Oracle Financial Services Investigation Hub

Administration and Configuration Guide

Release 8.1.2.6.0

April 2024

F49104-01

ORACLE
Financial Services

OFS Investigation Hub Administration and Configuration Guide

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

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

Document Control

Table 1 lists the document control of this guide.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.2.6.0	April 2024	Added the Authenticating the User to Access Investigation Tab in ECM section. The attention note is added in the Enable Investigation Hub Tab in ECM Case Designer section.
8.1.2.6.0	February 2024	Added the new “ECM Case Narrative” notebook in the Administration and Configuration Activities section. Added the following sections: ● Configuring the Investigation Hub Parameters ● Customizing Investigation Hub Notebook ● Integrating Investigation Hub with ECM ● Frequently Asked Questions Removed the “PGX Data Memory Limit” section as it is functionally deprecated. Removed the following sections and moved to the OFS Investigation Hub Installation Guide: ● Access the Notebook ● Creating the Encrypted Password ● Executing the Notebook
8.1.2.4.0	November 2023	Updated a note in the Managing User Administration section.
8.1.2.4.0	July 2023	Updated the investigation score in the Configuring Notebook Parameters section.
8.1.2.4.0	June 2023	Deprecated L2 and Generate Correlation Networks notebooks. Updated note in the Administration and Configuration Activities section. Removed the following sections: ● Generating Correlation Networks ● Creating a New CSV ● Loading Data to Graphs
8.1.2.1.0	December 2022	● OJET Upgrade (all UI elements are updated according to UI in the entire document).
8.1.2.0.0	April 2022	Added the new “PGX Data Memory Limit” section.
8.1.1.1.0	December 2021	There is no content update from the previous version. Only version number has been changed.

Table 1: Document Control

Version Number	Revision Date	Change Log
8.1.1.0.0	October 2021	This is the first version created for IH 8.1.1.0.0 release based on OFS Compliance Studio 8.1.1.0.0 release.

Table of Contents

1 Preface

This guide describes the physical and logical architecture of the Oracle Financial Services Investigation Hub (OFS IH) application. It also provides instructions for maintaining and configuring OFS IH, its subsystem components, and any third-party software required for operations.

Topics:

- Summary
- Documentation Accessibility
- Audience
- Related Documents
- Conventions
- Abbreviations

1.1 Summary

You can find the latest copy of this document in the Oracle Help Center (OHC) Documentation Library which includes all the recent additions/revisions (if any) done to date.

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website.

1.3 Audience

The Oracle Financial Services Investigation Hub Administration and Configuration Guide is intended for System Administrator and Implementation Consultant.

1.4 Related Documents

This section identifies additional documents related to the OFS IH application. Oracle Financial Services Analytical Applications Infrastructure Related Documents.

The following OFS IH documents are available in Oracle Help Center Documentation Library:

- OFS Investigation Hub Installation Guide
- OFS Investigation Hub User Guide
- OFS Investigation Hub Release Notes

1.5 Conventions

Table 2 lists the conventions used in this document.

Table 2: Conventions Used in This Guide

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

1.6 Abbreviations

Table 3 lists the abbreviations used in this document.

Table 3: Abbreviations Used in This Guide

Abbreviation	Meaning
OFS	Oracle Financial Services
AAI	Analytical Applications Infrastructure
PGX	Parallel Graph Analytics
PGQL	Property Graph Query Language
LHS	Left Hand Side
OFSAA	Oracle Financial Services Analytical Applications
OFS IH	Oracle Financial Services Investigation Hub
FCGM	Financial Crime Graph Model
FCDM	Financial Crime Data Model
SQL	Structured Query Language
IH	Investigation Hub
ECM	Enterprise Case Management
AML	Anti-money Laundering
BD	Behavior Detection
OOB	Out-of-the-Box

2 About Oracle Financial Services Investigation Hub

This chapter provides a brief overview of the OFS Investigation Hub application.

Topics:

- Introduction
- Import Notebooks
- Administration and Configuration Activities

2.1 Introduction

OFS Investigation Hub is an application built on OFS Compliance Studio that allows investigators to rapidly view enhanced cases within OFS Enterprise Case Management, both with and without a graphical view of the case.

The narrative (non-graph) Investigation Hub notebooks provide analysts with a pre-defined natural language narrative of the case and its entities, as well as highlighting risk factors and red flags within the case to enhance decision making.

The graphical Investigation Hub notebooks additionally have the case graph loaded, allowing for the graph's expansion and further investigation of the entities relationships by expanding the graph.

The notebooks are built on the Financial Crime Graph Model Schema which is configurable within OFS Compliance Studio and optionally provides the capability for matching to third-party sources of data like ICIJ and well as linking internal similar internal parties. OFS Investigation Hub accelerates investigations by bringing relevant information sources together (including external API calls to sources such as Quantifind), and preventing the need for the manual collation of information from disparate sources (where data is not available in OFS Enterprise Case Management). OFS Investigation Hub automatically generates case narratives and insights, highlights risk factors, and red flags that are meaningful to the investigation, and recommends actions based on the scoring algorithms of required.

2.1.1 Key Features

- Pre-built notebooks for case investigation and special investigation
- Configurable red flags and risk factors to highlight key areas for investigation
- Case summary in narrative format and case recommendation
- Exploration of the financial crimes global-graph using an interactive and visual Graph Explorer tool.
- Integrates fully with FCDM (data can be loaded directly from Behavior Detection (AML) or ECM instance) and ICIJ data sources. It can be enhanced to support other data sources such as watchlist and company hierarchy data
- It is built on OFS Compliance Studio, which includes a highly scalable in-memory Oracle Graph Analytics Engine (PGX), AI, and machine learning.
- Integrated with Quantifind API for additional information on case entities.

2.2 Import Notebooks

To import notebooks, see the **Importing Notebooks** section in the OFS Investigation Hub Installation Guide.

2.3 Administration and Configuration Activities

An administrator can configure the following Notebooks:

- **Special Investigation**: Enables the investigator to search for one or multiple names and/or addresses to examine the network, red flags, and risk factors.
- **ECM_Integration_L1**: Enable Level 1 Case Investigators to access additional rich information about a case such as a case summary, a detailed narrative about case entities, graph view of a case, and so on, which is otherwise not available in ECM. Allows the investigator to explore a case including graph, risk factors, and red flags.
- **ECM_Case_Narrative**: Enables the investigators to access only case summary and a detailed narrative about case entities. The graph view for the case is not available in this notebook.

> **NOTE** Administrators must share only the Special Investigation notebook with users (investigators) and users will clone the Notebook for their investigation.

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, map users to groups, domains and roles, grant permissions based on user roles and requirements, etc.

> **NOTE** The **DSUSRGRP** group must be assigned to the user using Investigation Hub.

For more information, see the **Mapping User Groups** section in the OFS Compliance Studio Administration and Configuration Guide.

4 Configuring the Investigation Hub Parameters

This chapter provides information on configuring the Investigation Hub parameters for the following seeded notebooks:

- Special Investigation
- ECM Integration L1
- ECM Case Narrative

Topics:

- Configuring Notebook Parameters
- Renaming Input Parameters

4.1 Configuring Notebook Parameters

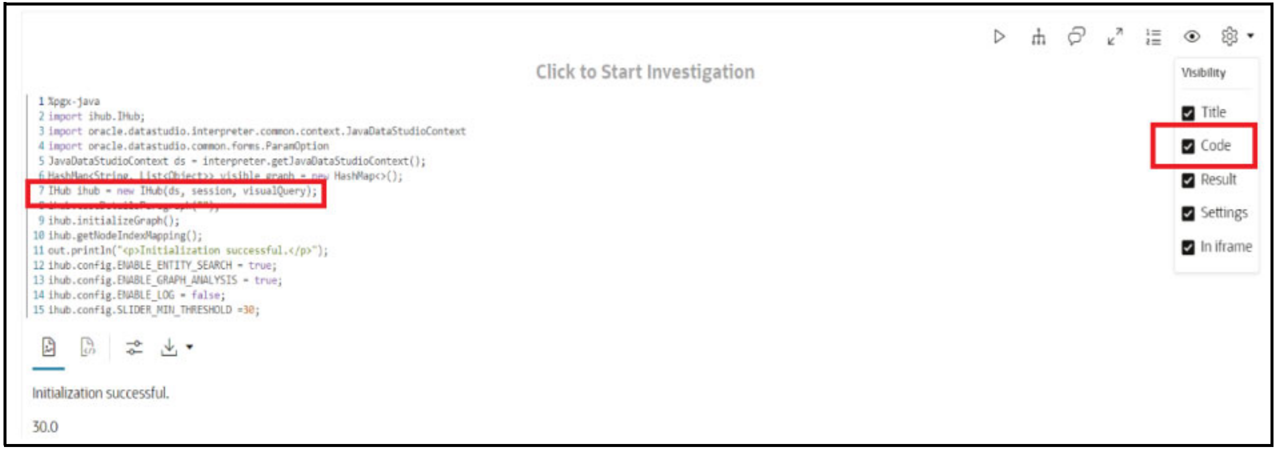
4.1.1 Updating Configuration

An admin user can configure the parameters of the Investigation Hub notebooks by updating the values.

To update the value, follow these steps:

1. Login to **Data Studio**.
2. Navigate to the **Investigation Hub** folder.
3. Open the desired notebook.
4. Open the notebook and navigate to the **Click to Start Investigation** paragraph in ECM Integration L1 and Special Investigation notebooks or **Entity Summary Risk Report** paragraph in the ECM Case Narrative notebook.

Figure 1: Click to Start Investigation Paragraph



```
%pgx-java
import ihub.IHub;
import oracle.datastudio.interpreter.common.context.JavaDataStudioContext
import oracle.datastudio.common.forms.ParamOption
JavaDataStudioContext ds = interpreter.getJavaDataStudioContext();
HashMap<String, List<Object>> visible_graph = new HashMap<>();
IHub ihub = new IHub(ds, session, visualQuery);
[illegible]
ihub.initializeGraph();
ihub.getNodeIndexMapping();
out.println("<p>Initialization successful.</p>");
ihub.config.ENABLE_ENTITY_SEARCH = true;
ihub.config.ENABLE_GRAPH_ANALYSIS = true;
ihub.config.ENABLE_LOG = false;
ihub.config.SLIDER_MIN_THRESHOLD =30;
```

```
Initialization successful.

30.0
```

5. Click on the **Visibility** icon and select the **Code** option.
6. Navigate to the line `IHub ihub = new Ihub(ds, session, visualQuery);`
7. Update the value by adding a line just after the above line with `ihub.config.` followed by **variable_name** and then the **value**.

   For example, `ihub.config.DATE_DISPLAY_FORMAT = "MM-dd-yyyy";`

8. To update another value, add the additional line as above. For example, with multiple parameter updates as follows:

```
IHub ihub = new IHub(ds, session, visualQuery);
ihub.config.DATE_DISPLAY_FORMAT = "MM-dd-yyyy";
ihub.config.HIGH_RISK_MIN_SCORE_BOUNDARY = 5;
```

4.1.2 Adding Parameter to the Notebook

An admin user can configure the parameters of the notebook as described in Table 4.

Table 4: Configure Parameter for Notebook

Parameter	Description	Example Code Snippet to Override
Generic Configuration		
ENABLE_GRAPH_ANALYSIS	It enables graph analysis if the value is set to “true”. The value is either true or false. **NOTE:** Set the value as “false” for the ECM case narrative notebook. Set the value as “True” for the ECM Integration L1 and Special Investigation notebooks.	To disable it in the ECM Integration L1 notebook: `ihub.config.ENABLE_GRAPH_ANALYSIS = false;`
ENABLE_ENTITY_SEARCH	It enables additional entity search if the value is set to “true”. The value is either true or false. By default, it is set to true.	To disable it in ECM Integration L1 notebook: `ihub.config.ENABLE_ENTITY_SEARCH = false;`

RISK_PROHIBITED_LIST_OF_BUSINESS	It indicates the prohibited list of businesses.
	For example, Bank.
		To update the list:
		`ihub.config.RISK_PROHIBITED_LIST_OF_BUSINESS = new ArrayList();`
		`ihub.config.RISK_PROHIBITED_LIST_OF_BUSINESS.add("BANK");`
		`ihub.config.RISK_PROHIBITED_LIST_OF_BUSINESS.add("AGRI");`
		`ihub.config.RISK_PROHIBITED_LIST_OF_BUSINESS.add("GOVT");`
		Or
		`ihub.config.RISK_PROHIBITED_LIST_OF_BUSINESS = List.of("BANK","AGRI","GOVT");`

Table 4: Configure Parameter for Notebook

Parameter	Description	Example Code Snippet to Override
TAX_HAVEN_COUNTRY_LIST	It indicates the list of countries having taxes. For example: CHE, BHS, ANB, US, etc.	To update the list: `ihub.config.TAX_HAVEN_COUNTRY_LIST = List.of("CHE", "BHS", "ANB", "US");`
DATE_DISPLAY_FORMAT	It indicates the date format to display in the narrative/tabular format. The format is YYYY-MMM-DD. **NOTE:** For more information on formatting the string, see the DateTimeFormatter.	Set the date format as follows: `ihub.config.DATE_DISPLAY_FORMAT = MM-dd-yyyy;`
ENABLE_LOG	It enables logs in to the paragraph output if the value is set to “true”. The value is true or false. It helps to debug the notebook. By default, set it to false.	To enable: `ihub.config.ENABLE_LOG = true;`
HIGH_RISK_MIN_SCORE_BOUNDARY	It indicates the risk score. If the values are more than HIGH_RISK_MIN_SCORE_BOUNDARY, it is considered as high risk. The values are 7 - 10.	To set the minimum score boundary to 6: `ihub.config.HIGH_RISK_MIN_SCORE_BOUNDARY = 6;`
Sub Graph Loading		

NODE_PROVIDER_EXPAND_EXCLUSION_LIST	It indicates subgraph loading for investigation; neighbors of the node providers mentioned in this list will be excluded from subsequent loading.
	The values are:
	“Derived Entity”, “Institution”,
	“ICIJ External Address”,
	“ICIJ External Entity”
		To override the exclusion list:
		`ihub.config.NODE_PROVIDER_EXPAND_EXCLUSION_LIST = List.of("Derived Entity",`
		`"ICIJ External Entity", "ICIJ External Address");`
		or
		`ihub.config.NODE_PROVIDER_EXPAND_EXCLUSION_LIST.remove("Institution");`

INITIAL_LOAD_HOPS	It indicates the number of hops to load for the initial step while creating the sub graph. The recommended values are between 0 to 2. The default value is 1. **Note:** This affects the sub graph loading time. For example: The value can be set to 0 to load entities without any edges.	To override the value: `ihub.config.INITIAL_LOAD_HOPS = 0;`

Table 4: Configure Parameter for Notebook

Parameter	Description	Example Code Snippet to Override
Real-time Matching **NOTE:** It is applicable for Entity Search only.		
SEARCH_TYPE	It indicates the search type for matching. The value is fuzzy.	To update it to "exact": `ihub.config.SEARCH_TYPE = "exact";`
NAME_SEARCH_METHOD	It indicates the scoring method for the name search. The values are mlboostednamematching and jaroWinkler.	To update the scoring method to “jaroWinkler”: `ihub.config.NAME_SEARCH_METHOD = “jaroWinkler”;`
ADDRESS_SEARCH_METHOD	It indicates the scoring method for address search. The values are mlboostedaddressmatching and jaroWinkler.	To update the scoring method to “jaroWinkler”: `ihub.config.ADDRESS_SEARCH_METHOD = “jaroWinkler”;`
CONFIGURABLE_CED	CED for address matching, where CED stands for Character Edit Distance. Comparison is good for matching textual values that may be misspelled and thus have one or two character differences between each other. For more information, see the OFS Compliance Studio Matching Guide. The value is auto.	To set the value to 2: `ihub.config.CONFIGURABLE_CED = 2`
SLIDER_MIN_THRESHOLD	It indicates the match score minimum value on the slider. For example, 50.	To set the minimum value to 70%: `ihub.config.SLIDER_MIN_THRESHOLD = 70;`
SLIDER_MAX_THRESHOLD	It indicates match score maximum value on the slider. For example, 100.	To set the maximum value to 90%: `ihub.config.SLIDER_MAX_THRESHOLD = 90;`
SLIDER_THRESHOLD_STEP	It indicates the slider step size. For example, 5.	To set the slider step to 10%: `ihub.config.SLIDER_THRESHOLD_STEP = 10;`
SLIDER_THRESHOLD_DEFAULT	It indicates the match score default value on the slider. For example, 50.	To set the max value to 80%: `ihub.config.SLIDER_THRESHOLD_DEFAULT = 80;`
Status Code Mapping		

Table 4: Configure Parameter for Notebook

Parameter	Description	Example Code Snippet to Override

STATUS_CODE_MAPPING	It indicates the value of the status code. It is used to show Account and Customer status in the narrative paragraph.
	The status codes are as follows:
	● A - Active
	● I - Inactive
	● N - Not a customer
	● P - Pending
		To set the more mapping:
		`ihub.config.STATUS_CODE_MAPPING.put("C", "Priority Customer");`
		To override the mapping:
		`ihub.config.STATUS_CODE_MAPPING = new HashMap<>;`
		`ihub.config.STATUS_CODE_MAPPING.put("A", "Active");`
		`ihub.config.STATUS_CODE_MAPPING.put("D", "Dormant");`
		`ihub.config.STATUS_CODE_MAPPING.put("P", "Priority Customer");`

Colour and Weight for Risk Factor and Red Flags		
DEFAULT_SCORE_FONT_COLOUR	It indicates the default score font colour. For more information, see colour Names for other colour code. For example, seagreen.	To set the default colour to blue: `ihub.config.DEFAULT_SCORE_FONT_COLOUR = blue;`
HIGHLIGHTED_SCORE_FONT_COLOUR	It indicates the highlighted score font colour. For example, crimson.	To set the default colour to red: `ihub.config.DEFAULT_SCORE_FONT_COLOUR = red;`
DEFAULT_SCORE_FONT_WEIGHT	It indicates the default score font weight. The supported values are normal and bold. For example, normal.	To set the default font to bold: `ihub.config.DEFAULT_SCORE_FONT_WEIGHT = bold;`
HIGHLIGHTED_SCORE_FONT_WEIGHT	It indicates the highlighted score font weight. The supported values are normal and bold. For example, bold.	To set the highlighted font to normal: `ihub.config.HIGHLIGHTED_SCORE_FONT_WEIGHT = normal;`
HIGHLIGHTED_SCORE_MIN_VALUE	It indicates the highlighted score minimum value. If the value is less than HIGHLIGHTED_SCORE_MIN_VALUE, it will be displayed with default colour and weight. The minimum value is 6.	To set the min value to highlight as 3: `ihub.config.HIGHLIGHTED_SCORE_MIN_VALUE = 3;`
Disposition Score		

Table 4: Configure Parameter for Notebook

<table>
<tr><th>Parameter</th><th>Description</th><th>Example Code Snippet to Override</th></tr>
<tr>
<td>DISPOSITION_COLOUR_MAP</td>
<td>It indicates a colour map for disposition.
To set the colour depending on score boundaries, add the minimum score as 1 and colour in the hashmap. The higher value for the boundary will be less than the next boundary.
For example, the colour and its boundary values are:
● seagreen [0,25]
● gold (25,51]
● darkorange (51,76]
● crimson (76,100]
If mapping is 0, the minimum value is 0 and maximum value is 25 for seagreen colour.
If mapping is 25, the minimum value is 26 and maximum value is 51 for gold colour.
If mapping is 51, the minimum value is 52 and maximum value is 76 for darkorange colour.
If mapping is 76, the minimum value is 77 and maximum value is 100 for crimson colour.</td>
<td>To override the colour:
<table>
<tr><th>Mapping</th><th>Colour</th><th>Min Value</th><th>Max Value</th></tr>
<tr><td>0</td><td>seagreen</td><td>0</td><td>24</td></tr>
<tr><td>24</td><td>yellow</td><td>25</td><td>39</td></tr>
<tr><td>39</td><td>gold</td><td>40</td><td>59</td></tr>
<tr><td>59</td><td>darkorange</td><td>60</td><td>74</td></tr>
<tr><td>74</td><td>crimson</td><td>75</td><td>89</td></tr>
<tr><td>89</td><td>brown</td><td>90</td><td>100</td></tr>
</table>
ihub.config.DISPOSITION_COLOUR_MAP = new HashMap<>;
ihub.config.DISPOSITION_COLOUR_MAP .put(0, "seagreen");
ihub.config.DISPOSITION_COLOUR_MAP .put(24, "yellow");
ihub.config.DISPOSITION_COLOUR_MAP .put(39, "gold");
ihub.config.DISPOSITION_COLOUR_MAP .put(59, "darkorange");
ihub.config.DISPOSITION_COLOUR_MAP .put(74, "crimson");
ihub.config.DISPOSITION_COLOUR_MAP .put(89, "brown");</td>
</tr>
<tr>
<td>RECOMMENDATION_CLOSE_MESSAGE</td>
<td>It indicates the default recommendation message. This is recommended when the disposition score is less than the first declared range.
For example,
Close Case (Reason: False Positive)</td>
<td>To update the recommendation close message:
ihub.config.RECOMMENDATION_CLOSE_MESSAGE = "False Positive, Close Case";</td>
</tr>
</table>

Table 4: Configure Parameter for Notebook

Parameter	Description	Example Code Snippet to Override

DISPOSITION_RECOMMENDATION_MAP	It indicates a recommendation map for disposition.
	To set the message depending on score boundaries, add the minimum score as 1 and the message in the hashmap. The higher value for the boundary will be less than the next boundary.
	For example, the message and its boundary values are:
	● Close Case (Reason: False Positive) [0,25]
	● Low Risk Network (25,51]
	● Medium Risk Network (51,76]
	● Potential High Risk Network (76,100]
	If mapping is 0, the minimum value is 0 and maximum value is 25 for Close Case (Reason: False Positive) message.
	If mapping is 25, the minimum value is 26 and maximum value is 51 for Low Risk Network message.
	If mapping is 51, the minimum value is 52 and maximum value is 76 for Medium Risk Network message.
	If mapping is 76, the minimum value is 77 and maximum value is 100 for Potential High Risk Network message.
		To override the recommendation message:
		If mapping is 0, the minimum and maximum values of the Close Case (Reason: False Positive) message are 0 and 24.
		If mapping is 24, the minimum and maximum values of the Low Risk Network message are 25 and 39.
		If mapping is 39, the minimum and maximum values of the Medium Risk Network message are 40 and 59.
		If mapping is 59, the minimum and maximum values of the Potential High Risk Network message are 60 and 74.
		If mapping is 74, the minimum and maximum values of the High Risk Network message are 75 and 89.
		If mapping is 89, the minimum and maximum values of the Very High Risk Network message are 90 and 100.
		`ihub.config.DISPOSITION_RECOMMENDATION_MAP = new HashMap<>;`
		`ihub.config.DISPOSITION_RECOMMENDATION_MAP .put(0, "Close Case (Reason: False Positive)");`
		`ihub.config.DISPOSITION_RECOMMENDATION_MAP .put(24, "Low Risk Network");`
		`ihub.config.DISPOSITION_RECOMMENDATION_MAP .put(39, "Medium Risk Network");`
		`ihub.config.DISPOSITION_RECOMMENDATION_MAP .put(59, "Potential High Risk Network");`

Table 4: Configure Parameter for Notebook

Parameter	Description	Example Code Snippet to Override

		`ihub.config.DISPOSITION_RECOMMENDATION_MAP .put(74, "High Risk Network");`
		`ihub.config.DISPOSITION_RECOMMENDATION_MAP .put(89, "Very High Risk Network");`

COUNTRY_SEPARATOR	It indicates the delimiter for the attribute "Country". The value is ~ ;	
Advance Configuration **NOTE:** The following parameters should be updated only when the graph pipeline is customized.		
GRAPH_NAME	It indicates the PG graph name. For example, FINANCIAL_CRIME_GLOBAL_GRAPH.	-
GRAPH_PIPELINE_ID	It indicates the graph pipeline ID. For example, 853e4164-XXXX-XXXX-XXXX-XXXXXXXXXXXX	-
RESULT_CASE_GRAPH	It indicates the subgraph name. For example, caseGraph.	-
SEPARATOR	It indicates the separator used in the graph pipeline. For example, ~	-
KEY_COLUMN_ID	It indicates the key attribute for the node name. For example, id.	-

4.2 Renaming Input Parameters

An admin user can rename the input parameters in the notebooks.

To rename the input parameters, follow these steps:

1. Login to **Data Studio**.
2. Navigate to the **Investigation Hub** folder.
3. Open the desired notebook.
4. Open the notebook and navigate to the **Click to Start Investigation** paragraph in ECM Integration L1 and Special Investigation notebooks or **Entity Summary Risk Report** paragraph in the ECM Case Narrative notebook.

5. Click on the **Visibility** icon and select the **Code** option.
6. Navigate to the line `IHub ihub = new Ihub(ds, session, visualQuery);`
7. Update the value by adding a line just after the above line with `ihub.dynamicForms.` followed by **variable_name** and then the **value**.

   For example, `ihub.dynamicForms.addressTextBox = "Complete Address";`
8. To update another value, add the additional line as above. For example, with multiple parameter updates as follows:

```
IHub ihub = new IHub(ds, session, visualQuery);
ihub.dynamicForms.addressTextBox = "Complete Address";
ihub.config.nameTextBox = "Full Name";
```

4.2.1 Configuring Parameters for Entity Search

The Dynamic Search enables you to identify non-case entities within the Notebook. Users can customize the dynamic forms for the notebook as described in the Table 5.

Table 5: Configure Parameters for Entity Search

Parameter	Significance
taxIdTextBox	It indicates the label of the text box for Tax ID. The value is Tax Id.
nameTextBox	It indicates the label of the text box for Name. The value is Name.
addressTextBox	It indicates the label of the text box for the Address. The value is Address.
dateTextBox	It indicates the label of the Date and Time picker for Date. The value is Date.
defaultDateFormat	It indicates the format for the Date and Time picker. The value is yyyy-MM-dd HH:mm:ss
defaultDateValue	It indicates the default Date and Time Value. The value is 1970-01-01 00:00:00
useDateCheckBox	It indicates the check box's label to suggest if to use the Date value for “Non-Case Entity”. The value is Use Date?
emptyListCheckBox	It indicates the label of the check box for resetting the non-case entities in the L1 notebook or List of Search Entity in the SI notebook. The value is Empty the existing entities list?
matchScoreThresholdSlider	It indicates the label of the slider for threshold score. The value is Minimum Match Score Cutoff in %

Table 5: Configure Parameters for Entity Search

Parameter	Significance
searchIndexCheckBox	It indicates the label of the check boxes to select the target search entities (Opensearch Indexes). The value is Target search entities (selected targets are used for "Name" and "Address" matching).
maxMatchCountTextBox	It indicates the label of the text box for Top Critical Matches. The value is Top Critical Matches.
noOfHopsToPreFetch	It indicates the label of the text box for number of hops to be considered in the case graph. The value is the Number of Hops to Pre-Fetch.
noOfHopsToDisplay	It indicates the label of the text box for the number of hops to be displayed initially. The value is the Number of Hops to Display.
additionalEntitiesTextBox	It indicates the label of the text box for additional Customer or Account internal ids for historical summary report. The value is Additional Entity Ids (supports multiple comma-separated Customer or Account entities).
minTransactionAmountTextBox	It indicates the label of the textbox for "Minimum transaction Amount". The value is the Minimum transaction amount.
maxTransactionAmountTextBox	It indicates the label of the textbox for "Maximum transaction Amount". The value is the Maximum transaction amount.

5 Customizing Investigation Hub Notebook

An admin user can refer to the Investigation Hub notebook with source code to understand and customize the output of each paragraph. Once satisfied with the customization, an admin user can compile the code which is present in the notebook to create a jar and then configure it in Compliance Studio to publish the changes for the Investigation Hub user.

> **NOTE** For more information reach out to My Oracle Support (MOS).

Investigation Hub notebook with source code is present inside the directory, Investigation Hub/Source Code in the Data Studio.

Customizing and publishing changes for Investigation Hub users involves the following process:

- Customize Notebook
- Preparing Java Archive (jar)
- Updating Investigation Hub Jar in the Compliance Studio
- Update Investigation Hub Notebook (without source code)
- Add/Update Case-Notebook Mapping

5.1 Customize Notebook

The Investigation Hub notebook (with source code) has complete java code in the "Entity Summary Risk Report" paragraph of the ECM Case Narrative notebook or "Click to Start Investigation" paragraph in the ECM Integration L1 or Special Investigation notebook. The code has multiple java classes for different entities. An admin user can go through the code to understand the implementation of each paragraph. The changes may be simple or complex based on the nature of customization. In this section, we will discuss a few common customizations.

An admin user who wants to customize the notebook should have experience with the following:

- Java
- SQL
- PGQL (required for graph-based analysis)

If customizations are complex, we recommend to setup IDE and then do the customization. For more information, see the Setup an Integrated Development Environment section.

5.1.1 Customize Entity Summary Risk Report (Narrative)

Entity Summary Risk Report is also referred to as Narrative and is generated in two step process. We collect all the information first and then generate the narrative. Let's evaluate the following scenarios:

- Customer Transaction Summary should show transaction type wise distribution
- Additional Attributes in Account Summary

5.1.1.1 Customer Transaction Summary should show transaction type wise distribution

Let's say in the pre-configured narrative, you want to see the transaction type wise count for each account-wise Customer's transaction summary.

To update the pre-configured Investigation Hub notebook, follow the steps:

- Enable the flag to show transaction breakup in the `SummaryGenerator#getAcctWiseDetail()` method as given below.

  // Replace below line

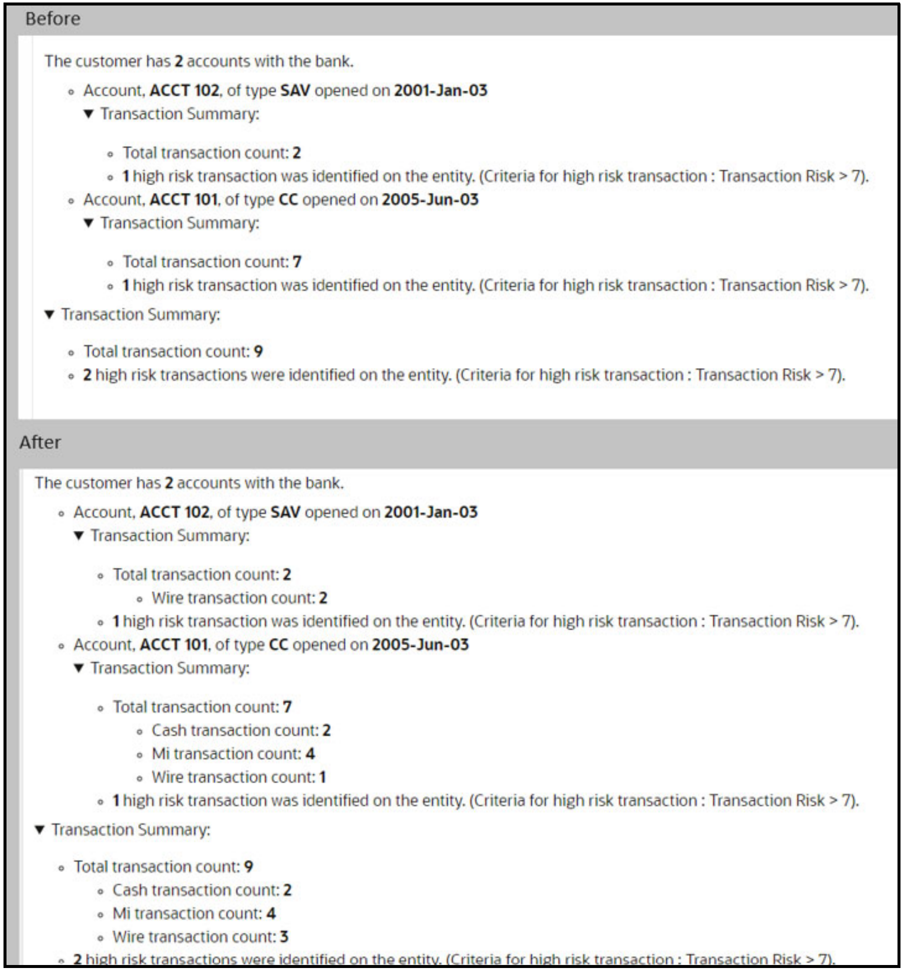
  ```
   getTransactionDetails(detail.transactionDetail, reportStringBuilder,
  false);
  ```

  // with below lines

  ```
   getTransactionDetails(detail.transactionDetail, reportStringBuilder,
  true);
  ```

  After the changes, the Customer's transaction summary will show transaction type wise distribution as shown below.

Figure 2: Customer's transaction summary



Before

The customer has **2** accounts with the bank.

- Account, **ACCT 102**, of type **SAV** opened on **2001-Jan-03**
  - ▼ Transaction Summary:
    - Total transaction count: **2**
    - **1** high risk transaction was identified on the entity. (Criteria for high risk transaction : Transaction Risk > 7).
- Account, **ACCT 101**, of type **CC** opened on **2005-Jun-03**
  - ▼ Transaction Summary:
    - Total transaction count: **7**
    - **1** high risk transaction was identified on the entity. (Criteria for high risk transaction : Transaction Risk > 7).

▼ Transaction Summary:

- Total transaction count: **9**
- **2** high risk transactions were identified on the entity. (Criteria for high risk transaction : Transaction Risk > 7).

After

The customer has **2** accounts with the bank.

- Account, **ACCT 102**, of type **SAV** opened on **2001-Jan-03**
  - ▼ Transaction Summary:
    - Total transaction count: **2**
      - Wire transaction count: **2**
    - **1** high risk transaction was identified on the entity. (Criteria for high risk transaction : Transaction Risk > 7).
- Account, **ACCT 101**, of type **CC** opened on **2005-Jun-03**
  - ▼ Transaction Summary:
    - Total transaction count: **7**
      - Cash transaction count: **2**
      - Mi transaction count: **4**
      - Wire transaction count: **1**
    - **1** high risk transaction was identified on the entity. (Criteria for high risk transaction : Transaction Risk > 7).

▼ Transaction Summary:

- Total transaction count: **9**
  - Cash transaction count: **2**
  - Mi transaction count: **4**
  - Wire transaction count: **3**
- **2** high risk transactions were identified on the entity. (Criteria for high risk transaction : Transaction Risk > 7).

5.1.1.2 Additional Attributes in Account Summary

Let’s say you want to see **account opening method** in the account summary.

To update the pre-configured Investigation Hub notebook, follow these steps :

- Check if the attribute is defined in the graph definition: Since the attribute "account opening method" is not defined in the Account node provider, update the graph pipeline to add the property. For more information, see **Graphs** section in the OFS Compliance Studio User Guide. Let's say it was added as an attribute, `Opening Method`.
- Query Update: Update the query for information collection.
  - **PGQL Query**: Modify the query in the `GraphPgqlQueries#getAccountDetails()` method to add the attribute `Opening Method` as shown below.

```
public String getAccountDetails(
    Set<String> nodeIds, PgxGraph resultGraph, GetInfoFromGraph
getInfo) {
  if (getInfo.verifyIfNodeProviderExist(List.of("Account"),
resultGraph, false, false)) {
    return "SELECT "
        + "n.Name,"
        + "n.Status,"
        + "n.\"Tax Id\","
        + "n.Address,"
        + "n.\"Entity Type\","
        + "n.City,"
        + "n.Country,"
        + "n.State,"
        + "n.Jurisdiction,"
        + "n.\"Business Domain\","
        + "n.Risk,"
        + "n.D_date,"
        + "n.\"Original Id\","
        + "n.\"Opening Method\""
        + "id(n) "
        + "MATCH (n:Account) where id(n) in ('"
        + String.join("','", nodeIds)
        + "')";
  } else {
    return null;
  }
}
```

- **SQL query**: Modify the query in the `SqlQueries#getAccountDetails()` method to add the attribute `Opening Method` as shown below.

```
public String getAccountDetails(Set<String> nodeIds, HashSet<String> tableHashSet) {
  StringBuilder queryString = new StringBuilder();
  if (tableHashSet.contains("VW_FCC_ACCOUNT853E4164_0968_4CB6_A6F3_2B4930614A8B")) {
    queryString
        .append("SELECT /*+ parallel(")
        .append(config.PARALLEL_HINT)
        .append(
            ") */ n.\"Name\", n.\"Status\", n.\"Tax Id\", n.\"Address\", n.\"Entity Type\", n.\"City\", n")
        .append(
            ".\"Country\", n.\"State\", n.\"Jurisdiction\", n.\"Business Domain\", n.\"Risk\", n.\"D_date\", n")
        .append(
            ".\"Original Id\", n.\"Id\" , n.\"Label\", n.\"Opening Method\" FROM vw_fcc_account853e4164_0968_4cb6_a6f3_2b4930614a8b n ")
        .append("WHERE n.\"Id\" IN ( '")
        .append(String.join("','", nodeIds))
        .append("')");
  }
  return queryString.toString();
}
```

- **Update the entity, Account, to store the value**: Add a variable for `openingMethod` and respective `getter` and `setter` in class `AccountDetail` as shown below.

```
String openingMethod;


public String getOpeningMethod() {
  return openingMethod ;
}


public void setOpeningMethod(String openingMethod ) {
  this.openingMethod = openingMethod ;
}
```

- **Setting the value**: Modify these methods, `GetInfoFromDb#gatherAccountDetails()` and `GetInfoFromGraph#gatherAccountDetails()` to set the value as shown below respectively,
  - — Set the value in `GetInfoFromDb#gatherAccountDetails()`

    ```
    accountDetail.setOriginalId(result.getString(13));
    accountDetail.setOpeningMethod(result.getString(16));
    ```

  - — Set the value in `GetInfoFromGraph#gatherAccountDetails()`

    ```
    accountDetail.setOriginalId(result.getString(13));
    accountDetail.setOpeningMethod(result.getString(14));
    ```

- **Risk Report update**: Modify the risk report in the `SummaryGenerator#getAccountReport()` method by appending the message and the value as shown below.

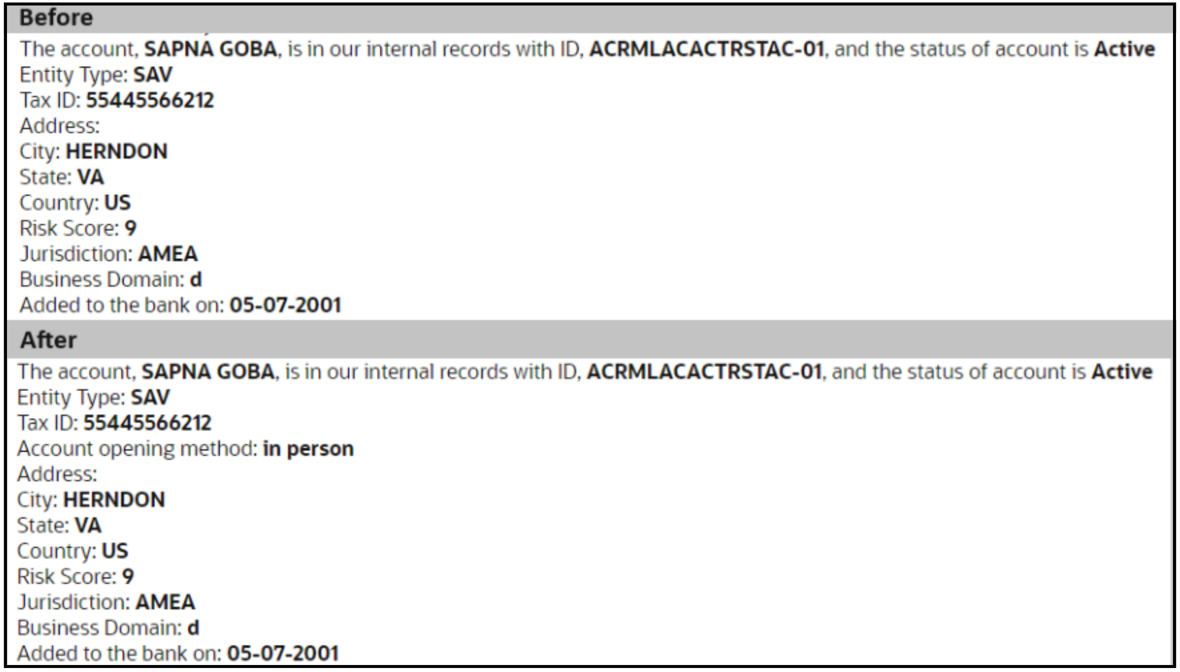
```
public void getAccountReport(AccountDetail accountDetail, StringBuilder
reportStringBuilder) {
  if (accountDetail != null) {
    reportStringBuilder
        .append("<details>")
        .append("<summary>Account Summary of <b>")
        .append(accountDetail.getName())
        .append("</b></summary>")
        .append("<p>")
        .append("The account, <b>")
        .append(accountDetail.getName())
        .append("</b>, is in our internal records with ID, <b>")
        .append(accountDetail.getOriginalId())
        .append("</b>, and the status of account ");
    String acctStatus = accountDetail.getStatus();
    reportStringBuilder
        .append(acctStatus.startsWith("code") ? "has <b>" : "is <b>")
        .append(accountDetail.getStatus())
        .append("</b>")
        .append("</br>")
        .append("Entity Type: <b>")
        .append(accountDetail.getEntityType())
        .append("</b>")
        .append("</br>")
        .append("Tax ID: <b>")
        .append(accountDetail.getTaxId())
```

```
        .append("</b>")
        .append("</br>")
        .append("Account opening method: <b>")
        .append(accountDetail.getOpeningMethod())
        .append("</br>")
        .append("Address: ")
        .append(getList(accountDetail.getAddresses()))
        .append("</br>")
        .append("City: <b>")
        .append(getList(accountDetail.getCities()).append("</b>"))
        .append("</br>")
        .append("State: <b>")
        .append(getList(accountDetail.getStates()).append("</b>"))
        .append("</br>")
        .append("Country: <b>")
        .append(getList(accountDetail.getCountries()).append("</b>"))
        .append("</br>")
        .append("Risk Score: <b>")
        .append(accountDetail.getRiskScore())
        .append("</b>")
        .append("</br>")
        .append("Jurisdiction: <b>")
        .append(accountDetail.getJurisdiction())
        .append("</b>")
        .append("</br>")
        .append("Business Domain: <b>")
        .append(accountDetail.getBusinessDomain())
        .append("</b>")
        .append("</br>")
        .append("Added to the bank on: <b>")
        .append(accountDetail.getAddedDate())
        .append("</b>");
    getTransactionDetails(accountDetail.transactionDetail,
reportStringBuilder, true);
    getRelatedCustSummary(accountDetail, reportStringBuilder);
    getComplianceSummary(accountDetail.getEventDetails(),
reportStringBuilder, false);
```

```
    getRiskFactorsAndRedFlags(accountDetail.getCustomerDetails(),
reportStringBuilder, true);

    reportStringBuilder.append("</p>").append("<hr>").append("</
details>");

  } else {

    ihubUtil.log("Skipping Account report as passed account detail is
null.");

  }

}
```

After all the changes are done, value of the **account opening method** will be shown in the account summary.

Figure 3: Account Opening Method



Before

The account, **SAPNA GOBA**, is in our internal records with ID, **ACRMLACACTRSTAC-01**, and the status of account is **Active**
Entity Type: **SAV**
Tax ID: **55445566212**
Address:
City: **HERNDON**
State: **VA**
Country: **US**
Risk Score: **9**
Jurisdiction: **AMEA**
Business Domain: **d**
Added to the bank on: **05-07-2001**

After

The account, **SAPNA GOBA**, is in our internal records with ID, **ACRMLACACTRSTAC-01**, and the status of account is **Active**
Entity Type: **SAV**
Tax ID: **55445566212**
Account opening method: **in person**
Address:
City: **HERNDON**
State: **VA**
Country: **US**
Risk Score: **9**
Jurisdiction: **AMEA**
Business Domain: **d**
Added to the bank on: **05-07-2001**

5.1.2 Additional Risk Factor and High Risk Entities

Let's say you want to add an additional risk factor, which shows the count of high-risk entities, where entities are either Customers or Accounts.

To update, follow these steps:

- **Add a query to get the count**: Add a method in `GraphPgqlQueries#getHighRiskEntitiesCount()` as given below.

```
public String getHighRiskEntityCount(
    boolean forVisibleGraph,
    long minRiskBoundary,
    PgxGraph resultGraph,
    GetInfoFromGraph getInfo) {
```

```
  if (getInfo.verifyIfNodeProviderExist(
      List.of("Account", "Customer"), resultGraph, false, false)) {
    StringBuilder queryBuilder = new StringBuilder();
    queryBuilder
        .append("SELECT ")
        .append("count(n.\"Original Id\")")
        .append(" MATCH (n) ")
        .append(
            "WHERE n.Label in ('Account', 'Customer') ")
        .append("and n.\"Risk\" > ")
        .append(minRiskBoundary);
    if (forVisibleGraph) {
      queryBuilder.append(" and id(n) in ? ");
    }
    return queryBuilder.toString();
  } else {
    return null;
  }
}
```

- To add a row in Risk Factors, either modify the `IHub#getRiskFactor` method or add a new method and then call that method inside the `getRiskFactor` method as given below.

  // Add below lines at the end of the method to add new rows

  ```
  log("Fetching high risk entity count.");
      long highRiskEntityCaseGraph =
          getCountBasedOnQuery(
              false,
              graphPgqlQueries.getHighRiskEntityCount(
                  true,
                  config.HIGH_RISK_MIN_SCORE_BOUNDARY,
                  resultGraph,
                  (GetInfoFromGraph) getInfo),
              null);
      long highRiskEntityVisibleGraph =
          getCountBasedOnQuery(
              true,
  ```

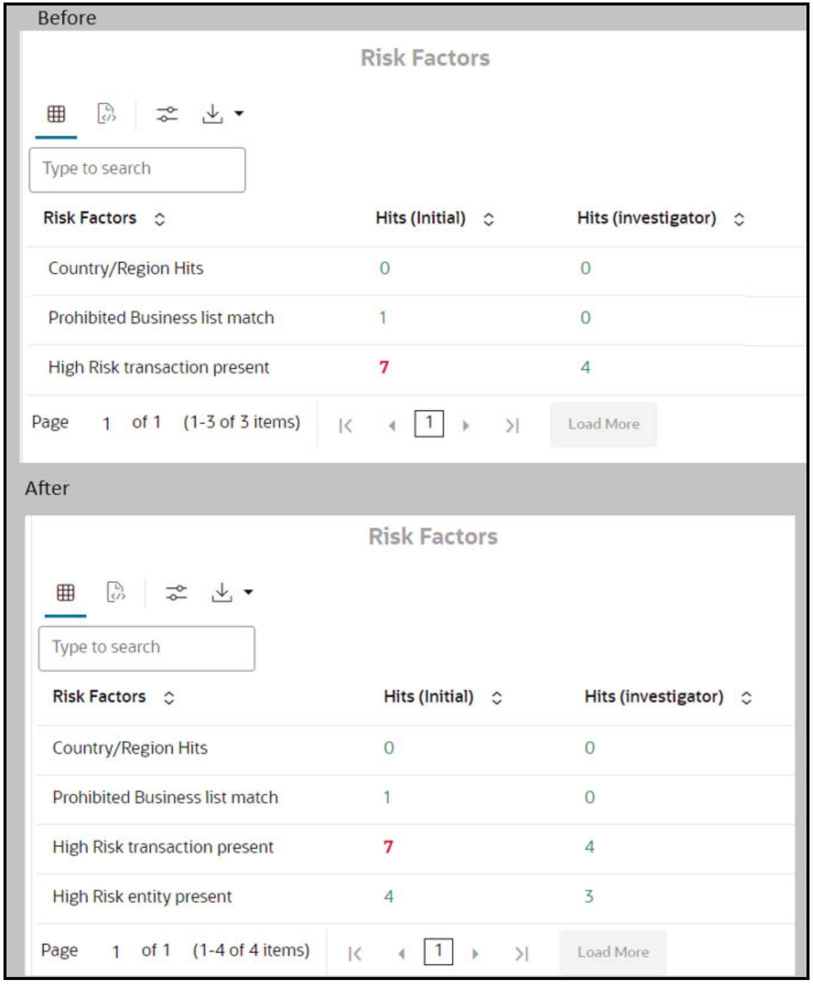
```
        graphPgqlQueries.getHighRiskEntityCount(
            true,
            config.HIGH_RISK_MIN_SCORE_BOUNDARY,
            resultGraph,
            (GetInfoFromGraph) getInfo),
        List.of(visibleNodeList));
report.addRow(
    "High Risk entity present",
    formatScore(highRiskEntityCaseGraph),
    formatScore(highRiskEntityVisibleGraph));
```

//Before the below line

```
printStatement(report.printTable(true));
```

After all the changes are done, the **Risk Factors** section will show additional row as shown below.

Figure 4: Risk Factors



Before

Risk Factors

Risk Factors	Hits (Initial)	Hits (investigator)
Country/Region Hits	0	0
Prohibited Business list match	1	0
High Risk transaction present	7	4

After

Risk Factors

Risk Factors	Hits (Initial)	Hits (investigator)
Country/Region Hits	0	0
Prohibited Business list match	1	0
High Risk transaction present	7	4
High Risk entity present	4	3

5.1.3 Additional Red flag, Customer with more than certain No. of Accounts

Let's say you want to add an additional red flag, which shows the number of customers with more than a certain number of accounts.

To update, follow these steps:

- **Configuration for the Number of Accounts**: Instead of fixing the value, let's add it as a dynamic form where Investigation Hub users can update if required. To update it as given below.

  // Code snippet to add a text box, with default value 3 and then additional validation to validate the user input is a valid integer.

  ```
   String textBoxMessage = "Minimum number of associated account to
  consider it as a red flag";

     String minAccountCountString = ds.textbox(textBoxMessage, "3",
  textBoxMessage).trim();

     int minAccountCount = 0;

     minAccountCount =

         validateTextBoxAndGetIntValue(minAccountCountString, 3,
  textBoxMessage);
  ```

- **Add a query to get the customer ids with more than certain account**: Add a method in `GraphPgqlQueries#getCustomerCountWithMoreThanCertainAccount()` as given below.

  ```
  public String getCustomerCountWithMoreThanCertainAccount(

      PgxGraph resultGraph,

      GetInfoFromGraph getInfo,

      Integer minAccountCount,

      boolean forVisibleGraph) {

    if (getInfo.verifyIfNodeProviderExist(

        List.of("Customer", "Account"), resultGraph, true, false)

       && getInfo.verifyIfEdgeProviderExists("Cust Has Acct", resultGraph,
  false)) {

      return "SELECT n.\"Original Id\", count(n.\"Original Id\") as
  count_id "

          + " FROM MATCH (n:Customer) - [e] -> (acct: Account) "

          + (forVisibleGraph ? " where id(n) in ?" : "")

          + " group by n.\"Original Id\" "

          + " having count_id > " + minAccountCount;

    } else {

      return null;

    }
  ```

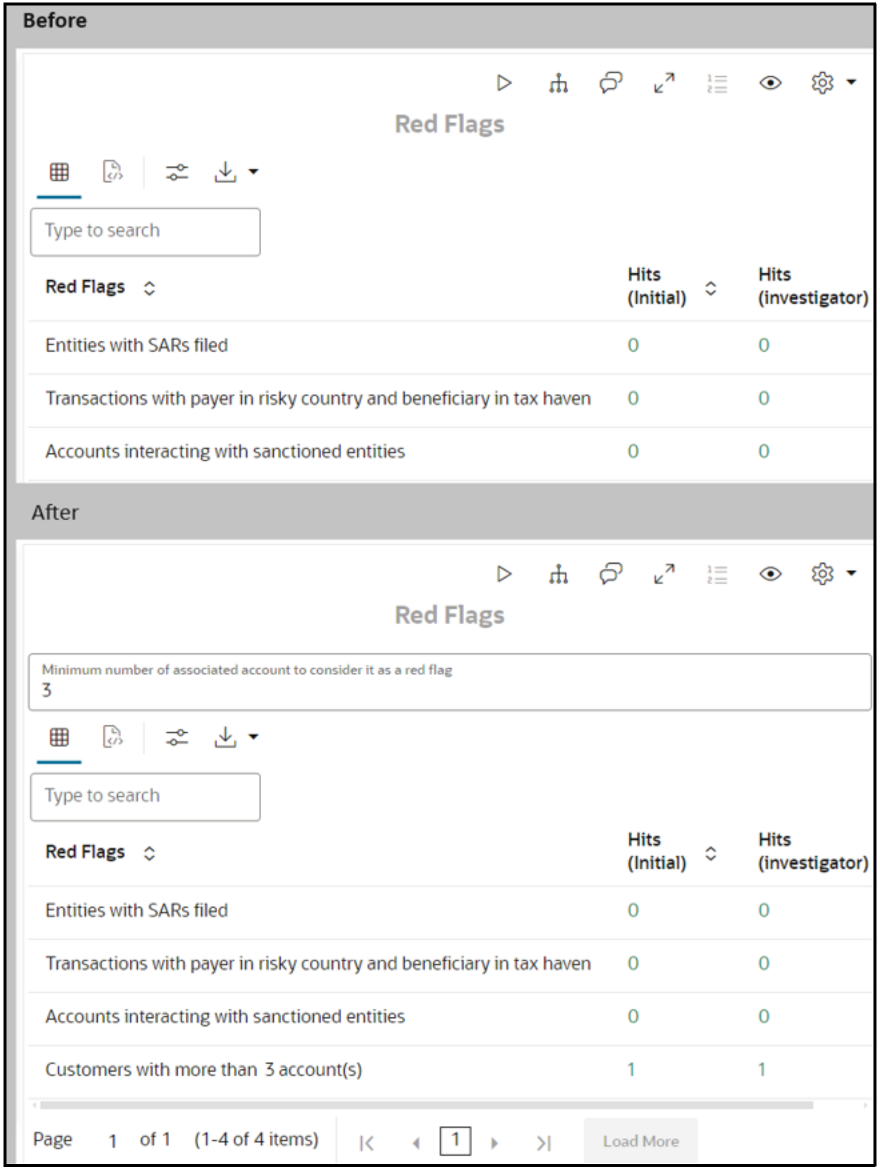
```
}
```

- To add a row in the Red flag, either modify the `IHub#getRedFlag()` method or add a new method and then call that method inside the `getRedFlag()` method .

```
public void addRedFlag(Table report, List<String> visibleNodeList)
throws DynamicFormsException {

   String textBoxMessage = "Minimum number of associated account to
consider it as a red flag";

   String minAccountCountString = ds.textbox(textBoxMessage, "3",
textBoxMessage).trim();

   int minAccountCount = 0;

   minAccountCount =

       validateTextBoxAndGetIntValue(minAccountCountString, 3,
textBoxMessage);

   log("Fetching entities with more than " + minAccountCount + "
accounts");

   long countVisibleGraph = queryVisibleGraph(resultGraph,
graphPgqlQueries.getCustomerCountWithMoreThanCertainAccount(resultGraph,

       (GetInfoFromGraph) getInfo,minAccountCount, true),
List.of(visibleNodeList)).getRows().size();

   long countCaseGraph = queryCaseGraph(resultGraph,
graphPgqlQueries.getCustomerCountWithMoreThanCertainAccount(resultGraph,

       (GetInfoFromGraph) getInfo,minAccountCount,
false)).getRows().size();

   report.addRow(

       "Customers with more than " + minAccountCount + " account(s)",
formatScore(countCaseGraph), formatScore(countVisibleGraph));

 }


 public void getRedFlag() {

 ...

   addRedFlag(report, visibleNodeList);

  // calling the method, addRedFlag(), before printing the final statement.

   printStatement(report.printTable(true));

 }
```

After all the changes are done, the **Red Flag** section will show additional row as shown below.

Figure 5: Red Flags



Before

Red Flags

Type to search

Red Flags	Hits (Initial)	Hits (investigator)
Entities with SARs filed	0	0
Transactions with payer in risky country and beneficiary in tax haven	0	0
Accounts interacting with sanctioned entities	0	0

After

Red Flags

Minimum number of associated account to consider it as a red flag
3

Type to search

Red Flags	Hits (Initial)	Hits (investigator)
Entities with SARs filed	0	0
Transactions with payer in risky country and beneficiary in tax haven	0	0
Accounts interacting with sanctioned entities	0	0
Customers with more than 3 account(s)	1	1

Page 1 of 1 (1-4 of 4 items) 1

5.1.4 Network Disposition Score and its Breakdown

Disposition score should consider only Customers and Accounts and the Breakdown must show an additional column to show contribution.

Let's say you want to change the score of disposition score to consider only Customers and Account and the breakdown must show an additional column to contribute toward the final risk score.

To update, follow these steps:

- Modify the `GraphPgqlQueries#getDispositionScore()` and `GraphPgqlQueries#getDispositionScoreBreakdown()` queries for disposition score and its breakdown, respectively, as shown below:

  // Modified method

  ```
  public String getDispositionScore(boolean forVisibleGraph) {
  ```

```
    return "SELECT sum(n.Risk * 10)/count(n) as "
        + "network_disposition_score "
        + "FROM MATCH (n) "
        + "where n.Label in ('Customer', 'Account') and n.Risk is not
null"
        + (forVisibleGraph ? " and id(n) in ?" : "");
  }
```

// A new method to get the node count of Customer and Account present in the graph/visible graph

```
  public String getNodeCount(boolean forVisibleGraph) {
    return "SELECT count(n) as "
        + " node_count "
        + " from match(n) "
        + " where n.Label in ('Customer', 'Account') and n.Risk is not
null "
        + (forVisibleGraph ? " and id(n) in ?" : "");
  }
```

// Modified method

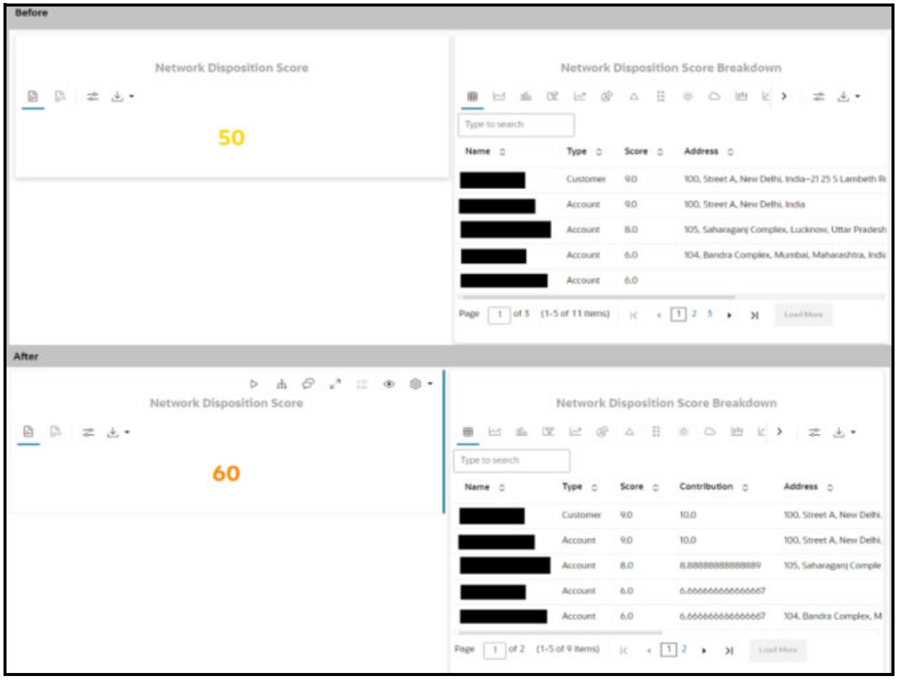
```
  public String getDispositionScoreBreakdown(long nodeCount, boolean
forVisibleGraph) {
    return "SELECT "
        + "n.Name as Name, "
        + "n.Label as Type, "
        + "n.Risk as Score, "
        + "(n.Risk * 10)/" + nodeCount + " as Contribution, "
        + "n.Address as Address, "
        + "n.D_date as \"Opened Date\", "
       + "CASE WHEN n.Label = 'Account' THEN '2' ELSE '0' END as Acct,  "
       + "CASE WHEN n.Label = 'Customer' THEN '1' ELSE '0' END as Cust  "
        + "FROM MATCH (n) "
        + "where n.Label in ('Customer', 'Account') and n.Risk is not
null"
        + (forVisibleGraph ? " and id(n) in ? " : "")
        + " order by Score desc";
  }
```

- Update the `IHub#getNetworkDispositionScoreBreakdown()` method to get the count and pass it to the updated `GraphPgqlQueries#getDispositionScoreBreakdown()` method as shown below.

```
public void getNetworkDispositionScoreBreakdown() {

   if (validateIfGraphAnalysisIsEnabled() && resultGraph != null) {

     long nodeCount = getCountBasedOnQuery(true,
graphPgqlQueries.getNodeCount(true), List.of(getVisibleGraphNode()));

     String query =
graphPgqlQueries.getDispositionScoreBreakdown(nodeCount,true);

     queryVisibleGraphAndPrintTable(resultGraph, query,
List.of(getVisibleGraphNode()));

   }

 }
```

After all the changes are done, the network disposition score and the breakdown are updated as shown below.

Figure 6: Network Disposition Score and its Breakdown



Before

Network Disposition Score

50

Network Disposition Score Breakdown

Type to search

Name	Type	Score	Address
[redacted]	Customer	9.0	100, Street A, New Delhi, India~21 25 S Lambeth R
[redacted]	Account	9.0	100, Street A, New Delhi, India
[redacted]	Account	8.0	105, Saharaganj Complex, Lucknow, Uttar Pradesh
[redacted]	Account	6.0	104, Bandra Complex, Mumbai, Maharashtra, Indi
[redacted]	Account	6.0	

Page 1 of 3 (1-5 of 11 items) 1 2 3 Load More

After

Network Disposition Score

60

Network Disposition Score Breakdown

Type to search

Name	Type	Score	Contribution	Address
[redacted]	Customer	9.0	10.0	100, Street A, New Delhi,
[redacted]	Account	9.0	10.0	100, Street A, New Delhi,
[redacted]	Account	8.0	8.888888888888889	105, Saharaganj Comple
[redacted]	Account	6.0	6.666666666666667	
[redacted]	Account	6.0	6.666666666666667	104, Bandra Complex, M

Page 1 of 2 (1-5 of 9 items) 1 2 Load More

5.2 Preparing Java Archive (jar)

The code needs to be complied as a java archive (jar) to publish the changes after customization.

5.2.1 Setup an Integrated Development Environment

NOTE An admin user can use the java principle to extend the default classes and override or add additional methods as required. This will reduce the effort for re-applying the customization on future upgrades.

To setup an Integrated Development Environment (IDE) to compile, follow these steps:

1. Download and Install JDK 11 and your choice of IDE with Java Support.
2. Download the jars from `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/interpreter-server/pgx-interpreter-*/lib` and `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/interpreter-server/pgx-interpreter-*/extralibs` directories into your local directory.
3. Add these jars as default jars (default class path of JDK) in the IDE.
4. Copy the code from the notebook and create the respective java class in IDE.

5.2.2 Create the Java Archive

As per your IDE, run the respective command to create a jar.

5.3 Updating Investigation Hub Jar in the Compliance Studio

To update the complied jar, follow these steps:

1. Copy the compiled jar and paste it in the following directories:

   ```
   <COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/interpreter-server/pgx-interpreter-*/extralibs
   ```

   ```
   <COMPLIANCE_STUDIO_INSTALLATION_PATH>/mmg-home/mmg-studio/interpreter-server/pgx-interpreter-*/extralibs
   ```

   NOTE Take a backup of the existing investigation jar into the folder outside of `<COMPLIANCE_STUDIO_INSTALLATION_PATH>` for safekeeping.

2. Restart Compliance Studio.

5.4 Update Investigation Hub Notebook (without source code)

To update Investigation Hub notebook (without source code), follow these steps:

1. Clone the without source code notebook and create a new notebook.
2. Update the code in the notebook (without source code) in respective paragraphs, if new methods were added to support additional paragraphs or customization.
3. Verify the changes with this notebook.

5.5 Add/Update Case-Notebook Mapping

If customization was done in the separate notebook, then configure this notebook against required case type and role in **FCC_CM_CTYPE_NB_MAPPING** table in the ECM Atomic Schema. For more information, see the Mapping User Roles and Case Type with Investigation Hub section.

Once the mapping is updated, the Investigation Hub users with role for that case type will see the updated notebook for case investigation.

6 Additional Configuration

This chapter provides information about additional configurations for OFS Investigation Hub.

Topics:

- Configuring Interpreters
- Managing Templates

6.1 Configuring Interpreters

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

The supported interpreters are PGX, PGQL, Python, Markdown, etc.

For more information, see the **Configure Interpreters** section in the OFS Compliance Studio Administration and Configuration Guide.

6.2 Managing Templates

Templates (**FCGM Default Template)** allow you to create a common way of viewing data in Investigation Hub and cover both graphs and other visualizations.

For more information, see the **Configuring Templates** section in the OFS Compliance Studio User Guide.

7 Integrating Investigation Hub with ECM

Investigation Hub is integrated with ECM to enable Case Investigators to access additional rich information about a case such as a case summary, a detailed narrative about case entities, graph view of a case, and so on, which is otherwise not available in ECM.

Topics:

- Prerequisites
- Enable Investigation Hub Tab in ECM Case Designer
- Investigation Hub User Role Precedence
- Mapping User Roles and Case Type with Investigation Hub
- Authenticating the User to Access Investigation Tab in ECM

7.1 Prerequisites

For more information on the ECM patch, see the **Prerequisites** section in OFS Investigation Hub Installation Guide.

7.2 Enable Investigation Hub Tab in ECM Case Designer

> **ATTENTION** In case ECM 8.0.7.* version is used; the Investigation Hub tab configuration in the ECM Case Designer has to be done manually. To configure manually, see **Adding Optional Entities to the Case Type** section in the OFS ECM Administration And Configuration Guide.

For ECM 8.0.8.* and above versions, the pre-configured ECM patch enables the Investigation Hub tab for **AMLSURV** case types. An admin user can add the tab for other case types by using the Case Designer component in the ECM.

For more information, see **Adding Optional Entities to the Case Type** section in the OFS ECM Administration And Configuration Guide.

> **NOTE**
> - Pre-configured Investigation Hub Notebook may not be applicable to all case types.
> - Add case type and notebook Id mappings in the FCC_CM_CTYPE_NB_MAPPING table.

7.3 Investigation Hub User Role Precedence

Investigation Hub uses role precedence in the FCC_CM_NB_ROLES table to decide which notebook to investigate when users have multiple roles where the mapped notebook ids are different.

To set the precedence among roles by Admin user, follow these steps:

1. Connect to ECM's Atomic Schema.
2. Edit records present in the **FCC_CM_NB_ROLES** table.

3. Enter the user role in the **V_USERROLE** column and the precedence in the **N_PRECEDENCE** column.

> **NOTE** Lower value of precedence has higher precedence.

7.4 Mapping User Roles and Case Type with Investigation Hub

This section can be used to configure specific roles and case types. An admin user can map the Investigation Hub notebook against a role and case type.

For more information on column details, see the **Configuring FCC_CM_CTYPE_NB_MAPPING Table** section in the OFS Investigation Hub Installation Guide.

Map additional case types, roles, and respective notebook id in the table. You can see examples as listed in Table 6.

Table 6: Example

V_CASETYPE	V_USERROLE	V_NOTEBOOK_ID	V_CREATED_DATE	V_CREATED_BY	V_UPDATED_BY	V_UPDATED_DATE	V_NB_TOOLBAR	V_ADD_PARA	V_PARA_ACTIONS	V_PARA_CODE
CASE_TYPE_1	ROLE_1	notebook_id_1	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_1	ROLE_2	notebook_id_2	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_2	ROLE_1	notebook_id_1	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_2	ROLE_2	notebook_id_3	02-02-2024	02-02-2024			N	N	Y	N
CASE_TYPE_2	ROLE_3	notebook_id_5	02-02-2024	02-02-2024			N	N	Y	N

> **NOTE** Roll out an update by replacing the existing notebook ids with updated notebook ids. The update may be due to an Investigation Hub version update or roll-out customization.

7.5 Authenticating the User to Access Investigation Tab in ECM

> **NOTE** The user needs a self-signed certificate to authenticate the user for accessing Investigation Tab in ECM.

If the user is not using the self-signed certificate, follow these steps:

1. Copy the following files from `<COMPLIANCE_STUDIO_INSTALLATION_PATH>/deployed/mmg-home/mmg-studio/conf` to the server where ECM is installed.
   - `studio_server.p12`
   - `studio_server.jks`

   > **NOTE** Make sure that the "studio_server.p12" and "studio_server.jks" certificates are compatible with Java 8. This is applicable only if the Compliance Studio server is in JDK 11 and the ECM application server is in Java 8. If there is a difference in Java versions, then both the files "studio_server.p12" and "studio_server.jks" need to be recreated in Compliance Studio server and replaced in all necessary locations. For more information about these certificates, see **Generate Self-signed Certificate** section in the OFS Compliance Studio Installation Guide.

2. Run the following command to create certificate files:

   ```
   openssl pkcs12 -in studio_server.p12 -nokeys -out server_cert.pem
   ```

   ```
   openssl pkcs12 -in studio_server.p12 -nodes -nocerts -out server_key.pem
   ```

   ```
   keytool -certreq -keystore studio_server.jks -alias studio_server -
   keyalg RSA -file client.csr
   ```

   ```
   openssl x509 -req -CA server_cert.pem -CAkey server_key.pem -in
   client.csr -out client_certificate.pem -days 365 –CAcreateserial
   ```

3. Modify the path and run the following command:

   ```
   keytool -import -file "/<ECM Installation Path>/client_certificate.pem"
   -alias studio_server -keystore "<JDK Installed Directory>/lib/security/
   cacerts" -storepass "changeit"
   ```

   For example,

   ```
   keytool -import -file "Testserver/client_certificate.pem" -alias
   studio_server -keystore "jdk-11.0.10/lib/security/cacerts" -storepass
   "changeit"
   ```

8 Appendix

Topics:

- Frequently Asked Questions

8.1 Frequently Asked Questions

You can refer to the Frequently Asked Questions, which are developed with interest to help you resolve some of the Investigation Hub Installation and configuration issues.

1. What happens when a case is opened by two different users who have different roles and mappings?
   - If two users have different roles mapped to two different notebook IDs, a case is opened with a different notebook for the two users.
   - If two users have different roles mapped to same notebook IDs, a case is opened with a same notebook for the two users.
   - User with multiple roles where each role is mapped with a different notebook, then the case is opened for the role with the highest precedence.
   - Two users open a case with different roles and different mapped notebooks at different point of time, then both users will see different notebooks for the same case.
2. How do I update mapping while an upgrade or customization is made in the case?

   In case of an upgrade, update notebook ID mappings in the FCC_CM_CTYPE_NB_MAPPING table.

   > **NOTE** When the user opens a case which is already opened by another user, then it clones the new notebook ID mapped to the role and opens the cloned notebook for the case.

OFSAA Support

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to OFSAA applications.

Send Us Your Comments

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

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

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

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