLScripts/Atomic_Schema/PKG_FCC_STUDIO_JURN_VPD.sql
../OFS_FCCM_STUDIO/metaservice/model/SQLScripts/Atomic_Schema/PKG_FCC_STUDIO_JURN_VPD_BODY_ATOMIC.sql
```
10. For using the new interpreter variant in the notebook paragraphs, use the following format:


```
%fcc-jdbc.newVariant
<Your SQL query>
```
11. Configure the required values for the properties.
12. Click **Update**.
A new variant is created for the JDBC interpreter.

9.3 Configure Interpreters

The list of interpreters in FCC Studio are as follows:

- [fcc-jdbc Interpreter](#)
- [fcc-ore Interpreter](#)
- [fcc-pyspark Interpreter](#)
- [fcc-python Interpreter](#)

- [fcc-spark-scala Interpreter](#)
- [fcc-spark-sql Interpreter](#)
- [jdbc Interpreter](#)
- [md Interpreter](#)
- [pgql Interpreter](#)
- [pgx-algorithm Interpreter](#)
- [pgx-java Interpreter](#)
- [pyspark Interpreter](#)
- [spark Interpreter](#)

9.3.1 fcc-jdbc Interpreter

- [Configuring fcc-jdbc Interpreter Variant](#)
- [Linking Wallet Credentials to fcc-jdbc Interpreter](#)

9.3.1.1 Configuring fcc-jdbc Interpreter Variant

The configurations for the ofsa-jdbc interpreter are given as follows:

Table 1: fcc-jdbc Interpreter

Field	Description
pgx.baseUrl	Enter the PGX URL in this field. This is the location where the data is pushed. For example: <code>http://<HOSTNAME>:7007</code>
default.url	Enter the ofsa jdbc URL in this field. For example: <code>jdbc:mysql://localhost:5554/world</code> NOTE: If you want to use the Oracle wallet credentials, you must enter the alias name in the following format: <code>jdbc:oracle:thin:@<alias_name></code>
zeppelin.jdbc.principal	Enter the principal name to load from the keytab.
default.driver	Enter the default JDBC driver name. For example: <code>com.mysql.jdbc.Driver</code>
default.completer.ttlInSeconds	Enter the time to live sql completer in seconds.
default.password	Enter the default password. NOTE: This value can be null if you have entered the alias name in the <code>default.url</code> parameter for the fcc-jdbc interpreter.
default.splitQueries	This field indicates the presence of default split queries. Each query is executed apart and returns the result. Enter “true” or “false”.

Table 1: fcc-jdbc Interpreter

Field	Description
default.completer.schemaFilters	Enter a comma-separated schema filters to get metadata for completions.
ofsaa.sessionservice.url	Enter the session service URL in this field. For example: <code>http://<HOSTNAME>:7047/sessionservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-Studio will be installed.
default.user	Enter the name of the default user in this field. For example: root
zeppelin.jdbc.concurrent.max_connection	Enter the number of maximum connections allowed.
ofsaa.metaservice.url	Enter the metaservice URL in this field. For example: <code>http://<HOSTNAME>:7045/metaservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.
common.max_count	Enter the maximum number of SQL result to display.
zeppelin.jdbc.auth.type	Enter the default jdbc authentication type.
zeppelin.jdbc.precode	Enter the snippet of code that executes after the initialization of the interpreter.
zeppelin.jdbc.concurrent.use	Enter to enable or disable concurrent use of JDBC connections. Enter "true" or "false".
zeppelin.jdbc.keytab.location	Enter the keytab file location.

9.3.1.2 Linking Wallet Credentials to fcc-jdbc Interpreter

You can link the credentials (a wallet and a password) to fcc-jdbc interpreter variant to enable secure data access. This linking enables the fcc-jdbc interpreter to securely connect to the specified Oracle DB. For more information, see [Managing Credentials](#).

To link the wallet credentials to fcc-jdbc interpreter, follow these steps:

1. Configure the following parameters in the ofsaa-jdbc interpreter as follows:

Field	Description
default.url	If you want to use the Oracle wallet credentials, you must enter the alias name in the following format: <code>jdbc:oracle:thin:@<alias_name></code>
default.password	This value can be null if you have entered the alias name in the <code>default.url</code> parameter for the fcc-jdbc interpreter.

2. Configure the other parameters in the ofsaa-jdbc interpreter as given in [Configuring fcc-jdbc Interpreter Variant](#).

3. Link the wallet credentials to the fcc-jdbc interpreter variant. For more information, see [Linking Credentials to Interpreter Variants](#)

9.3.2 fcc-ore Interpreter

The configurations for the fcc-ore interpreter are given as follows:

Table 2: fcc-ore Interpreter

Field	Description
ore.sid	Enter the SID of DB server where the fcc-ore interpreter wants to connect.
rendering.row.limit	Indicates the number of rows to be shown in the fcc-ore interpreter output. For example: 1000
ore.conn_string	Enter the DB connection URL with which the fcc-ore interpreter can make the connection to the schema. This field can be left blank.
https_proxy	Enter the Proxy server using which connection to the internet can be established. For example: <code>www-proxy-hqdc.us.oracle.com:80</code>
ore.type	Enter the fcc-ore interpreter type as Oracle.
ore.password	Enter the schema password where the fcc-ore interpreter wants to connect.
libpath	Indicates the custom library path from where R packages will be installed via FCC Studio and will be added to R lib Path. Enter the path to be mentioned under the home directory where FCC Studio is installed. For example: If you want the packages to be available under <code>/home/user/library</code> , and FCC Studio is installed at <code>/home/user/datastudio</code> , then mention <code>/library</code> as the libpath.
ore.host	Enter the hostname of the DB server where the fcc-ore interpreter wants to connect.
rserve.password	Indicates the Rserve password.
rendering.numeric.format	Indicates the Number of digits to round off. For example: <code>%.2f</code>
ore.service_name	Enter the Service Name of DB server where the fcc-ore interpreter wants to connect.
rserve.try.wrap	Enter False.
rserve.host	Indicates the Rserve host.
repo_cran	Indicates the CRAN URL from where R libraries are downloaded to install R packages. For example: <code>https://cran.r-project.org/</code>

Table 2: fcc-ore Interpreter

Field	Description
ofsaa.sessionservice.url	Enter the session service URL in this field. For example: <code>http://<HOSTNAME>:7047/sessionservice</code> Here, <HOSTNAM> refers to the server name or IP where fcc-studio will be installed.
ore.all	Indicates all tables are synced to the fcc-ore interpreter. Enter the value as True.
rserve.plain.qap.disabled	Indicates whether plain QAP is disabled in the server or not. If disabled, the connection will always be attempted using SSL. For example: False
ore.user	Enter the schema name where the fcc-ore interpreter wants to connect.
http_proxy	Enter the Proxy server using which connection to the internet is established. This value is used to set the initial setting that makes the environment compatible to download the libraries available in R. For example: <code>www-proxy-hqdc.us.oracle.com:80</code>
rserve.port	Indicates the Rserve port.
rserve.secure.login	Enter TRUE to enforce secure login.
rendering.knitr.options	Enter the Knitr output rendering option. For example: <code>out.format = 'html', comment = NA, echo = FALSE, results = 'verbatim', message = F, warning = F, dpi = 300</code>
rserve.user	Indicates the Rserve username.
ore.port	Enter the port number of the DB server where the fcc-ore interpreter wants to connect.
ofsaa.metaservice.url	Enter the metaservice URL in this field. For example: <code>http://<HOSTNAME>:7045/metaservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.
rendering.include.row.name	Indicates whether to include row names. For example: false
rendering.knitr.image.width	Indicates the image width specification for ore output. For example: 60

9.3.3 fcc-pyspark Interpreter

The configurations for the fcc-pyspark interpreter are given as follows:

Table 3: fcc-pyspark Interpreter

Field	Interpreter
pgx.baseUrl	Enter the pgx.baseUrl URL in this field. This is the location where the data is pushed. For example: http://##HOSTNAME##:7007
livy.spark.executor.instances	Enter the number of executors to launch for the current session.
livy.spark.dynamicAllocation.cachedExecutorIdleTimeout	Enter the cached execution timeout in seconds.
zeppelin.livy.url	Enter the Livy URL in this field. Livy is an interface between Data Studio and Spark. For example: http://##HOSTNAME##:8998
zeppelin.livy.pull_status.interval.millis	Enter the data pull interval in milliseconds.
livy.spark.executor.memory	Enter the amount of memory to use for the executor process.
livy.spark.dynamicAllocation.enabled	This field indicates whether Dynamic Allocation is enabled or not. Enter “true” or “false”.
livy.spark.dynamicAllocation.minExecutors	Enter the minimum number of required Dynamic Allocation executors.
livy.spark.executor.cores	Enter the number of executor cores to use for the driver process.
zeppelin.livy.session.create_timeout	Enter the Zeppelin session creation timeout in seconds.
zeppelin.livy.spark.sql.maxResult	Enter the maximum number of results that must be fetched.
livy.spark.jars.packages	Enter to add extra libraries to a livy interpreter.
livy.spark.driver.cores	Enter the number of driver cores to use for the driver process.
zeppelin.livy.displayAppInfo	This field indicates whether the application information must be displayed or not. Enter “true” or “false”.
livy.spark.driver.memory	Enter the amount of memory to use for the driver process.
zeppelin.livy.principal	Enter the principal name to load from the keytab.
ofsa.sessionservice.url	Enter the session service URL in this field. For example: http://##HOSTNAME##:7047/sessionservice Here, ##HOSTNAME## refers to the server name or IP where fcc-studio will be installed.
ofsa.metaservice.url	Enter the metaservice URL in this field. For example: http://##HOSTNAME##:7045/metaservice Here, ##HOSTNAME## refers to the server name or IP where fcc-studio will be installed.
zeppelin.livy.keytab	Enter the keytab location.

Table 3: fcc-pyspark Interpreter

Field	Interpreter
livy.spark.dynamicAllocation.maxExecutors	Enter the maximum number of required Dynamic Allocation executors.

9.3.4 fcc-python Interpreter

Topics:

- [Configuring fcc-python Interpreter](#)
- [Changing Python Version in the fcc-python Interpreter](#)
- [Adding Python Packages to the fcc-python Interpreter](#)

9.3.4.1 Configuring fcc-python Interpreter

The configuration for the fcc-python interpreter is performed with the following fields:

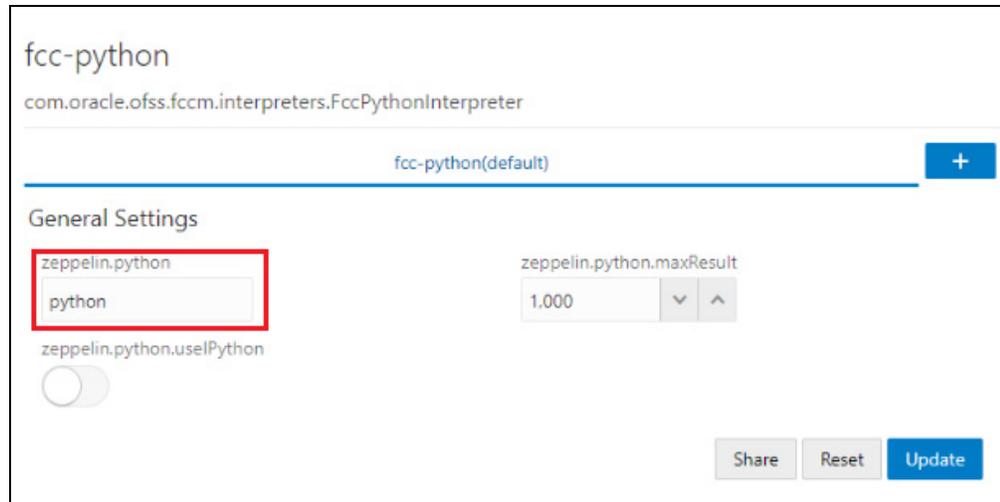
Table 4: fcc-python Interpreter

Field	Description
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 Changing Python Version in the fcc-python 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.

9.3.4.2 Changing Python Version in the fcc-python Interpreter

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

1. Navigate to the **fcc-python Interpreter Settings** page.
2. Perform one of the following:
 - Change the default Python version in the `zeppelin.python` parameter to the new version.
For example: `python3.6`



- Create a new interpreter variant and configure the version in the `zeppelin.python` parameter. For information on creating a new interpreter variant, see [Creating a New Interpreter Variant](#).

For example: To use Python 3.6, create a new fcc-python interpreter variant and enter the value as `python3.6`.

9.3.4.3 Adding Python Packages to the fcc-python Interpreter

To add desired Python packages to the fcc-python interpreter, follow these steps:

- For FCC Studio installed on-premise:
To install additional Python libraries to the fcc-python interpreter, contact System Administrator to install the required additional Python libraries on the Processing Server (Studio Notebook Server). The newly added Python libraries must be accessible to the Linux user for FCC Studio
- For FCC Studio installed using Kubernetes:
To install additional Python libraries to the fcc-python interpreter, see [Appendix - Modifying the Python Images for the Python Interpreter](#).

9.3.5 fcc-spark-scala Interpreter

The configurations for the fcc-spark-scala interpreter are given as follows:

Table 5: fcc-spark-scala Interpreter

Field	Description
<code>pgx.baseUrl</code>	Enter the <code>pgx.baseUrl</code> URL in this field. This is the location where the data is pushed. For example: <code>http://<HOSTNAME>:7007</code>
<code>livy.spark.executor.instances</code>	Enter the number of executors to launch for the current session.
<code>livy.spark.dynamicAllocation.cachedExecutorIdleTimeout</code>	Enter the cached execution timeout in seconds.

Table 5: fcc-spark-scala Interpreter

Field	Description
zeppelin.livy.url	Enter the Livy URL in this field. Livy is an interface between Data Studio and Spark. For example: <code>http://<HOSTNAME>:8998</code>
zeppelin.livy.pull_status.interval.millis	Enter the data pull interval in milliseconds.
livy.spark.executor.memory	Enter the amount of memory to use for the executor process.
livy.spark.dynamicAllocation.enabled	This field indicates whether Dynamic Allocation is enabled or not. Enter “true” or “false”.
livy.spark.dynamicAllocation.minExecutors	Enter the minimum number of required Dynamic Allocation executors.
livy.spark.executor.cores	Enter the number of executor cores to use for the driver process.
zeppelin.livy.session.create_timeout	Enter the Zeppelin session creation timeout in seconds.
zeppelin.livy.spark.sql.maxResult	Enter the maximum number of results that must be fetched.
livy.spark.jars.packages	Enter to add extra libraries to a livy interpreter.
livy.spark.driver.cores	Enter the number of driver cores to use for the driver process.
zeppelin.livy.displayAppInfo	This field indicates whether the application information must be displayed or not. Enter “true” or “false”.
livy.spark.driver.memory	Enter the amount of memory to use for the driver process.
zeppelin.livy.principal	Enter the principal name to load from the keytab.
ofsaa.sessionservice.url	Enter the session service URL in this field. For example: <code>http://<HOSTNAME>:7047/sessionservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.
ofsaa.metaservice.url	Enter the metaservice URL in this field. For example: <code>http://<HOSTNAME>:7045/metaservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.
zeppelin.livy.keytab	Enter the keytab location.
livy.spark.dynamicAllocation.maxExecutors	Enter the maximum number of required Dynamic Allocation executors.
livy.spark.dynamicAllocation.initialExecutors	Enter the initial Dynamic Allocation executors.

9.3.6 fcc-spark-sql Interpreter

The configurations for the fcc-spark-sql interpreter are given as follows:

Table 6: fcc-spark-sql Interpreter

Field	Description
pgx.baseUrl	Enter the pgx.baseUrl URL in this field. This is the location where the data is pushed. For example: <code>http://<HOSTNAME>:7007</code>
livy.spark.executor.instances	Enter the number of executors to launch for the current session.
livy.spark.dynamicAllocation.cachedExecutorIdleTimeout	Enter the cached execution timeout in seconds.
zeppelin.livy.url	Enter the Livy URL in this field. Livy is an interface between Data Studio and Spark. For example: <code>http://<HOSTNAME>:8998</code>
zeppelin.livy.pull_status.interval.millis	Enter the data pull interval in milliseconds.
livy.spark.executor.memory	Enter the amount of memory to use for the executor process.
livy.spark.dynamicAllocation.enabled	This field indicates whether Dynamic Allocation is enabled or not. Enter "true" or "false".
livy.spark.dynamicAllocation.minExecutors	Enter the minimum number of required Dynamic Allocation executors.
livy.spark.executor.cores	Enter the number of executor cores to use for the driver process.
zeppelin.livy.session.create_timeout	Enter the Zeppelin session creation timeout in seconds.
zeppelin.livy.spark.sql.maxResult	Enter the maximum number of results that must be fetched.
zeppelin.livy.spark.sql.field.truncate	Indicates to truncate field values longer than 20 characters or not. Enter "true" or "false".
livy.spark.jars.packages	Enter to add extra libraries to a livy interpreter.
livy.spark.driver.cores	Enter the number of driver cores to use for the driver process.
zeppelin.livy.displayAppInfo	This field indicates whether the application information must be displayed or not. Enter "true" or "false".
livy.spark.driver.memory	Enter the amount of memory to use for the driver process.
zeppelin.livy.principal	Enter the principal name to lead from the keytab.
ofsa.sessionservice.url	Enter the session service URL in this field. For example: <code>http://<HOSTNAME>:7047/sessionservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.
ofsa.metaservice.url	Enter the metaservice URL in this field. For example: <code>http://<HOSTNAME>:7045/metaservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.

Table 6: fcc-spark-sql Interpreter

Field	Description
zeppelin.livy.keytab	Enter the keytab location.
livy.spark.dynamicAllocation.maxExecutors	Enter the maximum number of required Dynamic Allocation executors.

9.3.7 jdbc Interpreter

Topics:

- [Configuring jdbc Interpreter Variant](#)
- [Linking Wallet Credentials to jdbc Interpreter](#)

9.3.7.1 Configuring jdbc Interpreter Variant

The configurations for the jdbc interpreter are given as follows:

Table 7: jdbc Interpreter

Field	Description
pgx.baseUrl	Enter the pgx.baseUrl URL in this field. This is the location where the data is pushed. For example: <code>http://<HOSTNAME>:7007</code>
default.url	Enter the jdbc URL in this field. NOTE: If you want to use the Oracle wallet credentials, you must enter the alias name in the following format: <code>jdbc:oracle:thin:@<alias_name></code>
zeppelin.jdbc.principal	Enter the principal name to load from the keytab.
default.driver	Enter the default JDBC driver name.
default.completer.ttlInSeconds	Enter the time to live sql completer in seconds.
default.password	Enter the default password. NOTE: This value can be null if you have entered the alias name in the <code>default.url</code> parameter for the jdbc interpreter.
default.splitQueries	This field indicates the presence of default split queries. Enter “true” or “false”.
default.completer.schemaFilters	Enter comma-separated schema filters to get metadata for completions.
ofsa.session.service.url	Enter the session service URL in this field. For example: <code>http://<HOSTNAME>:7047/session.service</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.

Table 7: jdbc Interpreter

Field	Description
default.user	Enter the name of the default user in this field.
zeppelin.jdbc.concurrent.max_connection	Enter the number of maximum connections allowed.
ofsaa.metaservice.url	Enter the metaservice URL in this field. For example: <code>http://<HOSTNAME>:7045/metaservice</code> Here, <HOSTNAME> refers to the server name or IP where fcc-studio will be installed.
common.max_count	Enter the maximum number of SQL result to display.
zeppelin.jdbc.auth.type	Enter the default jdbc authentication type.
zeppelin.jdbc.precode	Enter the snippet of code that executes after the initialization of the interpreter.
zeppelin.jdbc.concurrent.use	Enter to enable or disable concurrent use of JDBC connections. Enter "true" or "false".
zeppelin.jdbc.keytab.location	Enter the keytab location.

9.3.7.2 Linking Wallet Credentials to jdbc Interpreter

You can link the credentials (a wallet and a password) to jdbc interpreter variant to enable secure data access. This linking enables the jdbc interpreter to securely connect to the specified Oracle DB. For more information, see [Managing Credentials](#).

To link the wallet credentials to jdbc interpreter, follow these steps:

1. Configure the following parameters in the jdbc interpreter as follows:

Field	Description
default.url	If you want to use the Oracle wallet credentials, you must enter the alias name in the following format: <code>jdbc:oracle:thin:@<alias_name></code>
default.password	This value can be null if you have entered the alias name in the <code>default.url</code> parameter for the jdbc interpreter.

2. Configure the other parameters in the jdbc interpreter as given in [Configuring jdbc Interpreter Variant](#).
3. Link the wallet credentials to the jdbc interpreter variant. For more information, see [Linking Credentials to Interpreter Variants](#).

9.3.8 md Interpreter

The configurations for the md interpreter are given as follows:

Table 8: md Interpreter

Field	Description
markdown.parser.type	Enter the markdown parser type.

9.3.9 pgql Interpreter

The configurations for the pgql interpreter are given as follows:

Table 9: pgql Interpreter

Field	Description
graphviz.formatter.class	Indicates the class which implements the formatting of the visualization output. For example: <code>oracle.datastudio.graphviz.formatter.DataStudioFormatter</code>
graphviz.driver.class	The class which implements the PGQL driver. For example: <code>oracle.pgx.graphviz.driver.PgxDriver</code>
base_url	Enter the base URL of the PGX server in this field. For example: <code>http://<HOSTNAME>:7007</code>
zeppelin.interpreter.output.limit	Indicates that the output message from interpreter exceeding the limit will be truncated. For example: 102,400

9.3.10 pgx-algorithm Interpreter

The configurations for the pgx-algorithm interpreter are given as follows:

Table 10: pgx-algorithm Interpreter

Field	Description
graphviz.formatter.class	Indicates the class which implements the formatting of the visualization output. For example: <code>oracle.datastudio.graphviz.formatter.DataStudioFormatter</code>
graphviz.driver.class	The class which implements the PGQL driver. For example: <code>oracle.pgx.graphviz.driver.PgxDriver</code>
base_url	Enter the base URL of the PGX server in this field.

9.3.11 pgx-java Interpreter

The configurations for the pgx-java interpreter are given as follows:

Table 11: pgx-java Interpreter

Field	Description
graphviz.formatter.class	Indicates the class which implements the formatting of the visualization output. For example: <code>oracle.datastudio.graphviz.formatter.DataStudioFormatter</code>
graphviz.driver.class	The class which implements the PGQL driver. For example: <code>oracle.pgx.graphviz.driver.PgxDriver</code>
base_url	Enter the base URL of the PGX server in this field.
zeppelin.interpreter.output.limit	Indicates that the output message from interpreter exceeding the limit will be truncated. For example: 102,400

9.3.12 pyspark Interpreter

The configurations for the pgx-java interpreter are given as follows:

Table 12: pyspark Interpreter

Field	Description
zeppelin.pyspark.python	Indicates the Python binary executable to use for PySpark in both driver and workers. The default value is python. For example: <code>python</code>
zeppelin.pyspark.usePython	Set to true to use IPython, else set to false.

9.3.13 spark Interpreter

The configurations for the spark interpreter are given as follows:

Table 13: spark Interpreter

Field	Description
pgx.baseUrl	Enter the pgx.baseUrl URL in this field. This is the location where the data is pushed. For example: <code>http://<HOSTNAME>:7007</code>
spark.executor.memory	Enter the amount of memory to use for the executor process.

Table 13: spark Interpreter

Field	Description
spark.master	Enter the cluster manager to connect to. For example: <code>local[*]</code>
spark.yarn.archive	Enter the archive containing the required Spark jars for distribution to the YARN cache, to make Spark runtime jars accessible from the YARN side.
spark.app.name	Enter the name of the application. For example: <code>zeppelin</code>
zeppelin.spark.ui.hidden	Set to true or false.
zeppelin.spark.maxResult	Enter the maximum number of results that must be fetched.
spark.pyspark.python	Enter the Python binary executable to use for PySpark in both driver and executors. For example: <code>python</code>
zeppelin.spark.enableSupportedVersionCheck	Set to true or false.
args	Enter the Spark command-line args.
zeppelin.spark.useNew	Set to true to use the new version of the SparkInterpreter.
zeppelin.spark.useHiveContext	Set to true to use HiveContext instead of SQLContext.
zeppelin.spark.uiWebUrl	This value overrides the Spark UI default URL. Note: The value must be a complete URL.
zeppelin.spark.printREPLOutput	Indicates to print the REPL output.
spark.cores.max	Enter the total number of cores to use.

10 Configuring ETL

This chapter provides information on how to extract, transform, and load (ETL) information in FCC Studio. As part of running ETL, you can configure new data source for a graph. Also you can perform additional configuration for a graph. For information on performing graph configurations, see [Appendix - Performing Graph Configurations](#).

10.1 Performing Data Source Configuration

To configure a new data source for a graph, follow these steps:

1. Navigate to the `fcc_studio_etl_queries` table in the Studio Schema.
The FCDM related nodes and edges are available in the table.
2. If you want to add additional nodes or edges, you can add a new entry in the `fcc_studio_etl_queries` table.
3. Enter the following details in the `fcc_studio_etl_queries` table to add a new node or edge:

Table 1: fcc_studio_etl_queries Table Details

Column Name	Description	Applicable For
Type	Indicates the column name. Enter the value as NODE or EDGE.	Applicable for node and edge queries.
DF_NAME	Indicates the name for the node or edge.	Applicable for node and edge queries.
SOURCE	Indicates the source of the data. For example: FCDM or ICIJ	Applicable for node and edge queries.
DATAFRAME	Indicates the properties of the node or edge. NOTE: Enter this value only if the data source is Hive and not .a.csv file.	Applicable for node and edge queries.

Table 1: fcc_studio_etl_queries Table Details

Column Name	Description	Applicable For
QUERY	<ul style="list-style-type: none"> If the source is Hive, provide the Hive query. If the source is a .csv file, provide the query in the following format: <pre>spark.read.format("csv").option("header", "true").option("mode", "DROPMALFORMED").load("##FILE-PATH##").select("node_1", "node_2", "rel_type", "SourceID").withColumn("Label", lit("address of")).withColumnRenamed("node_1", "from").withColumnRenamed("node_2", "to").withColumnRenamed("rel_type", "EDGE_TYPE").withColumnRenamed("SourceID", "Source").filter(col("EDGE_TYPE")=="registered_address").withColumn("node_ID", concat(lit("#NUMBER #"), col("node_ID")))</pre> <p>For information on the Spark query parameters, see Appendix - Spark Query Parameters.</p> <p>NOTE: Ensure that the source .csv file is UTF-8 compatible.</p>	Applicable for node and edge queries.
KEY_COLUMN_NAME	Set the value to the column name of your unique identifier, if the query is for node. For example: 'node_id'.	Applicable for node query.
SOURCE_NODE	Provide the DF_NAME of the node from which the edge starts from.	Applicable for edge query.
DESTINATION_NODE	Provide the DF_NAME of the node from which the edge ends to.	Applicable for edge query.
SOURCE_KEY_COLUMN_NAME	Set the value to the column name which has key_column values of the Source Node. For example: 'from_id'	Applicable for edge query.
DESTINATION_KEY_COLUMN_NAME	Set the value to the column name which has key_column values of the Destination Node. For example: 'to_id'	Applicable for edge query.
ACTIVE	Expected values: 'Y'/'N'. Set the value to Y to consider ETL and Graph loading.	Applicable for node and edge queries.

- If the source is a .csv file, configure the file path in the `fcc_studio_etl_files` table.

NOTE Ensure that the source .csv file is UTF-8 compatible.

5. Enter the following details in the `fcc_studio_etl_files` table to add file path:

Table 2: fcc_studio_etl_files Table Details

Column Name	Description
DF_NAME	Indicates the name of the node or edge.
DF_SEQ_NO	Indicates the unique sequence ID for each file.
FILEPATH	Enter the path where the .csv files are stored. NOTE: If one data frame has multiple .csv files, then make separate entries for all the files. For example: see Figure fcc_studio_etl_files Table .
FILEORDER	If data must be imported from multiple files, specify the order in which the files must be read.

Figure 1: fcc_studio_etl_files Table

	DF_NAME	FILEPATH	DF_SEQ_NO	FILE_ORDER
1	Offshore_edges_is_related_to	12	1
2	Bahama_External_Address	13	1

11 Managing FCC Studio Batches

This chapter provides information on creating batches required for FCC Studio. Batches enable us to load graphs, run notebooks, and move data from Oracle Database or Big Data to FCC Studio.

Before running batches, you can configure new data sources to a graph as part of running ETL. For more information on configuring ETL, see [Configuring ETL](#).

Topics:

- [Preparing for Batches](#)
- [Performing Batches](#)

11.1 Preparing for Batches

Follow these steps to prepare the batches:

1. Copy all the jars from the `<STUDIO_INSTALLATION_PATH>/ficdb/lib` directory to the `<FIC_HOME of OFSAA_Installed_Path>/ficdb/lib` directory.
2. Copy the `NBExecutor.txt` file from the `<STUDIO_INSTALLATION_PATH>/ficdb/bin` directory to the `<FIC_HOME of OFSAA_Installed_Path>/ficdb/bin` directory.
3. Navigate to the `<Studio_Installation_Path>/ficdb/bin` directory.
4. Run the `FCCM_Studio_Set_UserPass.sh` command as follows:

- `FCCM_Studio_Set_UserPass.sh --username "Username" --password "Password"`
or
- `FCCM_Studio_Set_UserPass.sh -u "USERNAME" -p "PASSWORD"`

The `FCC_Studio_SecretKey.properties` and `NBExecutor.txt` files are created in the `<Studio_Installation_Path>/ficdb/conf` directory.

NOTE

1. Ensure that the `FCC_Studio_SecretKey.properties` and `NBExecutor.txt` files are present in the `<Studio_Installation_Path>/ficdb/conf` directory, before executing a notebook batch.
2. If only the `NBExecutor.txt` file is present in the `<Studio_Installation_Path>/ficdb/conf` directory, then re-execute the `FCCM_Studio_Set_UserPass.sh` command with username and password to create a new `FCC_Studio_SecretKey.properties` file and updated `NBExecutor.txt` file.

11.2 Performing Batches

The following type of batches are configured in FCC Studio.

- [Data Movement and Graph Loading for Big Data Environment](#)
- [Executing Published NotebookData Movement and Graph Loading for Big Data Environment](#)

11.2.1 Data Movement and Graph Loading for Big Data Environment

You can move data from the *BD or ECM Atomic* environment to the *Big Data* environment. As a result, a graph is created in the form of the `.csv` and `graph.json` files. These files can be used to view and query the PGQL and PGX graphs. For information on the Oracle Financial Crime Graph Model, see [Oracle Financial Crime Graph Model](#).

1. To create batches for data movement and graph loading, perform the following:
 - For FCC Studio with OFSAA: [Appendix - Creating and Executing Run Executable](#)
 - For FCC Studio with non-OFSAA: [Executing Sqoop Job](#)
2. Verify indices in elastic search. For more information, see [Verifying Indices in Elastic Search](#).

11.2.1.1 Executing Sqoop Job

NOTE This section is applicable for FCC Studio with non-OFSAA.

To execute the Sqoop job for data movement and graph loading for FCC Studio without OFSAA, follow these steps:

1. Execute the `FCCM_Studio_SqoopJob.sh` command with the required parameters as follows:

```
./FCCM_Studio_SqoopJob.sh <Batch_Name> <Batch_ID> EXEC <FROM_FIC_MIS_-
DATE> <TO_FIC_MIS_DATE> SNAPSHOT_DT=<SNAPSHOT_DATE>,DATAMOVEMENTCODE=ALL
<SOURCE FOR DATAMOVEMENT> <SOURCE TO INCLUDE IN GRAPH>
```

For example:

```
./FCCM_Studio_SqoopJob.sh A B C 20150618 20190618 SNAP-
SHOT_DT=20181219,DATAMOVEMENTCODE=ALL FCDM FCDM,ICIJ
```

Where,

- `<Batch_Name>` is A
- `<Batch_ID>` is B
- `EXEC` is C
- `<FROM_FIC_MIS_DATE>` is 20150618
- `<TO_FIC_MIS_DATE>` is 20190618
- `<SNAPSHOT_DATE>` is 20181219
- `DATAMOVEMENTCODE` is ALL
- `<SOURCE FOR DATAMOVEMENT>` is FCDM
- `<SOURCE TO INCLUDE IN GRAPH>` is FCDM,ICIJ.

Ensure that the values are comma-separated.

11.2.1.2 Verifying Indices in Elastic Search

To verify indices in elastic search, follow these steps:

1. Enter the URL in the following format into the browser:

```
http://<Elastic_Search_Hostname>:<Elastic_Search_Port>/_cat/indices
```

All the indices must be displayed with the same snapshot date with which the job is triggered with.

Format: `<Index name>_<Snapshot Date>`

For example:

- fcdm_customer_2020-03-01
- icij_bahama_external_address_2020-03-01

11.2.2 Executing Published Notebook

The published notebook can be scheduled for execution with a set of threshold values required for generating alerts or trends.

To execute the published notebook, follow these steps:

- For a published scenario notebook:
 - For FCC Studio with OFSAA, [Appendix - Creating and Executing Run Executable](#)
 - For FCC Studio with non-OFSAA, [Executing Published Scenario Notebook for FCC Studio with Non-OFSAA](#)
- For a published non-scenario notebook:
 - For FCC Studio with OFSAA, [Appendix - Creating and Executing Run Executable](#)
 - For FCC Studio with non-OFSAA, [Executing Published Non-Scenario Notebook for FCC Studio with Non-OFSAA](#)

11.2.2.1 Executing Published Scenario Notebook for FCC Studio with Non-OFSAA

NOTE

- This section is applicable for FCC Studio with non-OFSAA.
- Ensure that the username and password are set before executing a notebook batch. For more information, see [Preparing for Batches](#).

Execute the following command with the required parameters as follows:

```
./FCCM_Studio_NotebookExecution.sh "notebookID" "null" "scenarioID"
"thresholdsetID" "null" "BATCH_ID"
```

11.2.2.2 Executing Published Non-Scenario Notebook for FCC Studio with Non-OFSAA

NOTE

This section is applicable for FCC Studio with non-OFSAA.

To execute a non-scenario notebook for FCC Studio with non-ofsaa, follow these steps:

1. Execute the following command:

```
./FCCM_Studio_NotebookExecution.sh "notebookID" "null" "null" "null"
"paramkey1~~value1@@paramkey2~~value2"
```

For example, If

- paramkey1 is ficmisdate
- paramkey2 is lookbackperiod
- value1 is 20-09-2018
- value2 is 30

Then the extraparams must be written as follows:

ficmisdate~~20-09-2018@@lookbackperiod~~30

12

Appendix - Creating and Executing Run Executable

NOTE

Ensure that the username and password are set before executing a notebook batch. For more information, see [Preparing for Batches](#).

To create and execute run executable, follow these steps:

1. Log in to the OFSAA application with a user who has the privilege to create run executable.
2. Select **Financial Services Anti Money Laundering** from the tiles menu.

The Financial Services Anti Money Laundering Application Home Page is displayed with the Navigation list to the left.

3. Navigate to **Common Tasks > Rule Run Framework > Run** from the navigation list.

The **Run Definition** page is displayed.

4. Click **New** on the List toolbar.

The **Rule Run Framework** window is displayed.

5. Under the **Linked To** toolbar, click the button next to **Directory**.

The **Folder Selector** dialog box is displayed.



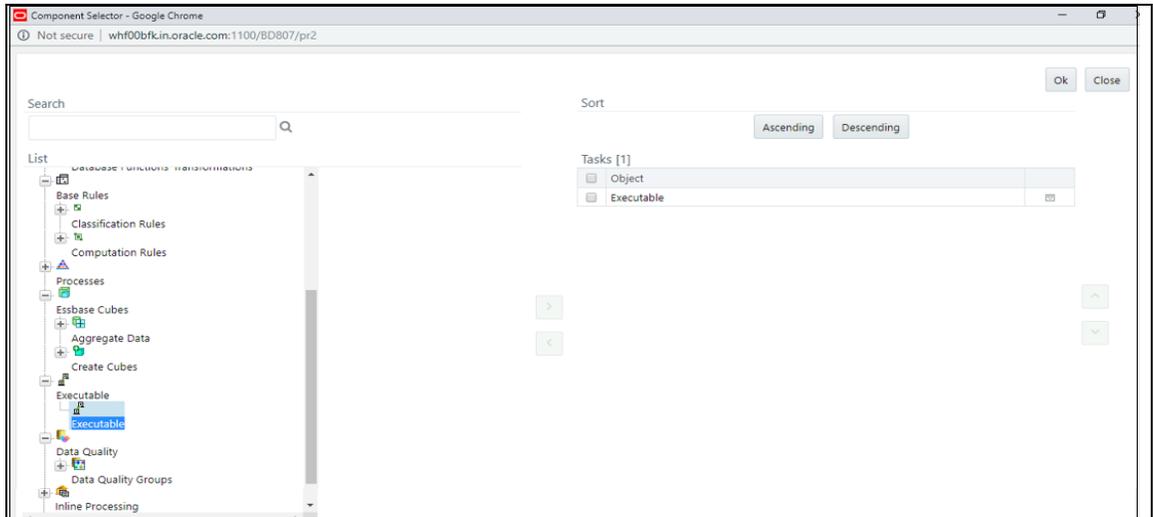
6. Select the directory that is to be linked to the run executable.
7. Click **OK**.
8. Enter the following details in the **Master Information** toolbar.

Table 1: Adding Run Definition

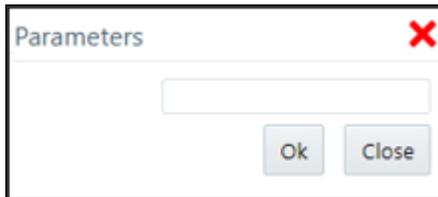
Field	Description
Code	Enter the Code of the process.
Name	Enter the Name of the process.
Type	Select Type for the process.

9. Click **OK**.

10. Click **Selector** on the List toolbar. From the options displayed, select **Job**.
The **Jobs** page is displayed.
11. Click **Executable** on the list. From the options displayed, select **Executable**.
The **Executable** gets displayed on the right.



12. Select **Executable** from the Tasks list, click the button next to the Executable option.



The **Parameters** dialog box is displayed.

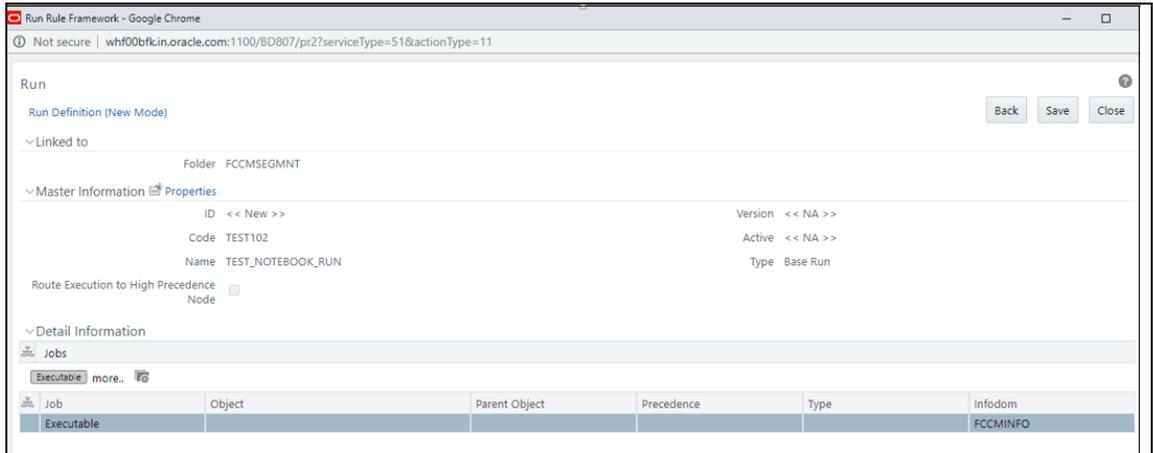
13. Enter the parameters in the following format to create run executable:

Table 2: List of Parameters for Batch Execution

SL. No.	Command Format		Description
1	"FCCM_Studio_SqoopJob.sh", "batchId", "ficmisdate", "userparams", "source for datamovement", "source to include in graph"		Use this command for data movement and graph loading.
	File Name	Parameters	Description
	FCCM_Studio_SqoopJob.sh		Used for data movement and graph loading in Big Data Environment
		batchId	Indicates the ID of the batch that you want to execute.
		ficmisdate	Indicates the date on which the source data is available.
		userparams	Includes the data movement code and snapshot date.
		source for datamovement	Indicates the source data type for data movement.
		source to include in the graph	Indicates the source to be included in the graph. The values must be comma-separated.
2	"FCCM_Studio_NotebookExecution.sh", "notebookID", "outputParagraphID", "scenarioID", "thresholdsetID", "extraparams"		Use this command for executing a published notebook.
	File Name	Parameters	Description
	FCCM_Studio_NotebookExecution.sh		Used for batch execution of published notebook.
		notebookId	Indicates the ID of the required notebook
		outputParagraphId	Indicates that the value is always "null"
		scenarioId	Indicates the ID of Scenario
		thresholdsetId	Indicates the ID of the threshold set with which notebook will run
		sessionId	Indicates the ID of the session in which notebook will run
		extraparams	For scenario notebook, it will be "null", but for notebook execution, it depends on the paramkeys used in the notebook

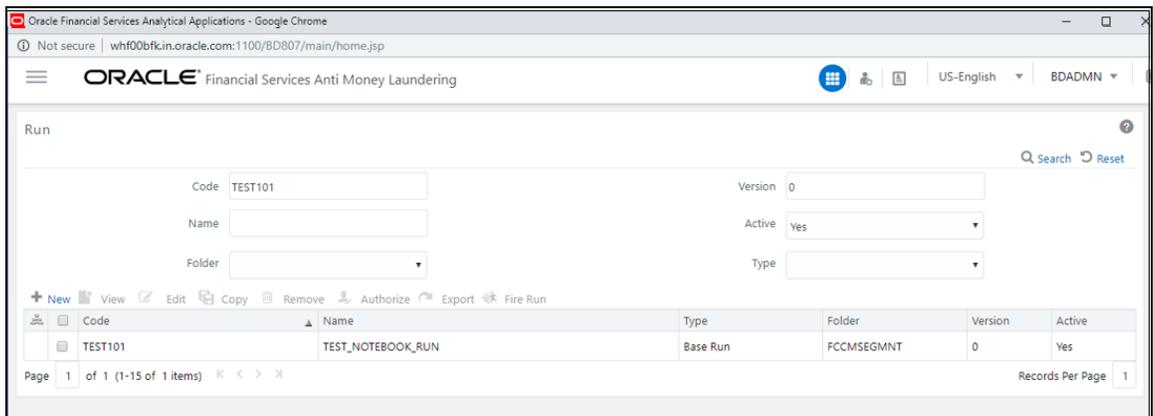
14. Click **OK**.

The run executable is displayed in the Detail Information section on the **Run Definition** page.



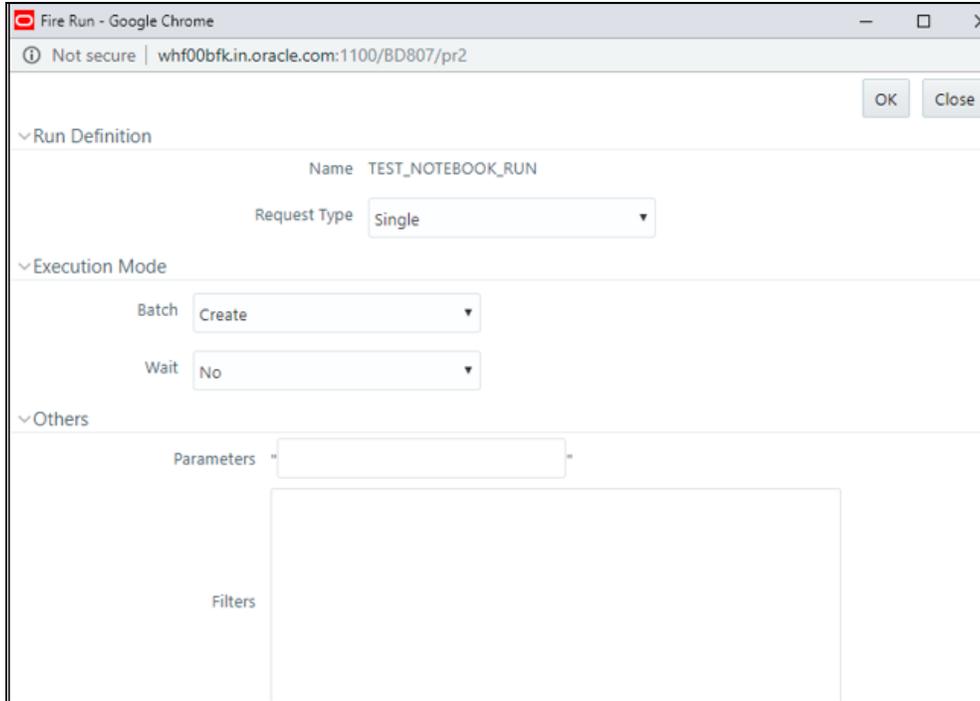
15. Click **Save**.

A confirmation message is displayed. The Run executable is created.



16. Select the newly created run executable from the **Run Definition** page that is to be created and click **Fire Run**.

The **Fire Run Rule Framework** dialog box is displayed.



17. Enter the following details:

Table 3: Adding Fire Run Details

Field	Description
Request Type	Select Request Type based on the following options: <ul style="list-style-type: none"> • Single: If the batch has to be executed once. • Multiple: If the batch has to be executed multiple times at different intervals.
Batch	Select Batch. It has the following options: <ul style="list-style-type: none"> • Create • Create and Execute From these options, select Create & Execute.
Wait	Select Wait. It has the following options: <ul style="list-style-type: none"> • Yes: This will execute the batch after a certain duration. Enter the duration as required. • No: This will execute the batch immediately.
Filters	Enter the filter details.

18. Click **OK** to run the Run Executable.

The Run executable starts executing.

19. From the Navigation List, navigate to **Common Tasks**, click **Operations**, and then select **Batch Monitor**.

The **Batch Monitor** window is displayed.

20. Select the batch that is run in step 18. Select the Information Date and Batch Run ID from the drop-down.
21. Click on **Start Monitoring** in Batch Run Details.
The Batch Run ID and Batch Status details are displayed in the Batch Status details section.

13 Appendix - Modifying the Python Images for the Python Interpreter

To modify the Python packages or change the Python version, you must modify the Python image.

NOTE

- This section is applicable for FCC Studio installed using Kubernetes.
- This section can be used as an example to understand the steps involved to modify the Python images for the Python interpreter in FCC Studio.

Topics:

- [Preparing for Building a Python Image](#)
- [Building and Pushing an Image](#)
- [Replacing Python Image in FCC Studio](#)

13.1 Preparing for Building a Python Image

For preparing to build Python images, you can either modify the Python packages in the Python interpreter or add different versions of Python to Python interpreter.

Topics:

- [Modifying the Python Packages in the Python Interpreter](#)
- [Adding Different Version of Python to Python Interpreter](#)

13.1.1 Modifying the Python Packages in the Python Interpreter

To modify the Python packages in the Python interpreter in FCC Studio, follow these steps:

NOTE

This process adds the Python packages to Python 3.6.

1. Create a directory in the same location as the `<Studio_Installation_Path>`, and create a file inside the directory as `Dockerfile`.
2. Copy and paste the following content as a template into the `Dockerfile` file:

```
FROM ofsaa-fccm-docker-release-local.dockerhub-den.oraclecorp.com/fcc-studio/fcc-python:8.0.7.4.0

RUN pip3 --no-cache-dir install scipy pandas cx_oracle --user
```
3. Modify the `Dockerfile` file depending on the following installation method:
 - a. If Internet connectivity is available, follow these steps:

Depending on the version of the Python package, install the `scipy` and `cx_oracle` Python packages using the following command:

```
RUN pip install scipy cx_oracle
```
 - b. If Internet connectivity is unavailable, follow these steps:
 - i. Download the Python package files.

- ii. Create a directory beside the `Dockerfile` file.
For example: `packages`
- iii. Place the downloaded files in the newly created `packages` directory.
- iv. Modify the `Dockerfile` using the following commands:

For example, to install using Python3:

```
COPY packages /var/olds-python-interpreter/packages
```

```
RUN cd /var/olds-python-interpreter/packages
```

```
RUN pip3 --no-cache-dir install --no-deps numpy-1.17.4-cp36-cp36m-manylinux1_x-86_64.whl
```

NOTE

For more information on how to write `Dockerfile`, see <https://docs.docker.com/engine/reference/builder/>.

4. Build and push the image. For more information, see [Building and Pushing an Image](#)
5. Replace the Python image in FCC Studio. For more information, see [Replacing Python Image in FCC Studio](#)

13.1.2 Adding Different Version of Python to Python Interpreter

To add a different version of Python to Python interpreter in FCC Studio, follow these steps:

1. Create a directory at the same location as the `<Studio_Installation_Path>`, and create a file inside the directory with the file name, `Dockerfile`.
2. Copy and paste the following content as a template into the `Dockerfile` file:

```
FROM ofsaa-fccm-docker-release-local.dockerhub-den.oraclecorp.com/fccstudio/fcc-python:8.0.7.4.0
```

```
USER root
```

```
RUN yum install -y python3.5
```

```
USER interpreteruser
```

```
RUN python3.5 -m pip install scipy pandas cx_oracle --user
```

3. Modify the `Dockerfile` file based on the preferred way of installing Python in the RHEL server.
For more information on how to modify the `Dockerfile`, see <https://docs.docker.com/engine/reference/builder/>.
4. Build and push the image. For more information, see [Building and Pushing an Image](#)
5. Replace the Python image in FCC Studio. For more information, see [Replacing Python Image in FCC Studio](#)

13.2 Building and Pushing an Image

To build and push an image, follow these steps:

1. Build the docker image using the following command:

```
docker build . --build-arg http_proxy=http://<proxy-url>:<port> --build-arg https_proxy=http://<proxy-url>:<port> -t <my.docker-
```

```
registry.com:port>/ofsaa-fccm-docker-release-local.dockerhub-  
den.oraclecorp.com/fcc-studio/fcc-python:<version>
```

Where:

- `<my.docker-registry.com:port>` is the docker-registry url with port number.
- `<version>` is the custom tag for this image.

For example:

```
docker build . --build-arg http_proxy=http://my-proxy-url:80 --build-arg  
https_proxy=http://my-proxy-url:80 -t my.docker-registry.com:5000/ofsaa-  
fccm-docker-release-local.dockerhub-den.oraclecorp.com/fcc-studio/fcc-  
python:8.0.7.4.0-C1
```

NOTE `build-arg` can be skipped if proxy is not required or if packages are placed locally.

2. Push the images using the following command:

```
docker push <my.docker-registry.com:port>/ofsaa-fccm-docker-  
releaselocal.dockerhub-den.oraclecorp.com/fcc-studio/fcc-  
python:<version>
```

13.3 Replacing Python Image in FCC Studio

To replace the Python images in FCC Studio, follow these steps:

1. Navigate to the `<Studio_Installation_path>/deployments/` directory.
2. Update the image name in the `fcc-python.yml` file as follows:

```
spec:  
  spec:  
    containers:  
      - name: python-interpretter  
        image: ofsaa-fccm-docker-release-local.dockerhub-  
den.oraclecorp.com/fcc-studio/fcc-python:<version>
```

3. Follow these steps to restart the FCC Studio application.
 - a. Execute the following command from the Kubernetes master node:

```
kubectl delete namespace <Namespace>
```

- b. Navigate to the `<Studio_Installation_Path>/bin` directory.
- c. Execute the following command:

```
./fcc-studio.sh --registry <registry URL>:<registry port>
```

14 Appendix - Spark Query Parameters

This chapter provide information on the Spark query parameters that is used to during configuring a new data source for a graph.

Table 1: Spark Query Parameter Details

Query Parameter	Description
<code>spark.read.format("csv")</code>	Indicates the input file format. For example: .csv.
<code>option("header", "true")</code>	Indicates the presence of a header in the input file. <ul style="list-style-type: none"> • true indicates that the header is available in the input file. • false indicates that the header is absent in the input file.
<code>load("Path").</code>	<ul style="list-style-type: none"> • Load indicates to load the data from the mentioned file path. • Path indicates the path where the files are placed. You can load to multiple paths using the following format: ("Path1", "Path2", ...)
<code>select("Col1", "Col2", "Col3", "Col4")</code>	Indicates the columns to be selected in the input file.
<code>withColumn("A", lit("Test1"))</code>	Indicates to add a new column with column name A and column value Test1.
<code>withColumnRenamed("A", "B")</code>	Indicates to rename a column with a different name. For example: rename column from A to B.
<code>filter(col("A")==="Test1")</code>	Indicates the "Where" filter condition. Here, the value for column A is Test1.
<code>withColumn("B", concat(lit("Test1"), col("A")))</code>	Indicates to add a new column B, whose value is the concatenated value of Test1 and column A. For example: Test1=ABC Column A contains Country and Pincode as the column values. Column B gets ABCCountry and ABCPincode as column values.

15 Appendix - Performing Graph Configurations

This chapter provides information on the additional configurations that can be performed on a graph.

Topics:

- [Attributes Case in Graph](#)
- [Extra Empty Nodes and Edges Providers](#)
- [Additional Configuration](#)

15.1 Attributes Case in Graph

The rules applied to the attribute names are as follows:

1. The Attribute name is split based on '_'. The '_' acts as a separator and the text that is separated by the separator are split into words.
2. The first alphabet of each word is set to uppercase and the remaining alphabets are set to lower-case.

For example:

- Attribute Name: sample_attribute
- Renamed Attribute Name: Sample Attribute

If any specific attribute name is required in the graph, you must update the same in the `FCC_GRAPH_COLUMN_NAME_MAPPING` table:

Where,

`COLUMN_NAME`: Indicates the attributes name in queries

`RENAMED_COLUMN_NAME`: Indicates the required attribute name

`COLUMN_DATA_TYPE`: Indicates the PGX's datatype of the attribute

NOTE

- The accepted PGX's datatype formats are boolean, integer, float, long, double, string, date, local_date, time, time-stamp, time_with_timezone, timestamp_with_timezone, and point2d.
- The date is deprecated, hence you can use one of following instead:
 - local_date
 - time
 - timestamp
 - time_with_timezone
 - timestamp_with_timezone

For example, if the values are as follows:

- `COLUMN_NAME`: sample_attribute
- `RENAMED_COLUMN_NAME`: Sample_AttributeName
- `COLUMN_DATA_TYPE`: string

Then the attribute name shown in the graph is, Sample_AttributeName

Figure 1: FCC_GRAPH_COLUMN_NAME_MAPPING Table

	❖ COLUMN_NAME	❖ RENAMED_COLUMN_NAME	❖ COLUMN_DATA_TYPE
1	original_id	Original ID	string
2	tax_id	Tax ID	string
3	debit_or_credit	Debit or Credit	string
4	initialShowPropName	initialShowPropName	string

15.2 Extra Empty Nodes and Edges Providers

Currently, the heterogeneous graph doesn't support the dynamic addition of Nodes and Edges Provider in the graph. If extra nodes or edge providers are required, then you must add the entries to the FCC_GRAPH_EMPTY_ENTITY_MAPPING table.

Where,

- **TYPE:** Indicates the type of empty entity provider to be added.
Expected value: "NODE" or "EDGE"
- **NAME:** Indicates the name of the entity provider.
- **COLUMN_MAPPING:** Indicates the attributes required for the entity with its datatype. The value must be a comma-separated paired value of column name and its type.

For example: column1:string,column2:long

NOTE

- In case of NODE, do not specify key_column for the node. In case of EDGE, do not specify source and destination key_columns.
- The accepted PGX's datatype formats are boolean, integer, float, long, double, string, date, local_date, time, timestamp, time_with_timezone, timestamp_with_timezone, and point2d.
- The date is deprecated, hence you can use one of following instead:
 - local_date
 - time
 - timestamp
 - time_with_timezone
 - timestamp_with_timezone

- Example 1:
 - **TYPE:** NODE
 - **NAME:** extra_node
 - **COLUMN_MAPPING:** name:string,phone_number:integer

Here, an extra vertex provider with the name "extra_node" is added with the attributes, Name and Phone Number, datatype, string, and integer respectively.

- Example 2:

- TYPE: EDGE
- NAME: extra_edge
- COLUMN_MAPPING: name:string,risk:long,edge_type:string

Here, extra edges will be formed between every node provider including itself with the name as "<source_node_provider>_extra_edge_<destination_node_provider>", with the attributes, Name, Risk and Edge Type, datatype, string, long, and string respectively.

Figure 2: FCC_GRAPH_EMPTY_ENTITY_MAPPING Table

ID	NAME	TYPE	COLUMN_MAPPING
1	searched_entity	NODE	source:string,label:string,name:string,address:string,tax_id:string,date:string,initialShowPropName:string
2	is_similar_to	EDGE	label:string,match_weight:float,match_score:string

15.3 Additional Configuration

For the Out- of-the-box (OOB) graph's configuration, the following parameters are set in the FCC_DATASTUDIO_CONFIG table:

1. local_date_format:

The default value: ["M/d/yyyy", "yyyy-MM-dd"].

NOTE

- Include '[', ']', "" and ','.
- The date format option can be used only to view the date type of an attribute on the graph, in the configured format.

2. vertex_id_type:

The default value is "long" as per the OOB queries.

This parameter represents the datatype of the vertex_id column or key_column of node providers.

NOTE

This datatype should be consistent across all nodes.

Figure 3: FCC_DATASTUDIO_CONFIG Table

ID	PARAMNAME	PARAMVALUE	DESCRIPTION
1	VERTEX_ID_TYPE	long	(null)
2	LOCAL_DATE_FORMAT	["M/d/yyyy", "yyyy-MM-dd"],	(null)

OFSAA Support Contact Details

Raise a Service Request (SR) in [My Oracle Support \(MOS\)](#) for queries related to the OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.

