

Oracle Financial Services Data Integration Hub

User Guide

Release 8.1.1.0.0

June 2021

F31715-01

ORACLE
Financial Services

Oracle Financial Services Data Integration Hub User Guide

Copyright © 2023 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 the 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

Table 1: Document Control

| Version Number | Revision Date | Change Log |
|----------------|---------------|--|
| 01 | March 2021 | Created the document with instructions for the usage of the OFS Data Integration Release 8.1.1.0.0. |
| 02 | June 2021 | <ul style="list-style-type: none">• Updated the document with instructions on how to add a connector task in PMF Pipeline.• Updated the Derived Column component section. |

Table of Contents

| | | |
|----------|--|-----------|
| 1 | Getting Started..... | 8 |
| 1.1 | Audience | 8 |
| 1.1.1 | <i>Prerequisites for the Audience.....</i> | 8 |
| 1.2 | Conventions | 8 |
| 1.3 | Acronyms | 8 |
| 1.4 | Glossary of Icons | 9 |
| 1.5 | Related Information Sources..... | 10 |
| 2 | Introduction to Data Integration Hub | 12 |
| 2.1 | Overview of Data Integration Hub (DIH)..... | 12 |
| 2.2 | Benefits of DIH..... | 13 |
| 2.3 | Summary of Chapters | 13 |
| 3 | Data Flow in DIH..... | 15 |
| 3.1 | DIH Dataflow Diagram | 15 |
| 4 | Accessing the DIH Application..... | 16 |
| 4.1 | Logging in to DIH | 16 |
| 5 | Configuring DIH | 18 |
| 5.1 | Setting up ODI Connectivity | 18 |
| 5.1.1 | <i>Mandatory Settings.....</i> | 18 |
| 5.1.2 | <i>Optional Settings</i> | 20 |
| 5.2 | Refreshing Application Data Interface | 22 |
| 5.2.1 | <i>Abstraction of Model Changes for Data Movement / ETL Processing</i> | 25 |
| 5.2.2 | <i>Handling Model Changes with Impact on Data Movement / ETL Processing.....</i> | 26 |
| 5.2.3 | <i>Data Model Validation Messages</i> | 27 |
| 5.3 | Configuring System Parameters | 27 |
| 5.3.1 | <i>Understanding the Parameters Window</i> | 27 |
| 5.3.2 | <i>Fields in Parameters Window.....</i> | 28 |
| 5.3.3 | <i>Defining a Parameter.....</i> | 29 |
| 5.3.4 | <i>Modifying and Viewing a Parameter.....</i> | 29 |

| | | |
|----------|--|-----------|
| 5.3.5 | <i>Deleting a Parameter</i> | 30 |
| 5.3.6 | <i>Unpublishing a Parameter</i> | 30 |
| 5.3.7 | <i>Dependency</i> | 30 |
| 5.3.8 | <i>Search and Filter</i> | 31 |
| 5.3.9 | <i>Parameters in EDD and Connectors</i> | 31 |
| 5.4 | Configuring External Data Sources | 31 |
| 5.4.1 | <i>External Data Store Fields</i> | 32 |
| 5.4.2 | <i>Defining an External Data Store</i> | 37 |
| 5.4.3 | <i>Modifying and Viewing an External Data Store</i> | 37 |
| 5.4.4 | <i>Deleting an External Data Store</i> | 38 |
| 5.4.5 | <i>Unpublishing an External Data Store</i> | 38 |
| 5.4.6 | <i>Dependency</i> | 38 |
| 5.4.7 | <i>Search and Filter</i> | 38 |
| 6 | External Data Descriptors and Connectors | 39 |
| 6.1 | External Data Descriptor | 39 |
| 6.1.1 | <i>External Data Descriptor Fields</i> | 42 |
| 6.1.2 | <i>Defining an External Data Descriptor</i> | 47 |
| 6.1.3 | <i>Modifying and Viewing an External Data Descriptor</i> | 50 |
| 6.1.4 | <i>Deleting an External Data Descriptor</i> | 50 |
| 6.1.5 | <i>Dependency</i> | 51 |
| 6.1.6 | <i>Search and Filter</i> | 51 |
| 6.1.7 | <i>Parameters in EDD Definition</i> | 51 |
| 6.2 | Application Data Interface | 52 |
| 6.3 | Connectors | 56 |
| 6.3.1 | <i>Icons and Description</i> | 56 |
| 6.3.2 | <i>Creating Connectors</i> | 58 |
| 6.3.3 | <i>Managing Existing Connectors</i> | 67 |
| 6.3.4 | <i>Understanding the Components</i> | 71 |
| 6.3.5 | <i>Specifying Alias in Connector</i> | 89 |
| 6.3.6 | <i>Connector Properties</i> | 90 |
| 6.3.7 | <i>Modifying and Viewing a Connector</i> | 93 |
| 6.3.8 | <i>Copying a Connector</i> | 94 |

| | | |
|-----------|---|------------|
| 6.3.9 | <i>Deleting a Connector</i> | 94 |
| 6.3.10 | <i>Search and Filter</i> | 94 |
| 6.3.11 | <i>Parameters in Connector</i> | 95 |
| 7 | Processing DIH Definitions in Preparation for Executing Data Movement | 96 |
| 7.1 | Refreshing Target Datastore | 96 |
| 7.2 | Publishing Connectors and Underlying Metadata | 97 |
| 7.2.1 | <i>Publish and Unpublish</i> | 98 |
| 8 | Executing Data Movement Based on Definitions | 101 |
| 8.1 | External Batch Management Tool | 101 |
| 8.2 | OFSAA Batch | 101 |
| 8.3 | Process Modelling Framework..... | 101 |
| 8.4 | Specifying Runtime Parameters | 103 |
| 9 | Analyzing Operational Details, Utilization of Available Artifacts and User Activity | 105 |
| 9.1 | Execution History | 105 |
| 9.2 | Utilizing Available Artifacts | 109 |
| 9.3 | DIH Activity | 113 |
| 10 | Command Line Utilities | 114 |
| 10.1 | Prerequisites..... | 114 |
| 10.2 | Invoking the Command Line Utilities..... | 114 |
| 11 | Metadata Object Migration | 116 |
| 11.1 | Overview..... | 116 |
| 11.2 | Performing Object Migration | 116 |
| 11.3 | Exporting Objects from Source Environment | 117 |
| 11.4 | Importing Objects From Target Environment | 118 |
| 12 | Metadata Browser | 119 |
| 12.1 | Overview..... | 119 |
| 13 | Data Domain Browser | 120 |
| 13.1.1 | <i>Prerequisites for Data Domain Browser</i> | 120 |

| | | |
|-----------|--|------------|
| 13.1.2 | <i>Audience</i> | 120 |
| 13.2 | About Oracle Insurance Data Foundation..... | 121 |
| 13.3 | Using Data Domain Browser | 122 |
| 13.3.1 | <i>Launching DDB</i> | 122 |
| 13.3.2 | <i>Data Domain Browser Navigation</i> | 122 |
| 13.3.3 | <i>Information Rendered by DDB</i> | 123 |
| 13.3.4 | <i>Navigating Data Domain Browser</i> | 125 |
| 13.3.5 | <i>Use Cases</i> | 139 |
| 14 | Appendix A: Frequently Asked Questions (FAQs) | 144 |
| 14.1 | General FAQs | 144 |
| 14.2 | Upgrade FAQs..... | 148 |

1 Getting Started

1.1 Audience

The DIH User Guide is intended for the following audience:

- **ETL Developers:** The ETL Developers from the IT Department of the financial services institution, who do the data sourcing.
- **Business Analysts:** The business analysts from the IT Department of the financial services institution, who do the mapping of the tables.

1.1.1 Prerequisites for the Audience

- One or more OFSAA applications bearing Staging entities, which may be installed before or after installing DIH.
- Oracle Data Integrator (ODI) must be installed.

1.2 Conventions

The following text conventions are used in this document.

Table 2: Document Conventions

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

1.3 Acronyms

The following table defines the acronyms used in this guide.

Table 3: Acronyms

| Acronym | Description |
|---------|----------------------|
| DIH | Data Integration Hub |
| UI | User Interface |








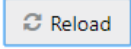





| Acronym | Description |
|---------|----------------------------|
| ODI | Oracle Data Integrator |
| ADI | Application Data Interface |
| KM | Knowledge Module |
| EDD | External Data Descriptor |
| Apps | Application |
| EDS | External Data Source |
| OLH | Oracle Loader for Hadoop |





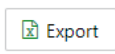
1.4

Glossary of Icons

This manual refers to all or some of the following icons.

Table 4: Icons

| Icons | Description |
|--|----------------------------------|
|  or  | To create a function |
|  | To edit a function |
|  | To delete a function |
|  | To view Dependencies |
|  | To copy a function |
|  or  | To refresh or reload a function |
|  | To start a function |
|  | To return to the previous window |
|  | To download a file |
|  | To add a Join |
|  | To remove a Join |

| Icons | Description |
|---|---|
|  | To auto-map source and target |
|  | To filter the items |
|  | To search for an item in the Source and Target list |
|  | To import mapping excel |
|  | To export mappings to excel |

1.5 Related Information Sources

We strive to keep this and all other related documents updated regularly; visit the [OHC Documentation Library](#) to download the latest version available there. The list of related documents is provided here.

- [OHC Documentation Library](#) for **OFS Data Integration (OFS DI) Application Pack**:
 - *Financial Services Data Integration Hub (DIH)*
 - *Oracle Financial Services Data Foundation Integration With Fusion Accounting Hub Cloud*
 - *Oracle Insurance Data Foundation Integration With Fusion Accounting Hub Cloud*
 - *Interface for Oracle FLEXCUBE Universal Banking System (FCUBS)*
 - *Interface for Oracle Banking Platform (OBP)*
 - *Interface for Data Relationship Management (DRM)*
 - *Financial Services Data Integration Installation Guide*
- [OHC Documentation Library](#) for **OFS AAI Application Pack**:
 - *OFS Advanced Analytical Applications Infrastructure (OFS AAI) Application Pack Installation and Configuration Guide*
 - *OFS Analytical Applications Infrastructure User Guide*
 - *OFS Analytical Applications Infrastructure Administration Guide*
 - *Oracle Financial Services Analytical Applications Infrastructure Environment Check Utility Guide*
- **Additional documents:**
 - [OFSAA Licensing Information User Manual Release 8.1.0.0.0](#)
 - [OFS Analytical Applications Infrastructure Security Guide](#)
 - [OFSAAI FAQ Document](#)

- [*OFS Analytical Applications 8.1.0.0.0 Technology Matrix*](#)
- [*Oracle Financial Services Analytical Applications Infrastructure Cloning Guide*](#)

2 Introduction to Data Integration Hub

This chapter provides an overview of the Data Integration Hub (DIH), its benefits, and the structure of the remaining chapters in this User Guide.

Topics:

- [Overview of Data Integration Hub](#)
- [Benefits of DIH](#)
- [Summary of Chapters](#)

2.1 Overview of Data Integration Hub (DIH)

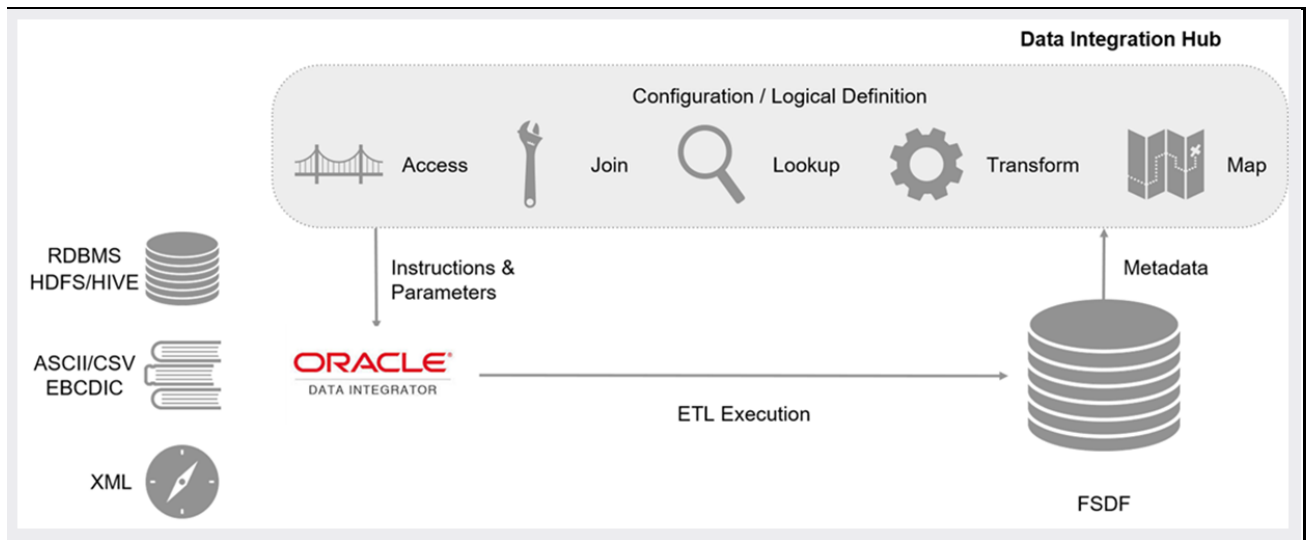
The Data Integration Hub enables data exchange between OFSAA and external systems. This is facilitated through logical abstraction of the OFSAA Data Foundation (Financial Services Data Foundation and Insurance Data Foundation) exposed as Application Data Interfaces (ADI). External Data Sources (EDS) and External Data Descriptors (EDD) are defined through the DIH user interface, which also facilitates the mapping of EDDs to ADIs, forming Connectors.

DIH publishes information on ADI, EDS, EDD, and Connectors, alongside other relevant metadata, to Oracle Data Integrator (ODI), which delivers high-performance data movement and transformation among enterprise platforms with its open and integrated E-LT architecture and extended support for Big Data.

DIH does not supplant ETL tools or processes. Instead, it provides a logical abstraction of physical attributes in FSDF and OIDF, allowing user interaction in business terms while defining data exchange tasks, without adding any additional overhead or hops upon execution of data movement. DIH also shields users from modeling complexities of Data Foundation, especially when it comes to handling data around slowly changing dimensions (SCD).

Factory integration between select Oracle applications and OFSAA is also rendered over DIH. Such applications currently include banking platforms (FLEXCUBE and OBP), change management solutions (DRM), and accounting platforms (FAH and FAHC). Note that, such factory integration is rendered through additional modules, the details of which are available from respective user guides.

Figure 1: DIH Overview



2.2 Benefits of DIH

DIH provides the following benefits:

- Provides a level of abstraction and insulates upstream ETL processes from staging model changes.
- Supports Big Data.
- Provides a simplified mapping window for loading data into OFSAA staging/result through an abstract layer.
- Removes ETL technicalities by prepackaging and predefining ODI parameters based on different use cases.
- Supports Download Specification which is a system managed meta object for upstream data mapping.
- Process/execution optimization for OFSAA data and known use cases. For example, in a direct one-to-one mapping, there will not be any hopping.
- Hosts pre-defined integration with Oracle Products that play the role of data providers to Analytical Applications.
- Helps in getting lineage directly from source data to staging till results.

2.3 Summary of Chapters

This user guide provides information on the mapping of an External Data Descriptor (EDD) with Application Data Interface (ADI). It begins with information on how a Parameter is defined, how a Source System is defined in DIH, how an EDD and ADI are set, and how they are connected. The metadata is then published to ODI. Execution depends on batches that are scheduled in the Financial Services Institutions and does not happen in DIH. The Execution process is discussed in the last chapter.

The guide has the following chapters:

- [Data Flow in DIH](#): This chapter explains the flow of data in DIH.
- [Logging in to DIH](#): This chapter explains how to log in to DIH.
- [Configuring DIH](#): This chapter explains how to manage:
 - Information on connectivity with ODI instance.
 - Data model validation and post-update refresh of ADI.
 - Constant-value, run-time, and current date variables for use with DIH.
 - External data stores that you interact with for data movement
- [Map](#): This chapter explains how to manage External Data Descriptors and Connectors.
- [Publish](#): This chapter explains how to manage the following:
 - Refresh of Target Datastore information, post model upload/update, and ADI refresh.
 - Publishing and un-publishing Connectors.
- [Executing DIH Metadata](#): This chapter explains how the scheduled batches are executed after publishing.
- [Analyze](#): This chapter explains how to analyze the following:
 - Information on the execution of Connectors.
 - Utilization of data model, ADI, and Connectors.
 - Activities are undertaken through DIH.
- [Object Migration](#): This chapter explains how to perform Object Migration.
- [Metadata Browser](#): This chapter details the metadata browser.

3 Data Flow in DIH

This chapter provides high-level information on the data flow used in DIH.

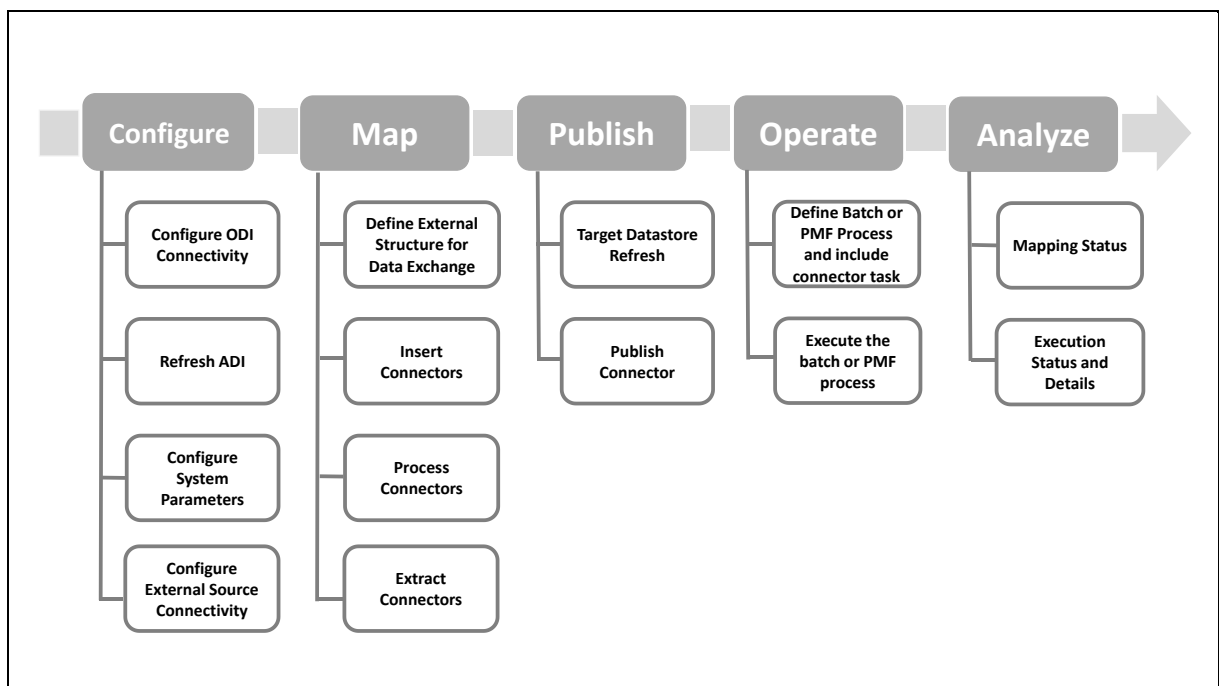
Topics:

- [DIH Dataflow Diagram](#)

3.1 DIH Dataflow Diagram

The following is the DIH data flow diagram.

Figure 2: DIH Dataflow Diagram



4 Accessing the DIH Application

This chapter helps you to access the DIH application.

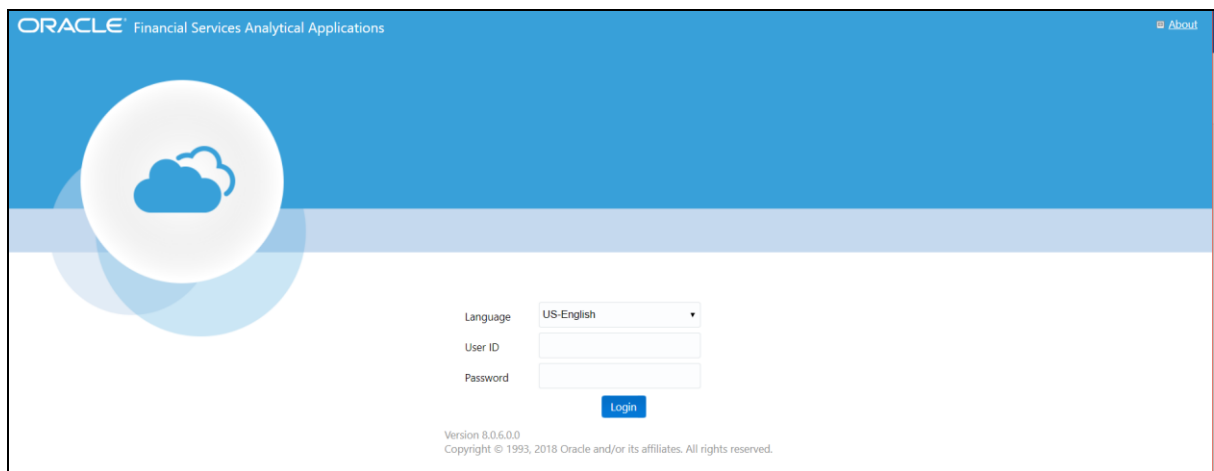
Topics:

- [Logging in to DIH](#)

4.1 Logging in to DIH

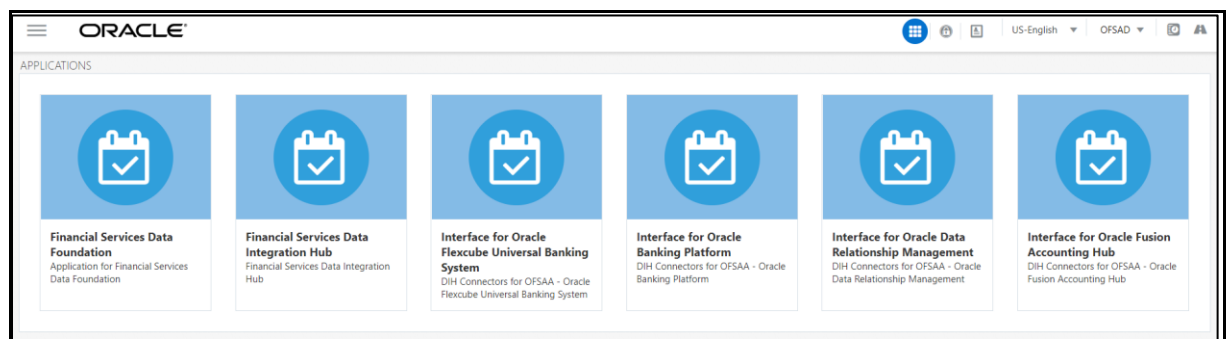
Access the DIH application using the login credentials (**User ID** and **Password**). The built-in security system ensures that you are permitted to access the window and actions based on the authorization only.

Figure 3: DIH - OFSAA Interface Login Window



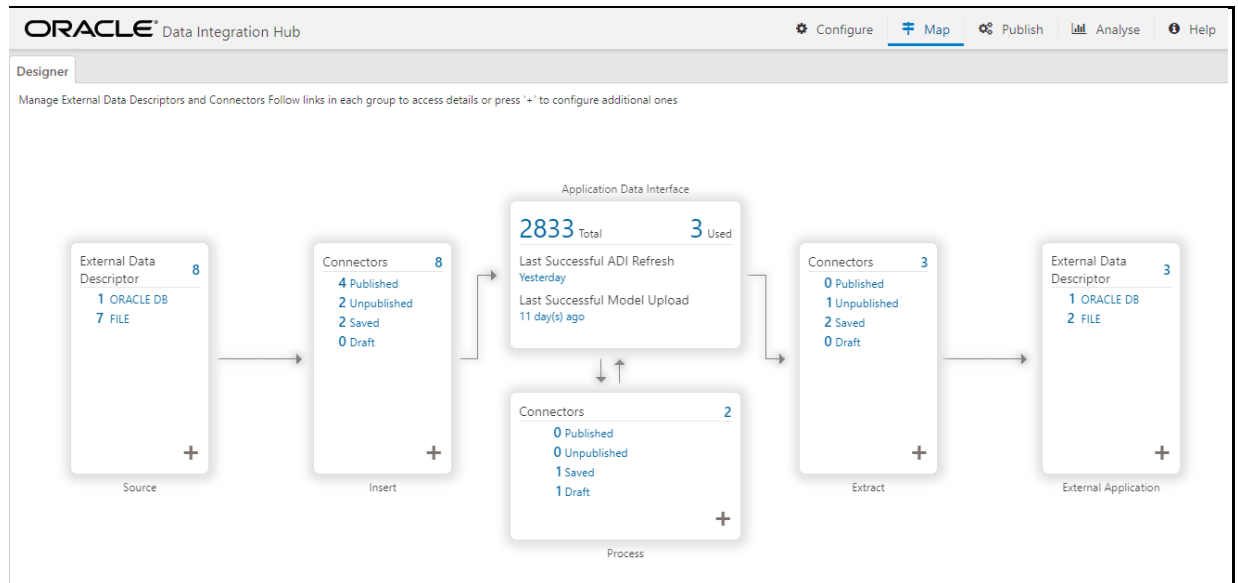
1. After logging into the application, select **Financial Services Data Integration Hub**.

Figure 4: Applications Window



2. Click the Navigation icon . The Navigation List is displayed.
3. Click **Data Integration Hub**. The **DIH Designer** window depicts the design of the application.

Figure 5: DIH Designer Window



The **DIH Designer** window displays the summary of the setup and the activity details. It lists the Parameters, EDS, EDD, and Connector details along with ADIs used. It also displays the details of published and not published connectors along with executed and not executed connectors.

Application Data Interface (ADI) data is available pre-seeded based on the application that is installed.

Application Data Interface enables us to view the logical definition of OFSAA physical entities of the staging and Result area. Select the application and its subtype to view the data.

5 Configuring DIH

This chapter helps you to configure DIH.

Topics:

- [Setting up ODI Connectivity](#)
- [Refreshing Application Data Interface](#)
- [Configuring System Parameters](#)
- [Configuring External Data Sources](#)

5.1 Setting up ODI Connectivity

One of the first tasks during DIH configuration for use is to register details of how it connects with Oracle Data Integrator (ODI). The application allows mandatory and optional settings to be captured for the purpose.

NOTE It is assumed that ODI is installed, configured, and verified as per its documentation, before steps in this section are carried out.

From the **Data Integration Hub Designer** window, select **Configure** and then select **Settings**. This window captures the ODI setup information.

Figure 6: Settings Mandatory Tab

The screenshot shows the Oracle Data Integration Hub interface. At the top, there are navigation tabs: Settings, Refresh ADI, Parameters, and External Data Store. The 'Settings' tab is selected. Below the tabs, there is a header area with 'Manage information on connectivity with your ODI instance Start by capturing mandatory info, add optional details, test connectivity and save' and buttons for 'Test Connection' and 'Save'. The main content area is divided into 'Mandatory' and 'Optional' sections. The 'Mandatory' section contains the following fields:

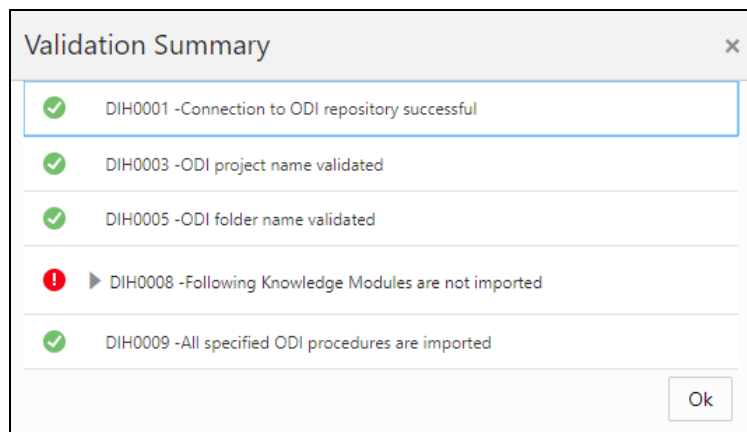
- * ODI User: SUPERVISOR
- ODI Password: [Empty field]
- Use JNDI?: [Toggle switch]
- * Master Repository User: DIH_ODI_REPO
- Master Repository Password: [Empty field]
- * Master Repository Driver: oracle.jdbc.OracleDriver
- * Master Repository Connection: jdbc:oracle:thin:@//10.40.142.248:1521
- * Work Repository: WORKREP
- * Project Name: DIH
- * Folder Name: DIH
- ODI Agent URL: http://whf00atu.in.oracle.com:20 ...

5.1.1 Mandatory Settings

To capture or edit the mandatory settings, follow these steps:

1. Edit the ODI User details.
2. Enter the field details as explained in [Fields in the Mandatory Settings](#) section.
3. To validate details specified, click **Test Connection**.
4. The **Validation Summary** dialog box displays the status of the following details:
 - ODI Connection
 - ODI Project Name
 - ODI Folder Name under the project
 - Import Status of Knowledge Module from DIH
 - Import Status of ODI procedures from DIH

Figure 7: Validation Summary



5. Enter the details and click **Save**.

5.1.1.1 Fields in Mandatory Settings

The following table describes the fields in the **Mandatory Settings**:

Table 5: Fields in Mandatory Settings

| Serial No. | Fields | Description |
|------------|-----------------|--|
| 1 | ODI User | <p>The ODI supervisor user name you defined when creating the master repository or an ODI user name you defined in the Security Navigator after having created in the master repository.</p> <p>The following ODI profiles are required for DIH:</p> <ul style="list-style-type: none"> • CONNECT: To connect to ODI. • DESIGNER: To perform development operations. • TOPOLOGY_ADMIN: To create dataservers for the EDS configured in the DIH. |

| Serial No. | Fields | Description |
|------------|--|--|
| 2 | ODI Password | The ODI supervisor password you defined when creating the master repository or an ODI user password you defined in the Security Navigator after having created the master repository. |
| 3 | Use JNDI | If Yes, enter the Master Repository JNDI. If No, enter Master Repository Database User, Master Repository Database Password, Master Database Driver, and Master Database Connection. |
| 4 | Master Repository Database User | Database user ID/login of the schema (database, library) that contains the ODI master repository. |
| 5 | Master Repository Database Password | Master Repository Database user password. |
| 6 | Master Database Driver | Specifies the driver required to connect to the RDBMS supporting the master repository created from the dropdown list. The default value is oracle.jdbc.OracleDriver. It is not changed when on the Oracle database. |
| 7 | Master Database Connection | The URL is used to establish the JDBC connection to the database hosting the repository. The format is jdbc:oracle:thin:@<Hostname/IP Address>:<Port Number>:<Service Name> |
| 8 | Master Repository JNDI | JNDI name for ODI Master Repository |
| 9 | Work Repository | The name of the work repository that is created previously (Example: WorkRep1). |
| 10 | Project | Enter the Project Name created in ODI. |
| 11 | Folder | Enter the folder name under the project created in ODI. All the packages are created under this location. |
| 12 | Agent URL | Specify the agent URL where the ODI agent is running. This is used to execute a DIH connector from OFSAAI batch/RRF. This is not needed to do data mapping. The format is http://<Hostname/IP Address where ODI agent is running>:<PORT Number>/<Agent Context Name> |

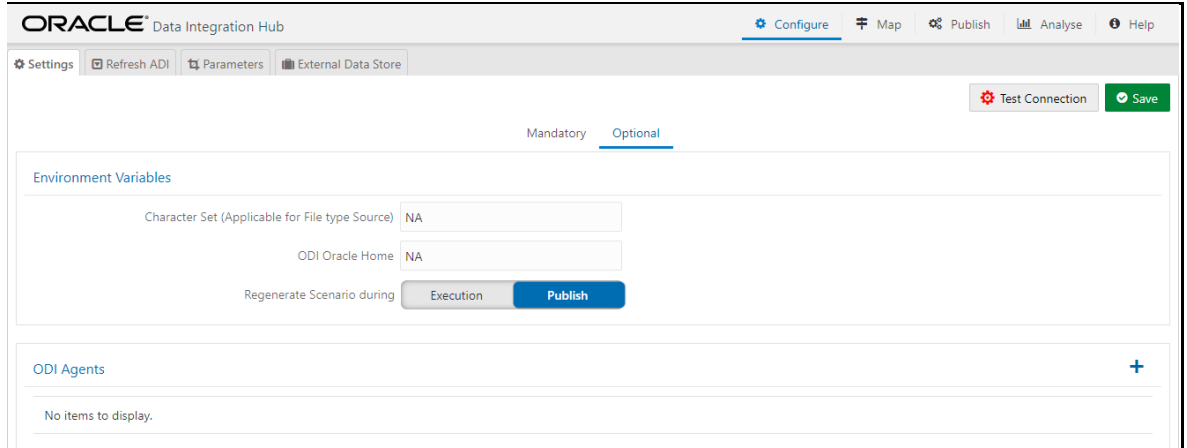
5.1.2 Optional Settings


NOTE The following properties are optional and need not be specified if they are already available as environment variables in the server where the ODI agent is running.

To capture or edit the optional settings, follow these steps:



1. Click the **Optional** tab to edit the optional ODI details.

Figure 8: Settings Optional Tab



2. Enter the field details explained in [Fields in Optional Settings](#).
3. Click **Add**  to add multiple rows for each agent. The **Save Agent As** dialog box is displayed.
4. Enter the **Agent Name** and **URL** and click **OK**.

NOTE It is mandatory to enter the agent's name.

5. Click **Edit**  to modify the saved agent.
6. Click **Delete**  to delete the agent.
7. Enter the details and click Save.

5.1.2.1 Fields in Optional Settings

The following table describes the fields in the **Optional Settings**.

Table 6: Fields in Optional Settings

| Serial No. | Fields | Description |
|------------|---|--|
| 1 | Character Set (Applicable for File type Source) | This field is applicable if the source system type is File. You must specify the character set when you are using the SQL loader for data loading. |
| 2 | ODI Oracle Home | This field is applicable if the source system type is File. You must specify the Oracle Home path where the ODI agent is located. |

| Serial No. | Fields | Description |
|------------|-----------------------|--|
| 3 | Scenario Regeneration | <p>This field allows a choice when Scenario Regeneration is performed in ODI. If you choose 'Publish', Scenario Regeneration is performed during Connector Publish in DIH. If you choose 'Execution', Scenario Regeneration is performed during each Connector execution.</p> <p>NOTE: Scenario Regeneration is needed such that ODI tasks are refreshed or maintained up to date with source, destination, mapping, and related information, and parameters set through the ODI user interface, including those for performance optimization/tuning purposes.</p> <p>Performing Scenario Regeneration during Connector execution is, therefore, a useful option. However, in certain circumstances, especially when several instances of the same Connector are executed concurrently, Scenario Regeneration during Connector execution can be suboptimal and even trigger execution errors.</p> |

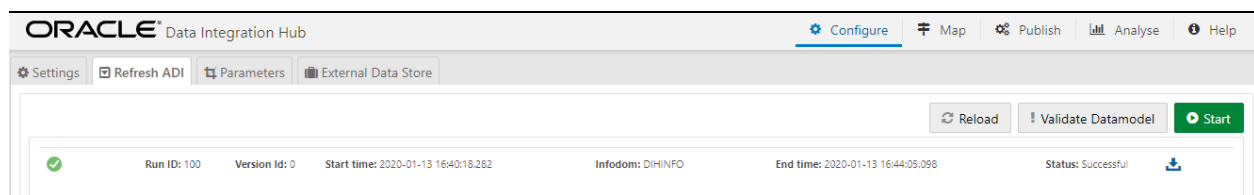
5.2 Refreshing Application Data Interface

The next logical step in setting up DIH for use is to refresh the Application Data Interface information. Application Data Interface (ADI) is a logical abstraction of the OFSAA Data Foundation (FSDF or ODF) data model that you interact with as you use DIH for data movements. The “Refresh ADI” step reads the data model uploaded into your instance of OFSAA and creates ADIs. This step is relevant both when a data model is first uploaded into a fresh OFSAA instance and when the data model is subsequently uploaded with incremental changes. THE “ADI Refresh” process compares the existing ADIs with those based on the updated model and reflects changes appropriately with additional ADIs, additional attributes for existing ADIs, or new data types for existing attributes, as applicable.

To refresh the ADI list, follow these steps:

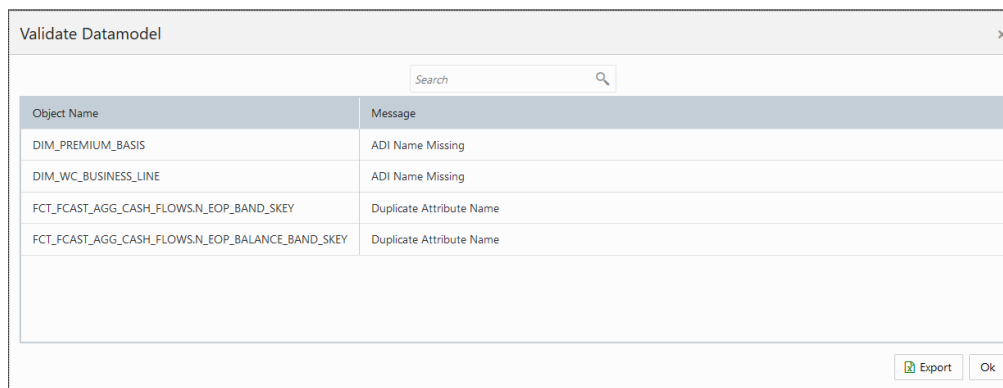
1. From the **Data Integration Hub Designer** window, select **Configure** and then select **Refresh ADI**. This **Refresh Application Data Interface** summary is displayed.

Figure 9: Refresh ADI Window



2. Click **Validate Datamodel** to validate the datamodel. If there are datamodel issues, the Validate Datamodel window is displayed. This validates and identifies the issues in values specified by the user-defined properties for the physical/logical view in the OFSAA Data Model. Once executed, the utility log errors/issues are identified.

Figure 10: Validate Datamodel Window



| Object Name | Message |
|--|--------------------------|
| DIM_PREMIUM_BASIS | ADI Name Missing |
| DIM_WC_BUSINESS_LINE | ADI Name Missing |
| FCT_FCAST_AGG_CASH_FLOWS.N_EOP_BAND_SKEY | Duplicate Attribute Name |
| FCT_FCAST_AGG_CASH_FLOWS.N_EOP_BALANCE_BAND_SKEY | Duplicate Attribute Name |

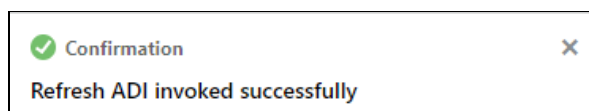
3. You can search for an Object Name or Message. On validation, you receive a message. See the Data Model Validation Messages table for information on each message.
4. Click **Export** to export the data model validation issues.
5. Verify the information and click **OK**.
6. Click **Start** to start the refresh of ADIs. The ongoing ADI refresh is displayed as follows:










| | | | | | |
|-------------|---------------|------------------------------------|------------------|----------------|-----------------|
| Run ID: 101 | Version Id: 0 | Start time: 2020-01-24 01:00:51.98 | Infodom: DIHINFO | End time: null | Status: Ongoing |
|-------------|---------------|------------------------------------|------------------|----------------|-----------------|

7. On successful invocation of ADI refresh, a message is displayed.

Figure 11: Refresh ADI Confirmation



8. If you need a detailed running log, click  download the log. A zip file is downloaded containing the detailed log for the execution. To view the log details, extract the log file from the zip folder.
9. You can check the status:
 -  Failed
 -  Successful
 -  Aborted
 -  Alert
 -  Warning
 -  Not Applicable

5.2.1 Abstraction of Model Changes for Data Movement / ETL Processing

DIH abstracts and automates many types of data model changes and their impact, minimizing manual steps to be undertaken. Refer to the following table for types of changes, DIH automation available, and manual actions to be taken, if needed.

Table 7: Abstraction of Model Changes for Data Movement / ETL Processing

| Scenario or Type of Change | Description | User Action |
|---|--|--|
| When the only logical name of an attribute is changed | ADI Refresh updates the logical name in the DIH repository. | No action is expected, changes are reflected automatically in connector/ADI. |
| When the only description of an attribute is changed | ADI Refresh updates the description in the DIH repository. | No action is expected, changes are reflected automatically in connector/ADI. |
| When the only logical name of an attribute is changed | ADI Refresh updates the domain in the DIH repository. | No action is expected, changes are reflected automatically in connector/ADI. |
| When both the logical name and domain of an attribute are changed | ADI Refresh updates the logical name and domain in the DIH repository. | No action is expected, changes are reflected automatically in connector/ADI. |
| When the physical name of an attribute is changed | ADI Refresh updates the physical name in the DIH repository. | Perform "Refresh Target Data Store", and re-publish Connectors by first unpublishing and then publishing them. |
| When the data type of an attribute is changed | ADI Refresh updates the data type in the DIH repository. | Perform "Refresh Target Data Store", and re-publish Connectors by first unpublishing and then publishing them. |
| When the precision/scale of an attribute is changed | ADI Refresh updates the precision/scale in the DIH repository. | Perform "Refresh Target Data Store", and re-publish Connectors by first unpublishing and then publishing them. |
| When the physical name of an entity is changed | ADI Refresh updates the physical name in the DIH repository. | Perform "Refresh Target Data Store", and re-publish Connectors by first unpublishing and then publishing them. |
| When the logical name (OFSAA Data Interset or SubType Name) of an entity is changed | ADI Refresh updates the logical name in the DIH repository. | No action is expected, changes are reflected automatically in connector/ADI. |

5.2.2 Handling Model Changes with Impact on Data Movement / ETL Processing

The following types of model changes impact data movement performed through DIH.

- One or more entities already configured for data movement in DIH is dropped
- One or more columns already configured for data movement in DIH is dropped

If these types of changes are encountered, the ADI Refresh process displays the execution status as “Impact Identified”. In such cases, the affected data movement definitions and entities, attributes, or both, the absence of which impacts data movement / ETL processing. These are listed under the “Alerts” tab, as displayed in the following window.

Figure 14: Refresh ADI Alerts Tab



The following model changes impact the connectors during ADI refresh.

Table 8: Handling Model Changes with Impact on Data Movement / ETL Processing

| Message | Action |
|--|---|
| The table which is dropped is already used in the connector | Unpublish and remove the ADI from the connector. |
| The column which is dropped is already used in the connector | Unpublish and remove the attribute references from the connector. For an Insert type connector, remove the attribute reference from mapping and truncate filter expression. For an Extract type connector, remove the attribute reference from the filter, join, lookup, derived column, mapping, and aggregation components. |

5.2.3 Data Model Validation Messages

DIH has a built-in mechanism to validate the data model uploaded to OFSAA before ADI Refresh. This mechanism verifies whether metadata relevant to DIH – entity classification, sub-type name, and so on is available and valid for use. The following table details the messages that DIH posts upon failure of verification steps, underlying reasons, and mechanisms for resolution:

Table 9: Data Model Validation Messages

| Message | Reason | Resolution |
|-------------------------------|---|---|
| Table Classification Missing | User-defined Property “OFSAA Data Interface Class” is not specified in the logical view of the table in the OFSAA Data Model. | Specify the value for User-Defined Properties in the OFSAA Data Model in ERWIN. |
| Sub Type Name Missing | User-defined Property “OFSAA Data Interface Sub-Type” is not specified in the logical view of the table in the OFSAA Data Model. | Specify the value for User-Defined Properties in the OFSAA Data Model in ERWIN. |
| Duplicate ADI Name | User-defined Property “OFSAA Data Interface Name” must be different for the specified tables. | Specify a unique value for OFSAA Data Interface Name UDP in the OFSAA Data Model in ERWIN. |
| No enabled application mapped | The application user-defined properties for all columns of the table does not have the value as “DL-MAN” or “DL-OPT” | Specify the “DL-MAN” or “DL-OPT” value for the application of user-defined properties. |
| ADI Name Missing | User-defined Property “OFSAA Data Interface Name” is mandatory and is not specified in the logical view of the table in the OFSAA Data Model. | Specify the value for User-Defined Properties in the OFSAA Data Model in ERWIN. |
| Invalid Table Classification | User-defined Property “OFSAA Data Interface Class” can have the value as R, S, or D only. | Specify a correct value for the mentioned user-defined property. |
| Invalid Subtype Name | User-defined Property “OFSAA Data Interface Sub-Type” is specified when there is no subType for ADI. | OFSAA Data Interface Sub-Type UDP is applicable when there are multiple subtypes for a given ADI. Leave it blank or the same as ADI name otherwise. |

5.3 Configuring System Parameters

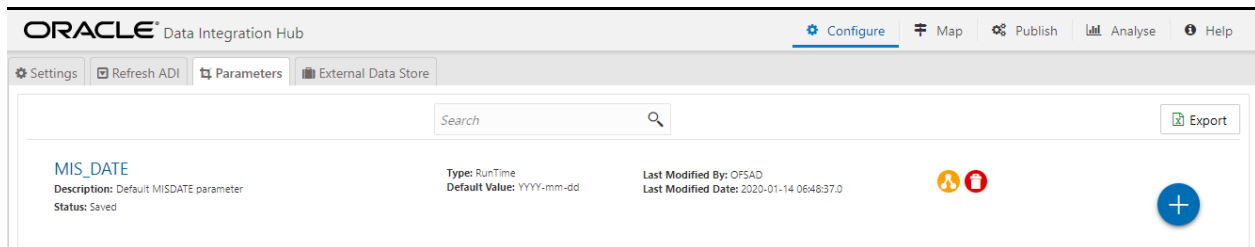
System Parameters are constant-value, run-time, or current-date variables intended for use with DIH. Apart from a seeded set of System Parameters, you can add, modify, or remove them as needed.


5.3.1 Understanding the Parameters Window

To access the Parameters window, follow these steps.

1. From the **Data Integration Hub Designer** window, select **Configure** and then select **Parameters**. The **Parameters Summary** is displayed.

Figure 15: Parameters Window



2. You can make use of the [Search](#) option to search for a specific Source.
3. Click **Export**. The List of Parameters are exported to an Excel sheet with the following information:
 - a. Parameter IDs
 - b. Parameter Name
 - c. Description
 - d. Type
 - e. Value
 - f. Default Value
 - g. Date Format
 - h. Status
 - i. Last Modified By and
 - j. Last Modified Date
4. Click  to create a Parameter. For more information, see Defining a Parameter section.

5.3.2 Fields in Parameters Window

Fields displayed in the Parameters window are explained in the following table.

Table 10: Fields in Parameters Window

| Fields | Description |
|----------------|--|
| | Fields marked in red asterisk(*) are mandatory |
| Parameter Name | The name for the placeholder that you want to define. For example, MISDATE, which can be used as a placeholder for Date. |

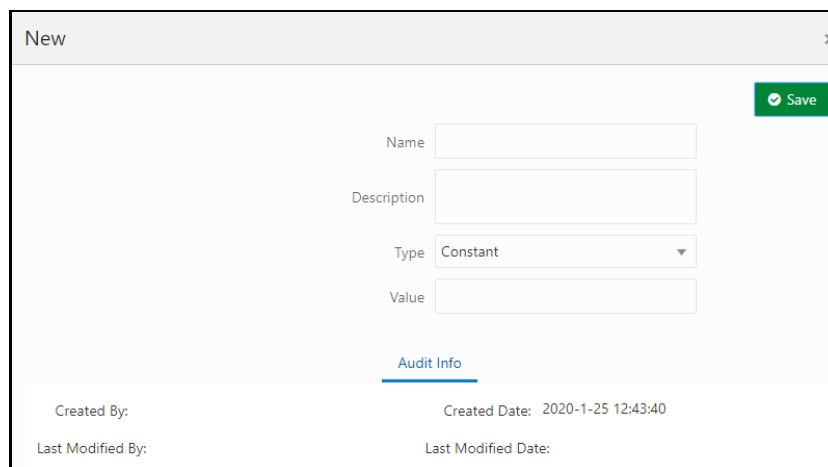
| Fields | Description |
|-----------------------|--|
| | Fields marked in red asterisk(*) are mandatory |
| Parameter Description | The description for the parameter you want to define. In this example, the description can be, "MISDATE can be used to substitute the date values for each day, dynamically, in mmddyyyy format." |
| Parameter Type | There are three parameter data types: <ul style="list-style-type: none"> • Constant data type is selected for substituting a constant value. • RunTime data type is selected for substituting a value, dynamically, in run time. In the example that is used here, MISDATE can be selected as Run Time because it is used to make a substitution dynamically. • CurrDate data type is selected for substituting a value as Current System Date. |
| Value | Only for constant types. Holds the actual value of the parameter. |

5.3.3 Defining a Parameter

To define a new Parameter, follow these steps:

1. Click **Add**  to define a parameter on the **Parameters Summary**. The **Parameters** window is displayed.

Figure 16: New Parameter Definition



The screenshot shows a 'New' window for defining a parameter. It includes the following fields and elements:

- Name:** A text input field.
- Description:** A text input field.
- Type:** A dropdown menu currently set to 'Constant'.
- Value:** A text input field.
- Save:** A green button with a checkmark icon in the top right corner.
- Audit Info:** A section at the bottom showing:
 - Created By:
 - Created Date: 2020-1-25 12:43:40
 - Last Modified By:
 - Last Modified Date:

2. Enter the fields, which are explained in the [Fields and their Description](#) section.
3. Click **Save**.
4. The Audit Trail section at the bottom of the window displays the information of the parameter created.

5.3.4 Modifying and Viewing a Parameter

You can edit or view an existing Parameter, other than Parameters which are in Published status.

NOTE You cannot edit parameters in Published status.

To edit or view a parameter, follow these steps:

1. To edit or view a Parameter, you can select the required parameter from the **Parameters Summary**.
2. The details of the selected Parameter are displayed. You can modify or view the details.
3. Only the Parameter description, Parameter Type, and Value / Default Value / Date Format can be edited on this window. Update the required details.
4. Click **Save** to save the changes made.

5.3.5 Deleting a Parameter

To delete an existing parameter, follow these steps:

1. On the **Parameters Summary**, click **Delete** . A confirmation dialog box is displayed.
2. Click **Yes**. The Parameter details are deleted.

NOTE Delete is enabled only in the following cases:

- If the parameter is not in Published status.
- If the parameter is not used by any higher object, for example, Connector/EDD.
- If the parameter is pre-seeded.

5.3.6 Unpublishing a Parameter

You can unpublish a parameter only when all the following conditions are met:


- The parameter is in Published status.
- All the higher objects using the parameter are unpublished, for example, Connector/EDD.

To unpublish a parameter, follow these steps:

1. Select the required parameter from the **Parameter Summary**. The details of the selected parameters are displayed.
2. Click **Unpublish**.

NOTE Parameters are published automatically by the system whenever the higher objects (EDD/Connector) which are using it are published.

5.3.7 Dependency

Clicking **Dependency**  lists where the entire parent Parameter has a dependency.

5.3.8 Search and Filter

The Search and Filter option in the UI helps you to find the required information. You can enter the nearest matching keywords to search and filter the results by entering information in the search box. You can search for a parameter using either the name, description, status, or type.

For example, enter the Parameter keyword as 'ODI' in the search box. The entire Parameter name with ODI is listed.

5.3.9 Parameters in EDD and Connectors

For information on parameters in EDD and Connectors, see sections Parameters in EDD Definition and Parameters in Connector respectively.

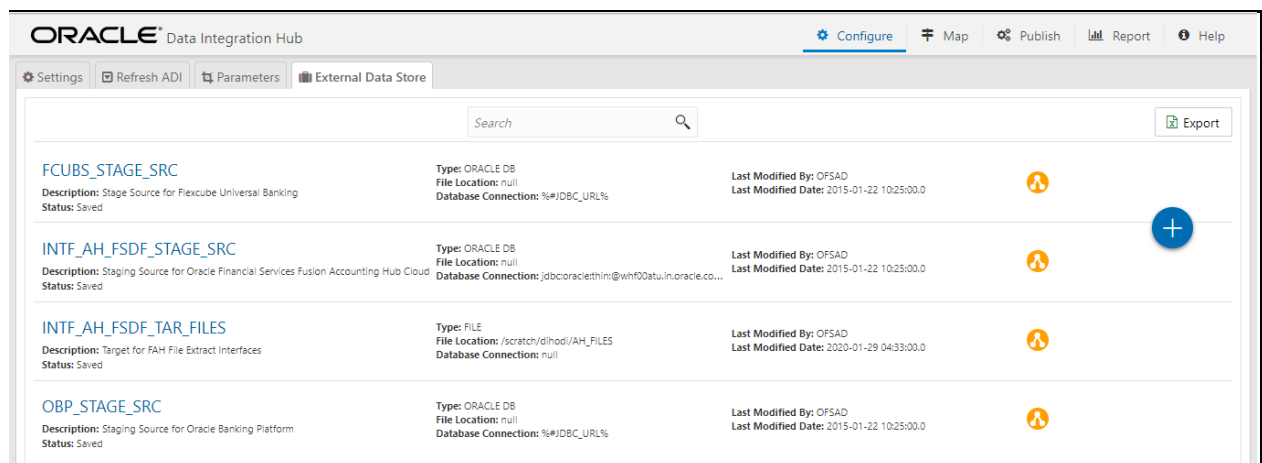
5.4 Configuring External Data Sources

DIH supports data ingestion from relational databases (Oracle Database, IBM DATABASE/2, Microsoft SQL Server, Sybase, and Teradata), files (XML, EBCDIC, and ASCII), and Big Data (Hadoop, HDFS, and Hive). Data stores of each of these types are registered with DIH, configured as External Data Sources (EDS). Also, DIH supports the extraction of data as ASCII files, which can also be defined as EDS.


To understand the **External Data Store** window, follow these steps:

1. From the **Data Integration Hub Designer** window, select **Configure** and then select **External Data Store**. The **External Data Store Summary** is displayed.

Figure 17: External Data Store Summary



2. In the **Source Systems** section of the External Data Store Summary, you can define, edit, and delete a source.
3. You can make use of the [Search](#) option to search for a specific Source.
4. Click **Export**. The List of EDSs are exported to an Excel sheet with the following information:
 - a. EDS IDs
 - b. EDS Name

- c. Description
 - d. EDS Type
 - e. JDBS URL
 - f. File Location
 - g. Status
 - h. Last Modified By and
 - i. Last Modified Date
5. Click  to create an EDS. For more information, see section Defining an External Data Store.

5.4.1 External Data Store Fields

The following table describes the fields in the External Data Store window.

Table 11: Fields in External Data Store

| Fields | Description |
|-------------|--|
| | Fields marked in red asterisk(*) are mandatory |
| Name | Enter the name of the Source. |
| Description | Enter a description for the Source. |
| Type | <p>The following source type for information on the additional fields specific to them are available:</p> <ul style="list-style-type: none"> • DB2 Type • EBCDIC Type • File Type • HDFS Type • HIVE Type • Oracle Type • SQL Server Type • Sybase Type • Teradata Type • XML Type |

5.4.1.1 DB2 Type

IBM DB2 is a family of database server products. These products support the relational model.

When you select EDS Type as DB2, the following fields are displayed.

Table 12: Fields in DB2

| Field | Description | Comments |
|----------|----------------------------------|--|
| JDBC URL | Provide the URL of the Database. | Example: jdbc:db2://<hostname>[:<port>]/<database> |
| User ID | Enter the User ID. | |
| Password | Enter the password. | |
| Schema | Enter the schema name. | |

5.4.1.2 EBCDIC Type

Extended Binary Coded Decimal Interchange Code (EBCDIC) File is a binary code for alphabetic and numeric characters.

When you select EDS Type as EBCDIC, the following fields are displayed.

Table 13: Fields in EBCDIC

| Field | Description | Comments |
|---------------|--|--|
| File Location | Enter the absolute path of the data file landing area. | ODI agent must be available and running in the server from where the data file is located. |

5.4.1.3 File Type

American Standard Code for Information Interchange (ASCII) File is a character-encoding scheme.

When you select EDS Type as File, the following fields are displayed.

Table 14: Fields in ASCII

| Field | Description | Comments |
|--------------------|--|--|
| File Location | Enter the absolute path of the data file landing area. | ODI agent must be available and running in the server from where the data file is located. |
| Encryption at Rest | If a source file is encrypted or a destination file should be encrypted upon data extraction needs, choose the “Encryption at Rest” option and enter the Encryption Key File Path. | <ul style="list-style-type: none"> • DIH must have access to the source file landing area. • The UNIX user, which is used for starting the agent, must have execution permission to DMT utility. Example: /landingzone/inputfiles |

5.4.1.4 HDFS Type

Hadoop Distributed File System (HDFS) is an open-source and fundamentally a new way of storing and processing data. It enables distributed processing of huge amounts of data across industry-

standard servers that both store and process the data and can scale without any limits HDFS is Big Data in a raw format.

When you select EDS Type as HDFS, the following fields are displayed.

Table 15: Fields in HDFS

| Field | Description | Comments |
|---------------|----------------------------------|--------------------------------------|
| JDBC URL | Provide the URL of the Database. | Example: hdfs://<Host>:<Port> |
| File Location | Enter the file location. | |

5.4.1.5 HIVE Type

HIVE provides a mechanism to project structure onto the data in Hadoop and Big data in a tabulated format.

When you select EDS Type as Hive, the following fields are displayed.

Table 16: Fields in HIVE

| Field | Description | Comments |
|--------------------------------|---|--|
| JDBC URL | Provide the URL of the Database. | Example: jdbc:hive://<Host>:<Port>/<Schema> |
| Driver | Enter the driver for the HIVE datastore. | For example, to connect to the Cloudera Hive server with JDBC 4.0 data standards, specify “com.cloudera.hive.jdbc4.HS2Driver” as a driver. See the Cloudera document for more information about Cloudera JDBC drivers. |
| If Kerberos is enabled: | | |
| Principal User Name | Enter the Principal User Name. | |
| Conf File Path | Enter the Kerberos Conf File path. | |
| Key Tab Path | Enter the path of the Key Tab file, generated for the principal user. | |
| If Kerberos is enabled: | | |
| User ID | Enter the User ID. | |
| Password | Enter the Password. | |

5.4.1.6 Oracle Type

An Oracle database is a collection of data treated as a unit. The purpose of a database is to store and retrieve related information.

When you select EDS Type as Oracle, the following fields are displayed.

Table 17: Fields in Oracle

| Field | Description | Comments |
|-----------------------|--|---|
| JDBC URL | Provide the URL of the database. | Example: jdbc:oracle:thin:@//<hostname>:<port>/<service_name> |
| User ID | Enter the User ID. | |
| Password | Enter a password. | |
| Schema | Enter the Schema name in the upper case. | |
| Encryption in Transit | Choose this option if you want the data to be encrypted while reading from the source. | |

5.4.1.7 SQL Server Type

Microsoft SQL Server is a relational database management system. It is a software product with the primary function of storing and retrieving data as requested by other software applications, which may run either on the same computer or on another computer across a network.

When you select EDS Type as SQL Server, the following fields are displayed.

Table 18: Fields in SQL Server

| Field | Description | Comments |
|----------|----------------------------------|---|
| JDBC URL | Provide the URL of the database. | Example: jdbc:sqlserver://<hostname>\SQLExpress |
| User ID | Enter the User ID. | |
| Password | Enter the password. | |
| Schema | Enter the Schema name. | |

5.4.1.8 Sybase Type

Sybase produces software to manage and analyze information in relational databases.

When you select EDS Type as Sybase, the following fields are displayed.

Table 19: Fields in Sybase Type

| Field | Description | Comments |
|-----------------|----------------------------------|--|
| JDBC URL | Provide the URL of the database. | Example: jdbc:sybase:Tds:<hostname>:<port> |
| User ID | Enter the User ID. | |
| Password | Enter the password. | |
| Schema | Enter the Schema name. | |

5.4.1.9 Teradata Type

Teradata Corporation provides analytic data platforms, applications, and related services. Its products enable users to consolidate data from different sources and make the data available for analysis.

When you select EDS Type as EBCDIC, the following fields are displayed.

Table 20: Fields in Teradata

| Field | Description | Comments |
|----------|--------------------------------|---|
| JDBC URL | Enter the URL of the database. | Example: jdbc:teradata://<hostname> |
| User ID: | Enter the User ID. | |
| Password | Enter the password. | |
| Schema | Enter the Schema name. | |

5.4.1.10 XML Type

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents.

When you select EDS Type as XML, the following fields are displayed.

Table 21: Fields in XML

| Field | Description | Comments |
|---------------|--|---|
| File Location | Enter the absolute path of the data file landing area. | The ODI agent must be available and running in the server where the data file is located. |

5.4.2 Defining an External Data Store

To define a new source from **External Data Store Summary**, following these steps:


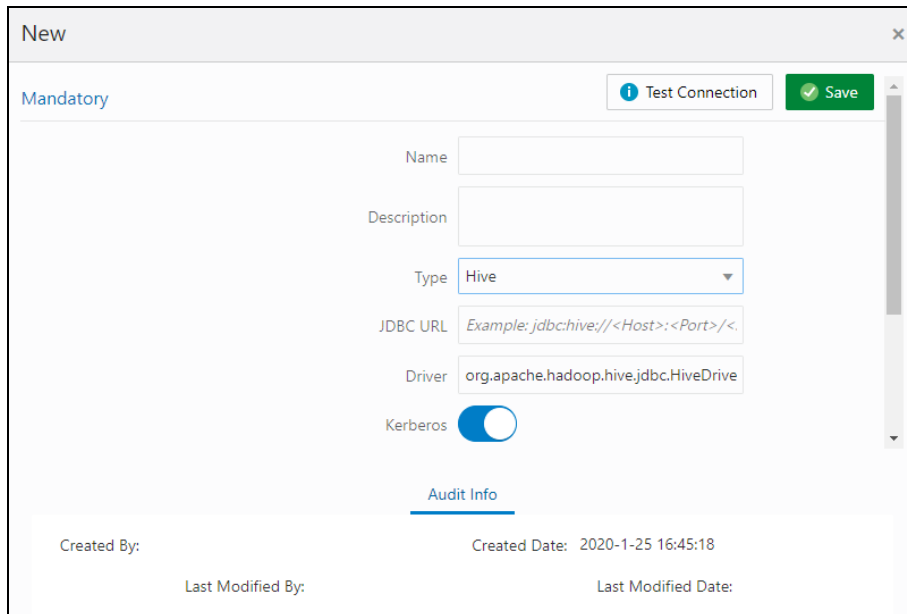
1. Click  to define a new **External Data Store** on the **External Data Store Summary**. The **External Data Store** window is displayed.

Figure 18: New External Data Store



2. Enter the values in the fields as described in the [External Data Store Fields](#) section.
3. The fields change depending on the Type option selected. For example, If Source Type is selected as File, the File Location field must be entered.
4. Click Test Connection to test the connection details (User ID/ Password) for the database types **DB2, HIVE, Oracle DB, SQL Server, Sybase, and Teradata**.
5. Enter these details and click Save.

5.4.3 Modifying and Viewing an External Data Store

To edit and view an external data store, follow these steps:


1. To edit or view an EDS, select the required EDS from the EDS Parameter Summary.

NOTE You cannot edit Published objects.

2. The Audit Info section at the bottom of the window displays the information of the source created.
3. EDS Name and Type cannot be edited.
4. Click Save to save the changes made.

5.4.4 Deleting an External Data Store

To delete an existing EDS, follow these steps:

1. On the **EDS Summary**, click **Delete** . A confirmation dialog box is displayed.
2. Click **Yes**. The EDS details are deleted.

- NOTE** Delete is enabled only in the following cases:
- If the EDS is not in Published status.
 - If the EDS is not used by any object.

5.4.5 Unpublishing an External Data Store

You can unpublish an EDS only when all the following conditions are met:


- The EDS is in Published status.
- All the higher objects using the EDS are unpublished, for example, Connector/EDD.

To unpublish an EDS, follow these steps:

1. Select the required EDS from the EDS summary. The details of the selected EDS are displayed.
2. Click **Unpublish**.

- NOTE** An EDS is published automatically by the system whenever the higher objects (EDD/Connector) using it are published.

5.4.6 Dependency

Clicking **Dependency**  lists where the entire parent EDS has a dependency. That is, you cannot delete a child file without deleting the parent file.

5.4.7 Search and Filter

The **Search and Filter** option in the UI helps you to find the required information. You can enter the nearest matching keywords to search and filter the results by entering information on the search box. You can search for a parameter using either the name, description, status, or type.

For example, enter the Parameter keyword as 'DRM' in the search box. All the EDS names with DRM are listed.

6 External Data Descriptors and Connectors

This chapter helps to configure and manage the External Data Descriptors and Connectors.

Topics:

- [External Data Descriptor](#)
- [Application Data Interface](#)
- [Connectors](#)

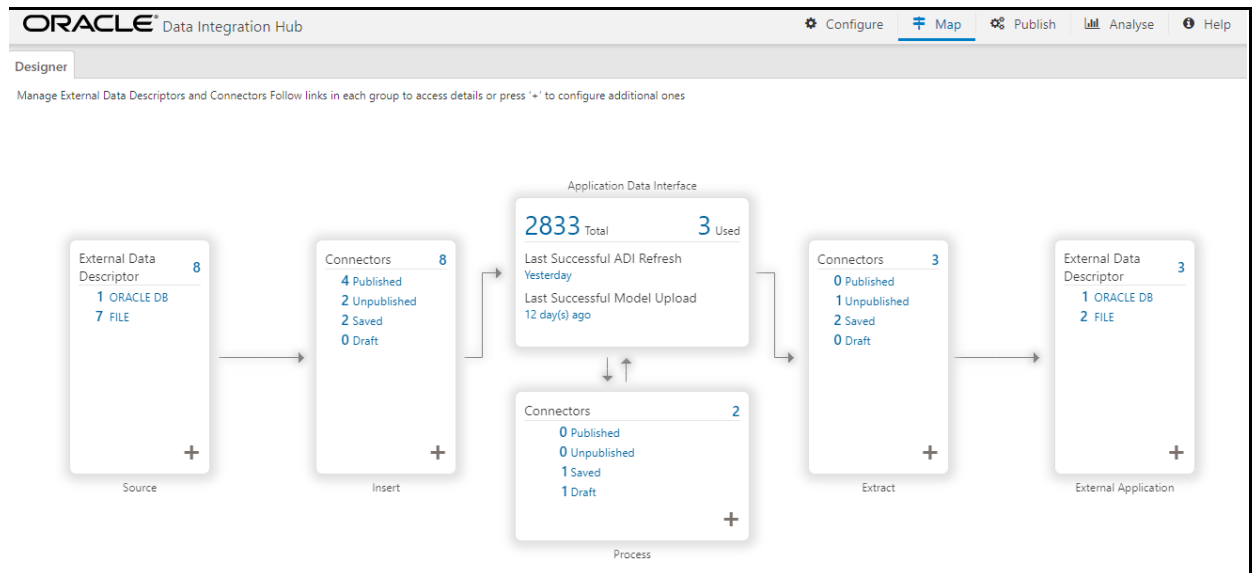
6.1 External Data Descriptor

External Data Descriptors (EDD) are definitions of specific data content from External Data Sources (EDS). Each EDS may have a number of EDDs defined against it. EDD definitions can be used for ingesting data into OFSAA Data Foundation or extracting data from it.

DIH also supports the usage of control information while ingesting data into the Data Foundation. Such control information may also be defined as EDDs.

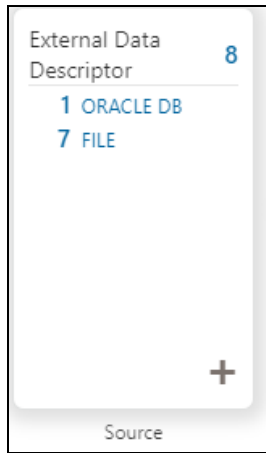
1. From the **Data Integration Hub Designer** window, to manage External Data Descriptors, click links in each group to access details.

Figure 19: DIH Designer Window

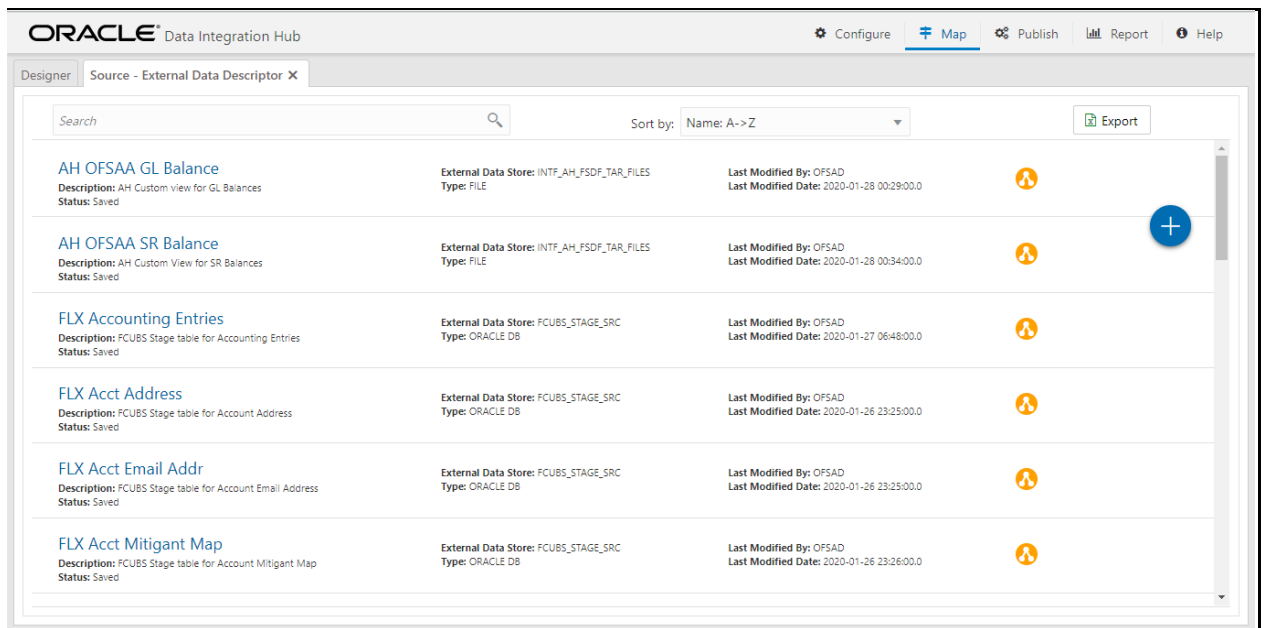


2. To manage **Source – External Data Descriptor**, click the links under Source.

Figure 20: Source – External Data Descriptor

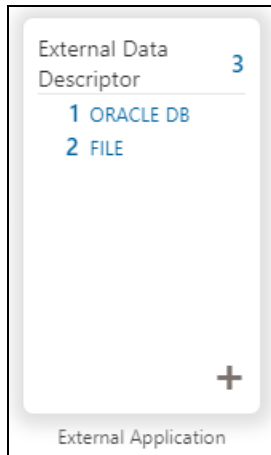


The **Source - External Data Descriptor** window is displayed.

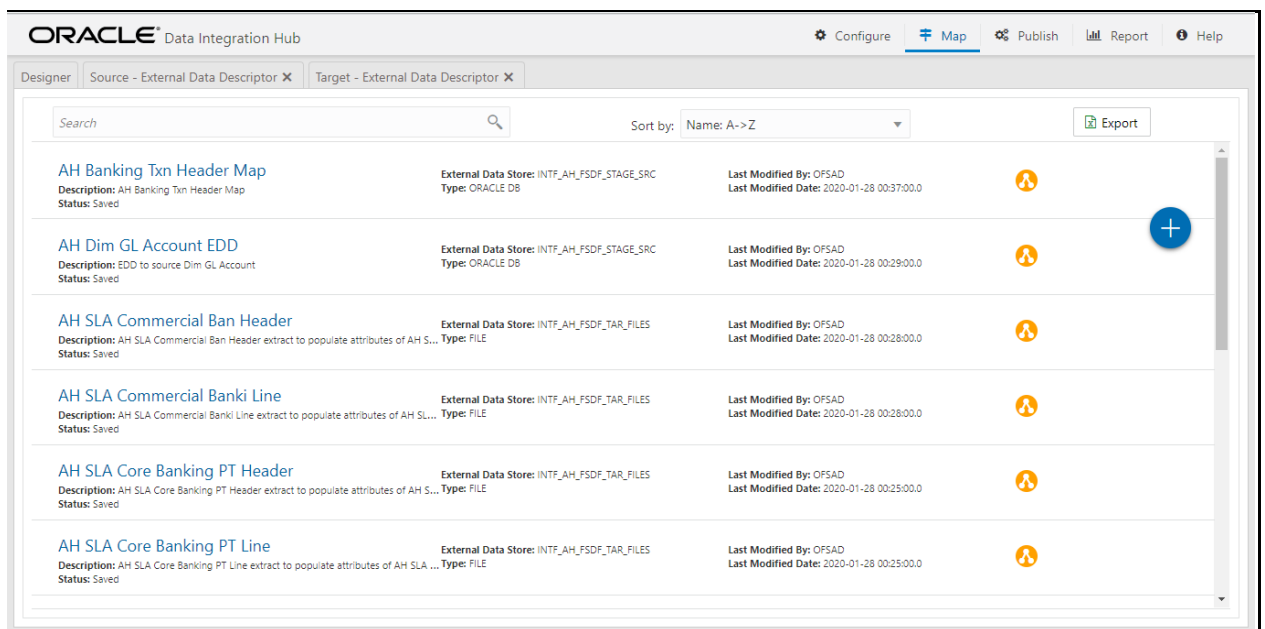




- To manage **Target – External Data Descriptor**, click the links under External Application.

Figure 21: Target – External Data Descriptor



The **Target - External Data Descriptor** window is displayed.



4. From the **Data Integration Hub Designer** window, click  **Source** or **Target EDD** windows. to configure additional EDDs or click **Add**  from the **Source or External Application**.
5. For more information, see Defining an External Data Descriptor.
6. Click Export. The List of EDDs are exported to an Excel sheet with the following information:
 - a. EDD IDs
 - b. EDD Name
 - c. Description
 - d. EDS Name
 - e. EDS Type

- f. Status
- g. Last Modified By and
- h. Last Modified Date

6.1.1 External Data Descriptor Fields


The following sections describe the fields in the External Data Descriptor window.

6.1.1.1 Data Tab

The following table describes the fields in the Data tab.

Table 22: Fields in Data Tab

| Fields | Description |
|------------------------|--|
| | Fields marked in red asterisk (*) are mandatory |
| Data File Name | <p>You can add multiple data files to an EDD.</p> <p>For example, you need to add the Term Deposits Contracts data file. There are Term Deposits Contracts data files for Retail as well as Corporate accounts. Therefore, to get both these details, you first add the Term Deposits Contracts data file for Retail accounts, such as <code>td_contracts%#MISDATE%_1.csv</code>, and as the next record, add Term Deposits Contracts data file for Corporate accounts.</p> <p>Example: <code>td_contracts%#MISDATE%_1.csv</code></p> |
| Record Delimiter | <p>The records are stored differently in different operating systems. The options available are:</p> <ul style="list-style-type: none"> • MS-DOS • Unix • No Record Delimiter • Other <p>For example, select Unix.</p> |
| File Format | <p>There are two options:</p> <ul style="list-style-type: none"> • Fixed Length: The file has records and columns with a fixed length. Each column has a predetermined and unchanging size, set when the record layout is designed, and the sum of the column sizes add up to the record size. • Delimited: There is a separation of the records and columns using a delimiter character like a comma, semicolon, hyphen, and so on. <p>For example, select Delimited.</p> |
| Text Qualifier | <p>A character that identifies a text. This is used when some characters exist within a text. Generally, double quotes are used, prefixed, and suffixed with text. This is optional.</p> |
| Skip Number Of Records | <p>Provide the number of records to be skipped. The records are skipped from the top. Generally, this is used to skip Headers.</p> |

| Fields | Description Fields marked in red asterisk (*) are mandatory |
|---|---|
| Decimal Separator | This mentions up to which decimal digit you want to view the result. |
| Read from template | A template contains all the values and is in Excel file format. If the template is not available, you must create it manually by clicking Add  , under the Source Data Elements tab. If the template is available, you can browse for the template. See the File EDD Template . You can also drop the template in the area “Drop template here or click to select “. |
| Select Template (* .xls, * .xlsx, * .CSV Files Only) | Click Browse and select the required template. |
| Data Elements | |
| Name | Name of the field in EDD. Example: Field name in a file or column name in a table. NOTE: The Field name of XML type must not be more than 25 characters and for others must not be more than 30 characters. |
| Type | This shows the Data type of the field. Example: String, Number, EBCDIC, and so on. |
| Length | This is applicable only for the EBCDIC format. This is the length of the EBCDIC data type. In the case of a file, it is length only. |
| Precision | This is used to specify the decimal point. Example: 10.3. |
| Format | Specify the format for columns of type date here. If left blank, a default format of DD/MM/YYYY is assumed and used. NOTE: <ul style="list-style-type: none"> • The default format is fixed and does not change with database or system language settings. • For data ingestion, enter the format in which date fields are provided. • For data extraction, enter the format in which extracted date fields must be recorded. For example, in the file extract, to represent the date, 31st January 2020, as “31/01/2020”, specify the format as DD/MM/YYYY for the date columns. |
| Record Type Code | This identifies the Record type in a file where Header, Trailer, and Data are of different record length and type. The values can be any string available in the text file. This value is only possible for the first field in a file. Example: The values can be DATA; CTRL to specify it is a control record. |

6.1.1.2 Control Tab

In the Control tab, DIH enables you to:

1. Reconcile between source data received and data loaded into OFSAA, using the control information available from the source system. This feature is used to implement record count validation or amount reconciliation from a source file. It also allows specifying a threshold for validation. The specified threshold is compared with the difference (record count or amount) to perform the validation. Only in case, the difference is more than the threshold value the execution fails. A detailed report on control information from both source and the final data loaded along with threshold breach is available in [Execution History](#).
2. The threshold can be absolute or percentage value. The connector execution process considers the threshold type while performing reconciliation.
3. Generate control information for File extracts from OFSAA. Specifying control for extract EDDs generates control details, based on the configuration in the EDD. Both record count and amount value (sum, average, max, and so on) for specific columns are recorded into the control file.

NOTE This option is applicable only for File type EDDs (ASCII and EBCDIC).

The following fields must be configured to achieve this functionality.

Table 23: Fields in Control Tab

| Fields | Description Fields marked in red asterisk(*) are mandatory |
|---|---|
| When Separate File is selected as Yes . | |
| File Name | Specify the name of the file. |
| File Format | <p>There are two options:</p> <ul style="list-style-type: none"> • Fixed Length: The file has records and columns with a fixed length. Each column has a predetermined and unchanging size, set when the record layout is designed, and the sum of the column sizes add up to the record size. • Delimited: There is a separation of the records and columns using a delimiter character like a comma, semicolon, hyphen, and so on. <p>In the previous example, select Delimited.</p> |
| Column Delimiter | <p>If the File Format is selected as Fixed Length, the Column Delimiter would by default be Other.</p> <p>If the File format is selected as Delimited, the following options are available in the drop-down list.</p> <ul style="list-style-type: none"> • Other • Space • Semicolon • Comma • Tab <p>In the previous example, select Comma.</p> |

| Fields | Description Fields marked in red asterisk(*) are mandatory |
|--|--|
| Record Type Code | Used to uniquely identify a record within a file. A Financial Institution sometimes provides files that have data and control records within the same file. In that case, to distinguish between data record and control record, the first field is Record Type. It has a specific value to identify that. Here, specify the value that identifies the Data. Values can be 'DATA' and so on. For the Control record, the value is specified under the Control tab. Only the first field of a file is used for Record Type. |
| Record Delimiter | The records are stored differently in different operating systems. The following options are available: <ul style="list-style-type: none"> • MS-DOS • Unix • No Record Delimiter • Other For example, select Unix. |
| Skip number of records | Provide the number of records to be skipped. The records are skipped from the top. Generally, this is used to skip Headers. |
| Text Qualifier | A character that identifies a text. This is used when some characters exist within a text. Generally, double quotes are used, prefixed, and suffixed with text. This is optional. |
| Decimal separator | Specify up to which decimal digit you want to view the result. |
| Record Type Length | The length of the record type value to pick up the correct record. For example, if the control record is "DATATotal Records400" and DATA is the Record type, the length is '4'. This is applicable only for Control records that are of Fixed length. |
| Control Name Length | Based on the previous example, the Control name is "Total Records". Hence, the Control Name Length is '13'. |
| Control Value Length | Based on the previous example, the Control value is 400. Hence, the length of the control value is '3'. |
| When Separate File is selected as No . | |
| Record Type Code | Used to uniquely identify a record within a file. A Financial Institution sometimes provides files that have data and control records within the same file. In that case, to distinguish between data record and control record, the first field is Record Type. It has a specific value to identify that. Here, specify the value that identifies the Data. Values can be 'DATA' and so on. For the Control record, the value is specified under the Control tab. Only the first field of a file is used for Record Type. |
| Control Value Length | Based on the previous example, the Control value is 400. Hence, the length of the control value is '3' |
| Control Name Length | Based on the previous example, the Control name is "Total Records". Hence, the Control Name Length is '13'. |
| Controls | |

| Fields | Description |
|-------------------------|--|
| | Fields marked in red asterisk(*) are mandatory |
| Control Name | Specify the name of the control. |
| Aggregation Method | Select either Aggregation Method or Count . The supported aggregation methods are as follows: <ul style="list-style-type: none"> • Min • Max • Average • Sum |
| Aggregation Column Name | Select the column on which the aggregation method is applied. NOTE: For count, no column needs to be selected. |
| Threshold Type | This field is optional. There are two selections of threshold, percentage, or absolute. If the percentage is selected, the reconciliation difference in percent is matched against this threshold value. If absolute is selected, the absolute percent difference is matched against this threshold value. |
| Threshold Value | Specify the difference value in percent or absolute. |

6.1.1.3 Transformation Tab

The following table describes the fields in the Transformation tab.


Table 24: Fields in Transformation Tab

| Fields | Description |
|--|--|
| | Fields marked in red asterisk(*) are mandatory |
| Transformation Type | A drop-down listing different types of transformation supported. Currently, the only Aggregation is supported. |
| If the Transformation Type is selected as None : | |
| Derived Data Elements | |
| Name | Name of the derived field in EDD. NOTE: Field names must not be more than 30 characters. |
| Type | Shows the Data type of the field. Example: Varchar2, Number, Date, and so on. |

| Fields | Description |
|---|--|
| | Fields marked in red asterisk(*) are mandatory |
| Expression | When you select the 'Add option', the Specify Expression window is displayed. Here, you can select the required Entities, Functions, and Operators. That is, you can write your expression. Enter the field name and click OK. Now the newly created field name is listed. |
| If the Transformation Type is selected as Aggregation : | |
| Derived Data Elements | |
| Name | Name of the derived field in EDD. Note: Field names must not be more than 30 characters. |
| Type | This shows the Data type of the field. Example: Varchar2, Number, Date, and so on. |
| Expression | When you select the 'Add option', the Specify Expression window is displayed. Here, you can select the required Entities, Functions, and Operators. That is, you can write your expression. Enter the field name and click OK. Now the newly created field name is listed. |
| Aggregation Properties | |
| Group By | This is available when Aggregation is selected. |
| Having | This is available when Aggregation is selected. |

6.1.2 Defining an External Data Descriptor

To define a new EDD from the DIH Designer window, follow these steps:

1. From the **Source** or **External** application, click Add . The **External Data Descriptor** new window is displayed.
2. In the External Data Store Name section, select **Data Source** from the drop-down list. The Data Source is the Source you had created. In this example, it is, DRM_SRC_FILES. The values in Defining an External Data Store example are used. The description comes up automatically.

NOTE The fields in the External Data Descriptor Specification Details section change as per the Source Code selected.

For HDFS data, the working date format is YYYY-MM-DD.

For the Hive table, the working date format is YYYY-MM-DD.

For the Sybase source database, the date data type is not supported. It must be a timestamp.

Figure 22: External Data Descriptor Data Tab

External Data Descriptor Details

External Data Store Name: INTF_AH_FSDF_TAR_FILES
 External Data Store Description: Target for FAH File Extract Interfaces

External Data Descriptor Name: AH OFSAA GL Balance
 External Data Descriptor Description: AH Custom view for GL Balances

Data Control Transformation

Data File Name: ah_gl_balance.csv
 File Format: Delimited
 Column Delimiter: Comma
 Skip Number Of Records: 1
 Read From Template:

Record Delimiter: Unix
 Text Qualifier:
 Decimal Separator:

| Order | Name | Type | Length | Scale | Format | Record Type Code |
|-------|-----------------|--------|--------|-------|--------|------------------|
| 1 | ACCOUNTING_DATE | DATE | 0 | 0 | null | Record Type Code |
| 2 | PERIOD_NAME | STRING | 4000 | 0 | null | Record Type Code |
| 3 | IFDGFR_NAME | STRING | 4000 | 0 | null | Record Type Code |


3. Click **Reload/Refresh**  to reload/refresh the data elements.
4. Enter the values in the fields as described in the [Fields and their Description](#) section.

Figure 23: Read From Template

Read From Template

Select Template (*.xls,*.xlsx,*.csv Files Only)

Data Elements

Drop template here or click to select

5. If data needs to be reconciled post-loading, then click the **Control** tab. In this version, only the Number of Records controls is possible.

Figure 24: External Data Descriptor Controls Tab

Separate File

File Name

Column Delimiter: Comma

Record Delimiter: Unix

Text Qualifier

File Format: Delimited

Record Type Code

Skip Number Of Records

Decimal Separator

| Order | Name | Aggregation Method | srcCol2 | Absolute | Threshold Value |
|-------|------|--------------------|---------|----------|-----------------|
| 1 | Name | Aggregation Method | srcCol2 | Absolute | Threshold Value |

Page 1 of 1 (1 of 1 items) | K < 1 > X

Audit Info

Created By: OFSAD

Created Date: 2020-01-14 05:18:42.0

6. To transform the EDD, click the **Transformation** tab.

Figure 25: External Data Descriptor Transformation Tab

ORACLE Data Integration Hub

Configure | Map | Publish | Report | Help

Designer | Source - External Data Descriptor X | AH OFSAA GL Balance X

External Data Descriptor Details

External Data Store Name: INTF_AH_FSDf_TAR_FILES

External Data Store Description: Target for FAH File Extract Interfaces

External Data Descriptor Name: AH OFSAA GL Balance

External Data Descriptor Description: AH Custom view for GL Balances

Unpublish | Save As Draft | Save

Data | Control | Transformation

Derived Data Elements

| Order | Name | Type | Expression |
|-------|------|------|------------|
| 1 | Name | DATE | |

+



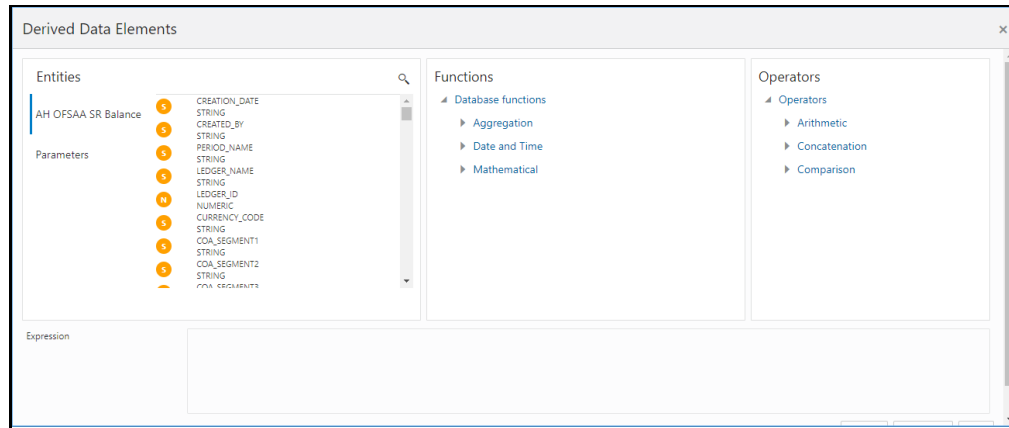


7. You can add derivation to data elements of the EDD.
- Click **Add**  to create derived data elements.
 - To edit the derived data element, click **Edit** . The **Expression** window is displayed.
 - The expression can be specified using the data elements defined in the Data tab and functions.

Figure 26: Derived Data Elements Window



- d. To delete the derived data element, click **Delete** .
8. Click the **Transformation** tab and select the **Transformation Type**.
 - a. Select **Aggregation** and click **Edit**  to view **Expression Window**.
 - b. Specify the Group by clause and Having expression, if applicable.
 - c. Define Derived Data Elements for the field to be aggregated under the previous tab.

NOTE

It is applicable only in the case of DB2, Oracle, Sybase, My SQL, and Hive.

9. Click **Save**.

6.1.3 Modifying and Viewing an External Data Descriptor

You can edit or view an existing EDD, other than EDDs in Published status.

NOTE

You cannot edit EDDs in Published status.

To edit or view an EDD, follow these steps:


1. Select the required EDD from the EDD Summary.
2. The details of the selected EDD are displayed. You can modify or view the details.
3. Update the required details.
4. Click **Save** to save the changes made.
5. Click **Save as Draft** to save and update later. The status shows as Draft.

6.1.4 Deleting an External Data Descriptor

This option only checks the higher-order object. That is, if the order has a dependency, you cannot delete it unless the dependency is removed.

For example, assume EDD is used in Connector. Then, unless the Connector is deleted, the used EDD cannot be deleted.

To delete an existing EDD, follow these steps:

1. On the **EDD Summary**, click **Delete** . A confirmation dialog box is displayed.
2. Click **Yes**. The EDD details are deleted.

NOTE Delete is enabled only in the following cases:
 If the EDD is not in Published status.
 If the EDD is not used by any object.

You can unpublish an EDD only when all the following conditions are met:


- The EDD is in Published status.
- All the connectors using the EDD are unpublished.

To unpublish an EDD, follow these steps:

1. Select the required EDD from the **EDD Summary**. The details of the selected EDD are displayed.
2. Click **Unpublish**.

NOTE The EDD gets published automatically by the system whenever the higher objects (Connector) using it are published.

6.1.5 Dependency

Clicking **Dependency**  lists where the entire parent EDD has a dependency.

6.1.6 Search and Filter

The Search and Filter option in the UI helps you to find the required information. You can enter the nearest matching keywords to search and filter the results by entering information on the search box. You can search for an EDD using either the name, description, status, or type.

For example, enter the EDD keyword as 'AATB_ACCT' in the search box. The entire EDD name with AATB_ACCT is listed.

6.1.7 Parameters in EDD Definition

While defining an EDD, the parameter can be used as a placeholder in a data filename.

For example:

Consider a table with two columns, such as Account number and Balance.

Table 25: Example showing Account Number and Balance

| Account Number | Balance |
|----------------|---------|
| A1 | 1000 |
| A2 | 1000 |
| A3 | 1000 |
| A1 | 1000 |
| A2 | 1500 |
| A3 | 1500 |

In this example, a customer has three accounts (A1, A2, and A3).

The customer has deposited different amounts on January 1st and 2nd 2014. The CSV data files can be created for those two dates as follows:

- The account transaction for January 1st, 2014 is saved as `td_contracts_/01012014/.csv`
- The account transaction for January 2nd, 2014 is saved as `td_contracts_/01022014/.csv`

If a parameter, `MISDATE`, is defined as a runtime, this can be used as a placeholder that substitutes date in `mmddyyyy` format. That is, the data filename can be mentioned as `td_contracts_%#MISDATE%.csv`. When this file is called, it substitutes the date in the file name, dynamically, in the runtime.

Parameters Data Types need not always be runtime. They can be Constants or values like Current Date, which can also be used to substitute a value in a data filename.

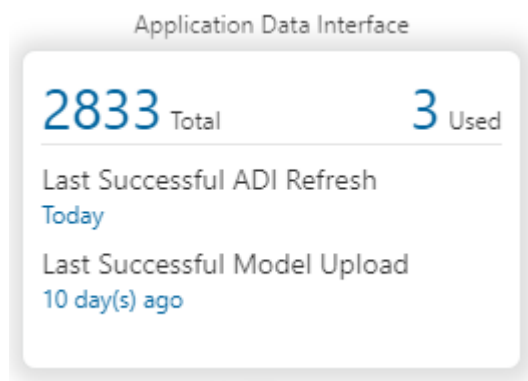
6.2 Application Data Interface

This tile displays the total number of ADIs that are available in the setup and the number of used ADIs in connectors.

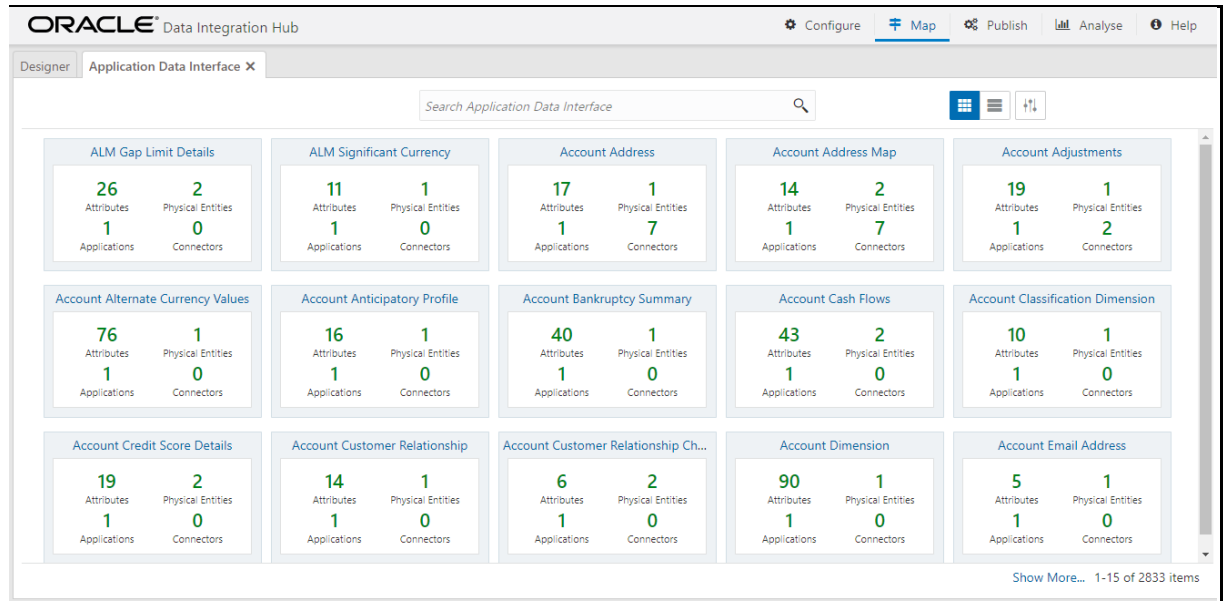
To view the total number of ADIs that are present and used, follow these steps:

1. Click **Total** or **Used** on the **Application Data Interface**.

Figure 27: Application Data Interface Summary Window



The **Application Data Interface Summary** window is displayed:






2. Click **Analyse** to view the Mapping Report for that particular ADI. For more information, see the **Error! Reference source not found.** section.
3. You can view the summary details of all the ADIs that are present or used in either Card view  or List view .
4. The search bar helps you to find the required information. You can enter the nearest matching keywords to search and filter the results by entering information on the search box. You can search for an ADI name using either the name or description.
5. Click  to filter the ADI. The RHS displays the applications you can select to filter. Select the required application and the feature.

Figure 28: Application Selection Drawer

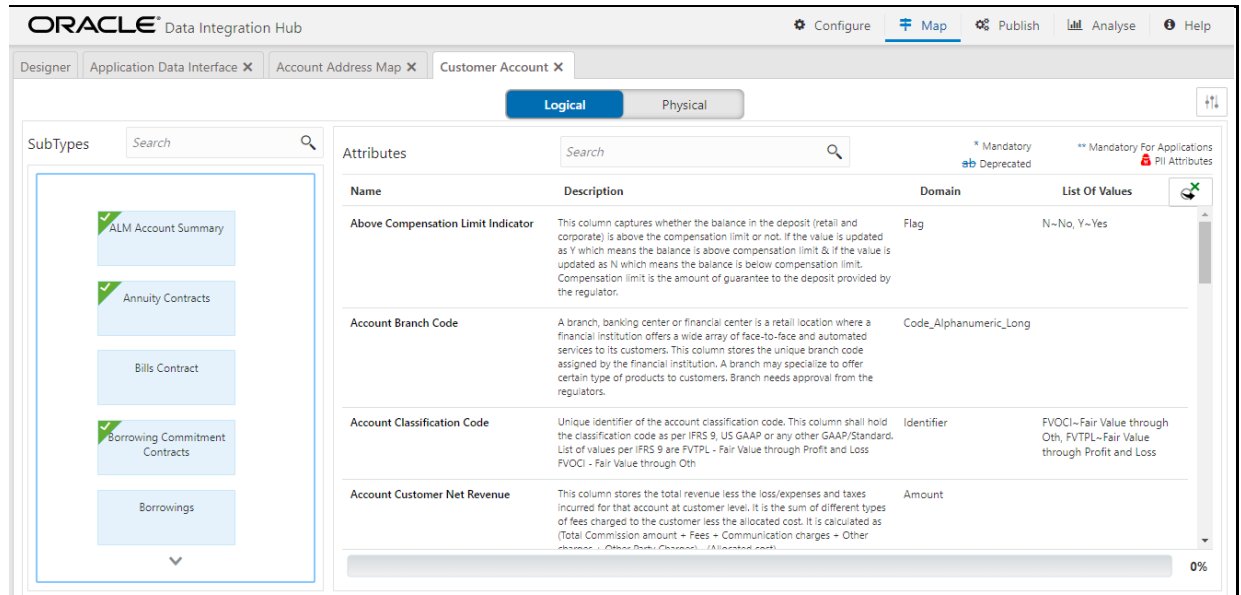
The screenshot shows a web interface titled "Applications" with several sections of filter options:

- Applications:** A list of checkboxes for "Liquidity Risk Management", "Asset Liability Management", "Basel Regulatory Capital Basic", "Basel Regulatory Capital IRB Approach", "Financial Services Data Foundation", "Fund Transfer Pricing", "GL Reconciliation Framework", and "Profitability Management".
- OFSAA Module:** A list of checkboxes for "Staging" and "Result".
- Domain:** A text input field with the placeholder text "Select Domain".
- Others:** A list of checkboxes for "Mandatory", "Valid For Applications", and "PII Attributes".


At the bottom right of the form, there are two buttons: "Reset" and "Apply".

6. Select the required application and then click **Apply**. The summary window displays the filtered ADIs.
7. Click **Reset** to deselect the filter options and clear the Subject area.
8. Depending on the ADI selected, there may or may not be additional subtype filters. Such as, for Transactions: Customer Account, there is a Product Class list as subtype filters available. You can choose one or more Product classes to filter the attributes listed below.
9. The selected ADI details are displayed. There are two views for each ADI:
 - **Logical View:** shows all the attributes and their associated description with additional information. For example, if the attribute is mandatory or not for the selected application, its domain and LOV (List of values) are possible for the particular attribute.
 - **Physical View:** shows the underlying physical table name of the selected ADI. Selecting the physical table name shows the mapping between the logical attribute name and its corresponding physical column name.
10. At any given time, you can switch between Logical and Physical Views.
11. In **Logical View**, you can see the attribute details as follows:
 - List with the logical name
 - Description
 - Domain
 - List of values

Figure 29: Logical View Window



12. Attributes contain:

- Sensitive information **PII** (Personally Identifiable Information) is displayed with  before the attribute.
- Deprecated attributes are shown as **strikethrough** text.
- Mandatory attributes have * next to them.
- Attributes that are mandatory for selected applications have ** next to them.

13. In **Logical View**, you can search for an attribute name or description.

14. The progress can also be viewed based on the filters applied to this window.

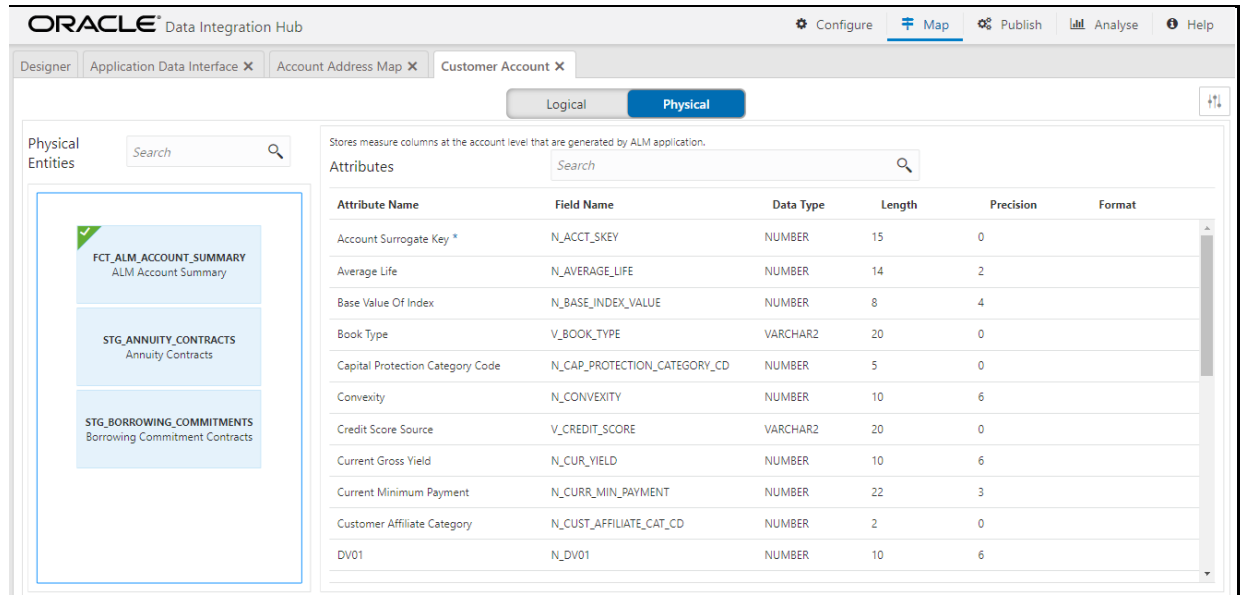
NOTE The percentage is calculated based on the number of attributes sourced versus the total number of attributes.

15. In **Physical View**, click the table name. You can view the attribute name, field name, data type, length, precision, and .format.

16. For example, in the case of ADI with a subtype such as a Customer Account, the physical table name is based on the Subtype. Hence, one or more physical table names are displayed.

17. When you select the table, the respective attribute is displayed.

Figure 30: Physical View Window



18. In Physical View, you can search with either an attribute name or physical column name.
19. In both Logical and Physical views, you can click the filter.
20. A filter drawer is displayed with options to filter based on applications, OFSAA Module, Logical Domain, and other properties.
21. Select the required application and then click **Apply**. The Summary window displays the filtered ADIs.
22. Click **Reset** to deselect all filter options.
23. Click **Apply** to filter the attribute list.

6.3 Connectors











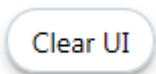


Connectors allow mapping one or more External Data Descriptors with an Application Data Interface, in the case of “Insert Connectors” or vice versa, in the case of “Extract Connectors”.

Factory-defined and maintained Connectors are available for specific Oracle applications – FLEXCUBE, Oracle Banking Platform, Data Relationship Management, and Accounting Hub Cloud Service. See the User Guides of these applications from the OFSAA Data Integration Application Pack at [OHC Documentation Library](#). You can configure Insert and Extract Connectors for data exchange with other applications.

6.3.1 Icons and Description

To create a connector, you must understand each of the icons at the beginning. While creating a Connector, the icons that are displayed are explained as follows:

Table 26: Icons and Description

| Icon | Description |
|---|--|
|  | Click this icon to view the list of all External Data Descriptors created in the setup. You can drag the desired EDD on the canvas. |
|  | Click this icon to view the list of all ADIs created in the setup. You can drag the desired ADI on the canvas. |
|  | Click this icon to open the Mapping window. You can map the source column to the target column in the window. |
|  | This component is used for defining a join between two entities. Click this icon to open the window where you can define the join condition between two entities. |
|  | This component is used for defining the filter of a given entity. Click this icon to open the window where you can define the filter condition. |
|  | This component is used for defining the lookup condition. Click this icon to open the window where you can define the join condition between two entities. |
|  | This component is used for defining the Derived column. Click this icon to open the window where you can define an expression, which can be mapped to the target column. |
|  | This component is used for transforming flattened hierarchy entities into parent-child hierarchy entities. |
|  | This component is used for Transpose (Rows to Columns) for a given entity. Click this icon to open the window where you can define the pivot data element and the new columns, which are transposed from multiple rows of source entity. |
|  | This component is used for Transpose (Columns to rows) for a given entity. Click this icon to open the window where you can define the unpivot data element and new rows which are transposed from columns of the source entity. |
|  | This component is used for defining a group by and having a clause for Aggregation. Click this icon to open the window where you can define a group by and having a clause for aggregation. |
|  | Click this button to remove all the nodes added to the canvas. |
|  | This is displayed on the connector window when the connector is published and is opened in view mode. The connector is not editable. |
|  | This is displayed on the connector window when the connector is not published. The connector is editable. |

6.3.2 Creating Connectors

The connector allows mapping one or more External Data Descriptor with Application Data Interface. It allows mapping of one or more ADI with EDD as well, in the case of extract type connector.

There are pre-built Connectors for Oracle applications such as FLEXCUBE, Oracle Banking Platform, and so on. For other applications, you need to define Connectors for your EDDs.

6.3.2.1 Loading Data into OFSAA

To create a Connector for loading data into OFSAA, follow these steps:



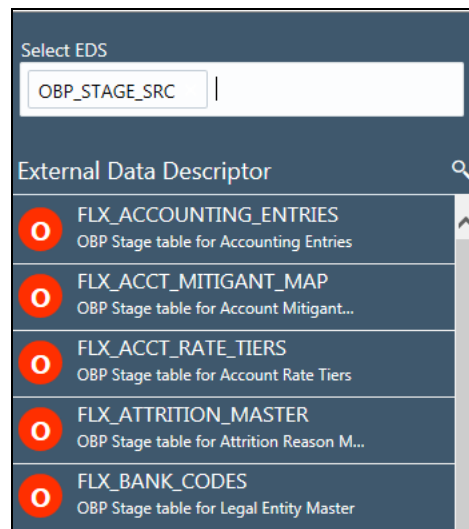
1. On the **Connector Summary** window, click  .
Or:
2. From the **Data Integration Hub Designer** window, click  in **Insert Connectors**, to move the data from an EDD to an ADI.
3. The **New Connectors Definition** window is displayed.
4. To define a connector, you must have a source with EDD and a target, which is ADI.
5. Click **Source** to select the required EDDs. Here, you can filter your selection based on the EDS selected. The EDD node's color depends on the source system type.

Figure 31: Connectors - Source



For example:

- File types are in blue.
 - Oracle types are in red.
 - HDFS types are in orange.
6. If you select 'OBP_STAGE_SRC' as the EDS, it displays the EDDs for that particular EDS selected.


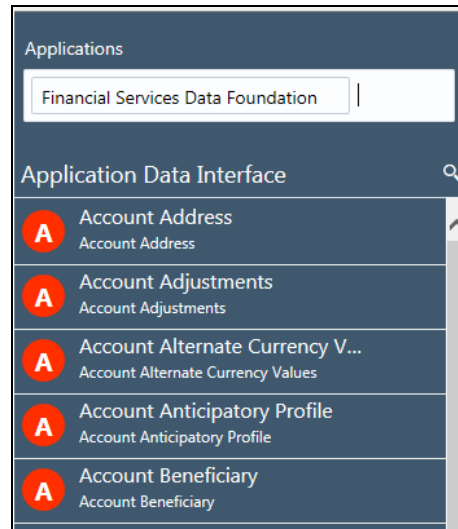
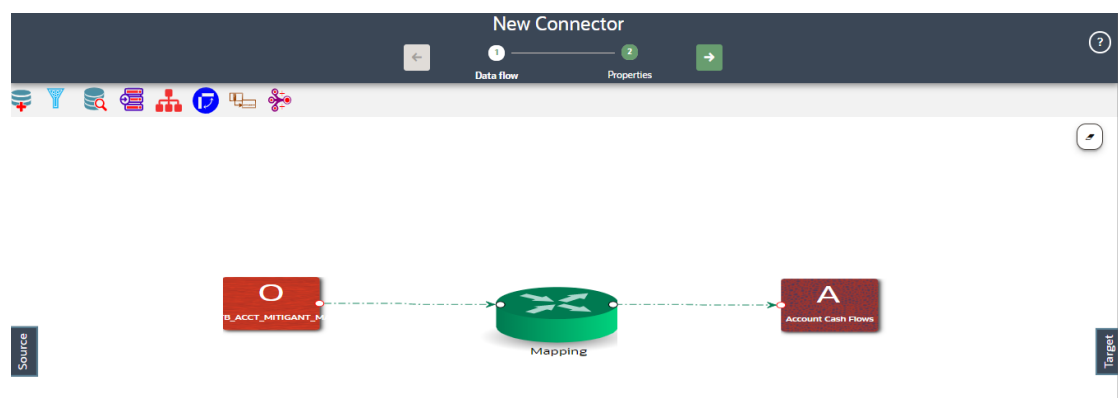
7. Click **Search**  to search for a particular EDD. You can select multiple EDS.
8. Select the required EDD and drag it to the canvas.
9. Click **Target**. Here you can filter ADIs based on the application selected.

Figure 32: Connectors - Target



10. Click **Search** to search for a particular ADI.
11. Select the required ADI. Drag it to the canvas and then link the input and output nodes.
12. Click the input white circle. The anchor symbol is displayed. Drag and drop the line to link it to the required component.

Figure 33: New Connector Window




13. At any given time, you can right-click the node to either delink or remove links/outline or delete a node.
14. To edit or view the properties, on the **Connector** window, click .
15. In **Connector Details**, enter the name and description for the connector.

Figure 34: Connector Details Window

The screenshot displays the 'Connector Details' window for a connector named 'testingConnectorTruncate'. The interface is divided into several sections:

- Connector Details:** Includes a 'Name' field with the value 'testingConnectorTruncate' and a 'Description' field.
- Pre Load Options:** Features a 'Truncate' dropdown menu set to 'Selected Rows' and a 'Filtered Expression' field containing the expression '[(Account Address)|(Account Or Contract Number) = '']'. There is an 'Edit' icon next to the expression field.
- Properties:**
 - Default Properties:** Includes 'Parallel' set to 'True', 'Avoid Partition Exchange' set to 'No', and 'Degree Of Parallel' set to '5'.
 - Table Properties:** Includes 'Do you want to use DBLink?' set to 'No'.
- Audit Trail:** Shows 'Created By: OFSAD', 'Created Date: 2018-11-21 22:39:12.0', 'Modified By: OFSAD', and 'Last Modified Date: 2018-11-26 20:14:46.0'.


16. In Pre Load Options, select the truncate option to be defined in the target. To remove data from the table as per the truncate option specified, select **Truncate**.

- Select **No**, if you do not wish to truncate the table before loading.
- If you select **Partial Truncate**, provide the Partition Name. The parameter name can be provided here. If you want to truncate a partition, the Partial Truncate option must be selected. Specify the partition to be truncated before load.

NOTE For multi-target loads, the truncate type must be the same for all targets. However, truncate expression may vary.

- Select **Full Truncate** to fully truncate. Here no expression is required. If you want to truncate the entire table, the Full Truncate option must be selected.
- Select **Selected Rows** to truncate on the selected expressions. If you remove specific rows, the selected rows option must be selected. Specify the filter condition for the rows to be deleted. Specific rows are removed from the table before load.

NOTE If OFSAA is hosted on a Big Data environment, then the Truncate with **Selected Rows** option is not supported.

- Click **Edit**  to filter the selected rows.
- Select the required entity and click **Validate**. This validates the expression.
- Click **Ok** once the expressions are selected.

In the image, truncate details are selected for Account Address.

17. In **Properties**, select the Default Properties, File Properties, and Table Properties if you have selected a default type or file type or a table type respectively.

NOTE See the [Connector Properties](#) section for more information on the properties.

6.3.2.2 Processing Data within OFSAA

To create a Connector for processing data within OFSAA, follow these steps:




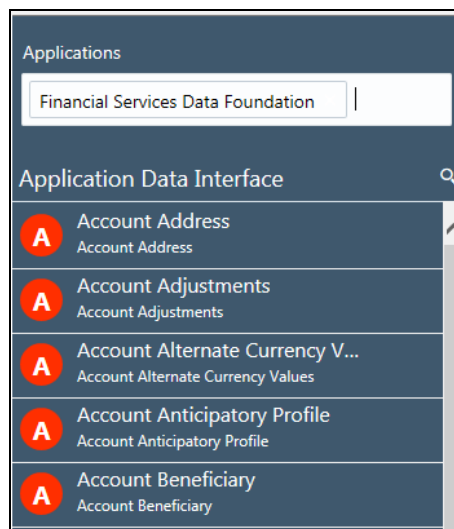
1. To create a new connector, on the **Connector Summary** window, click  .
- Or:
2. From the **Data Integration Hub Designer** window, click  from Process Connectors to move the data within an ADI.
3. The **New Connectors Definition** window is displayed.
4. To define a connector, you must have an ADI for source and another for Target.
5. Click **Source** to select the required ADIs.
6. Click Search  to search for a particular ADI. You can select multiple ADIs.
7. Select the required ADI and drag it to the canvas.
8. Click **Target**. Here you can filter ADIs based on the application selected.

Figure 35: Connectors - Target




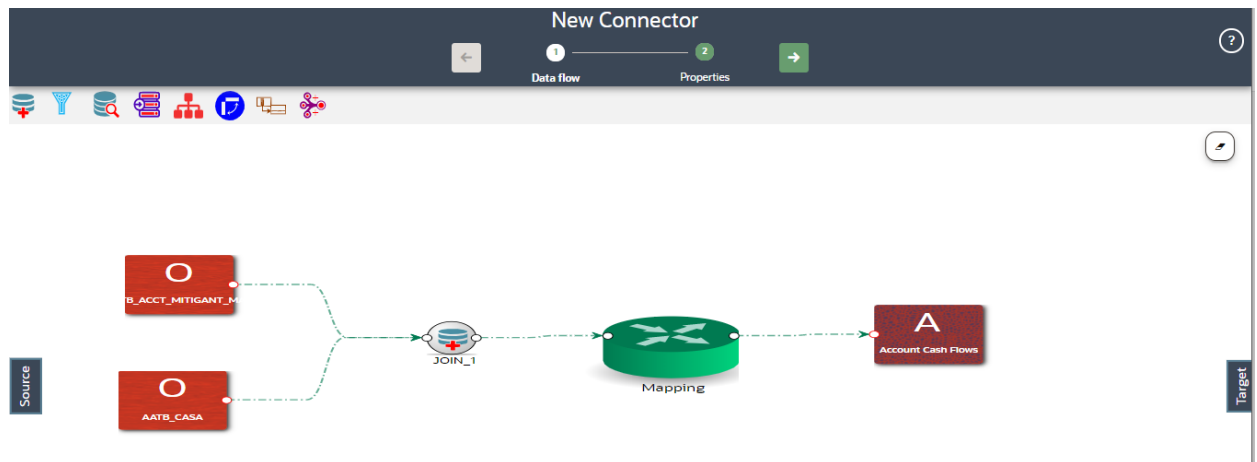
9. Click Search  to search for a particular ADI.
10. Select the required ADI. Drag it to the canvas and then link the input and output nodes.
11. Click the input white circle. The anchor symbol is displayed. Drag and drop the line to link it to the required component.

Figure 36: New Connector Window




12. At any given time, you can right-click the node to either delink or remove inlinks / outlink or delete a node.
13. To edit or view the properties, on the **Connector** window, click .
14. In **Connector Details**, enter the name and description for the connector.


Figure 37: Connector Details Window

15. In **Pre Load Options**, select the truncate option to be defined in the target. When you select Truncate, it removes data from the table as per the truncate option specified.
- Select **No**, if you do not wish to truncate the table before loading.
 - If you select **Partial Truncate**, provide the Partition Name. The parameter name can be provided here.
 - If you want to truncate a partition, the Partial Truncate option must be selected. Specify the partition to be truncated before the load.

NOTE For multi-target loads, the Truncate type must be the same for all targets. However, truncate expression may vary.

- Select **Full Truncate** to fully truncate. Here no expression is required. If you want to truncate the entire table, the Full Truncate option must be selected.
- Select **Selected Rows** to truncate on the selected expressions. If you want to remove specific rows, the selected rows option must be selected. Specify the filter condition for the rows to be deleted. Specific rows are removed from the table before load.

NOTE If OFSAA is hosted on a Big Data environment, then the Truncate with **Selected Rows** option is not supported.

- Click **Edit**  to filter the selected rows.
- Select the required entity and click **Validate**. This validates the expression.
- Click **Ok** once the expressions are selected.


In the image, truncate details are selected for Account Address.

16. In **Properties**, select the Default Properties, File Properties, and Table Properties if you have selected a default type or file type, or a table type respectively.

NOTE See the Connector Properties section for more information on the properties.

6.3.2.3 Extracting Data from OFSAA

To create a Connector for extracting data from OFSAA, follow these steps:

1. On the **Connector Summary** window, click  .
Or:
2. From the **Data Integration Hub Designer** window, click  from **Extract Connectors** to move the data from an ADI to an EDD.
3. The **New Connectors Definition** window is displayed.


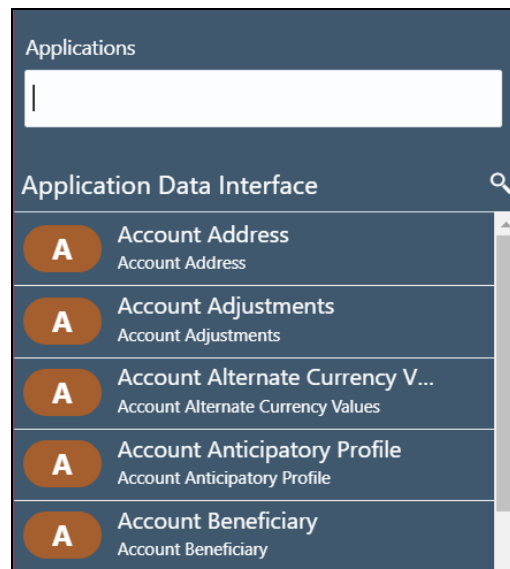
4. To define a connector, you must have a source with EDD and a target, which is ADI.
5. Click **Source** to select the required ADIs.
6. Here, you can filter your selection based on the ADI selected. The ADI node's color depends on the source system type.
7. Click Search  to search for a particular ADI.
8. Select the required ADI. Drag it to the canvas and then link the input and output nodes.

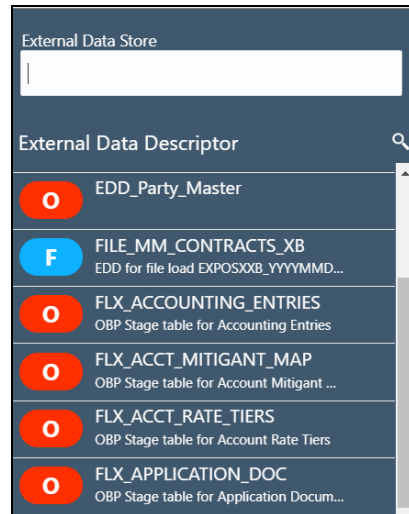
Figure 38: Connectors - Source



For example:

- The file types are in blue.
 - Oracle types are in red.
 - HIVE types are in brown.
9. Click **Target**. Here you can filter EDS based on the application selected.

Figure 39: Connectors - Target




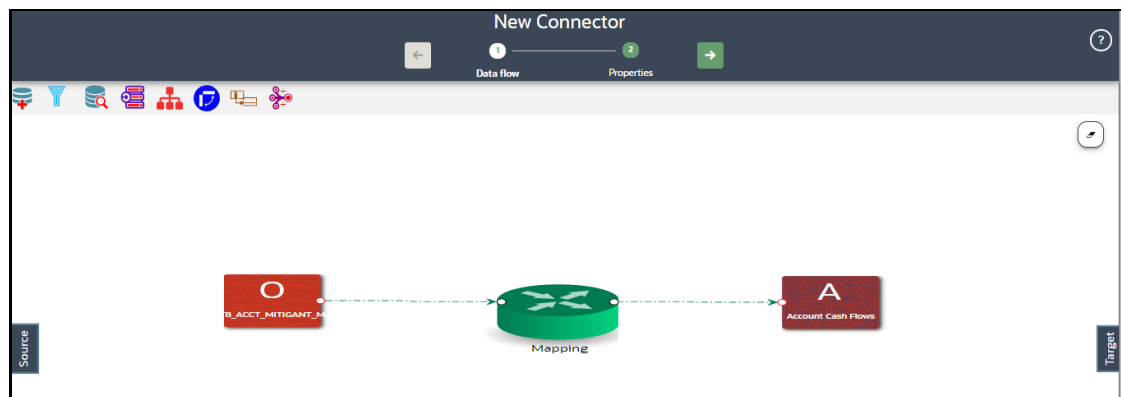
10. If you select 'OBP_STAGE_SRC' as the EDS, it displays the EDDs for that particular EDS selected.
11. Click Search  to search for a particular EDD. You can select multiple EDS.
12. Select the required EDD and drag it to the canvas.
13. Click the input white circle. The anchor symbol is displayed. Drag and drop the line to link it to the required component.

Figure 40: New Connector Window




14. At any given time, you can right-click the node to either delink or remove inlinks / outlink or delete a node.
15. To edit or view the properties, on the Connector window, click .
16. In **Connector Details**, enter the name and description for the connector.


Figure 41: Connector Details Window

The screenshot shows the configuration page for the 'Account Address Extraction' connector. The 'Pre Load Options' section has 'Address master' set to 'Truncate'. The 'Properties' section has 'Parallel' set to 'True', 'Avoid Partition Exchange' set to 'No', and 'Degree Of Parallel' set to '5'. The 'Table Properties' section has 'Do you want to use DBLink?' set to 'No'. The 'Audit Trail' section shows the connector was created and last modified by 'OFSAD' on '2018-12-05'.

17. In Pre Load Options, select the truncate option to be defined in the target. When you select Truncate it removes data from the table as per the truncate option specified.

- Select **No**, if you do not wish to truncate the table before loading.
- If you select **Partial Truncate**, provide the Partition Name. The parameter name can be provided here. If you want to truncate a partition, the Partial Truncate option must be selected. Specify the partition to be truncated before load.
- Select **Full Truncate** to fully truncate. Here no expression is required. If you want to truncate the entire table, the Full Truncate option must be selected.
- Select **Selected Rows** to truncate on the selected expressions. If you remove specific rows, the Selected Rows option must be selected. Specify the filter condition for the rows to be deleted. Specific rows are removed from the table before load.

NOTE If data is extracted into a Big Data environment, then the Truncate with Selected Rows option is not supported.

- Click **Edit**  to filter the selected rows.
- Select the required entity and click **Validate**. This validates the expression.
- Click **Ok** once the expressions are selected.

In this image, truncate details are selected for the Account Address.

18. In **Properties**, select the Default Properties, File Properties, and Table Properties in case you have selected a default type or file type or a table type respectively.

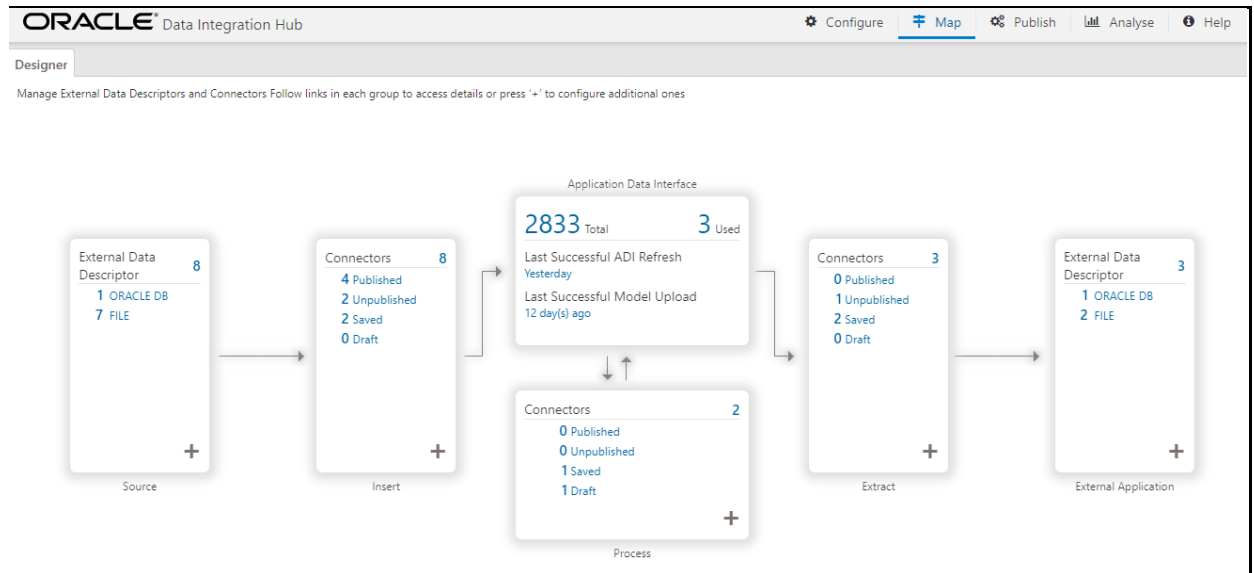
NOTE See the Connector Properties section for more information on the properties.

6.3.3 Managing Existing Connectors

To manage existing connectors, follow these steps:

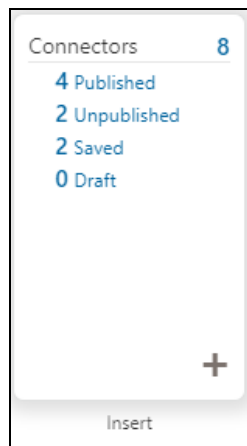
1. From the Data Integration Hub Designer window, to manage **Connector**, click links in each group to access details.

Figure 42: Connector Designer Window



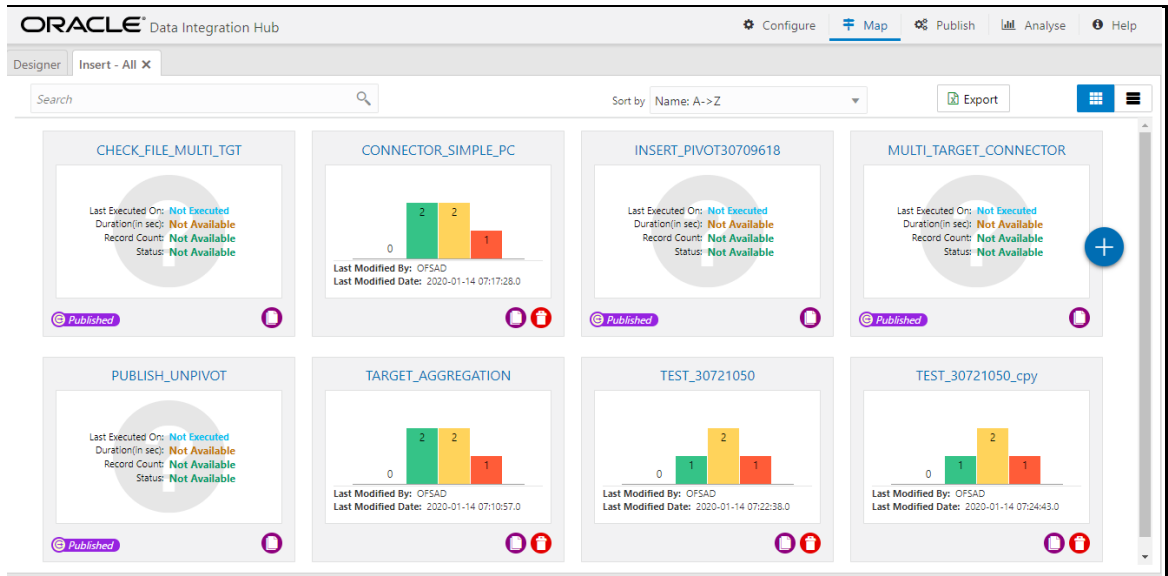
2. To manage **Insert Connectors**, click the links under Insert.

Figure 43: Insert Connectors



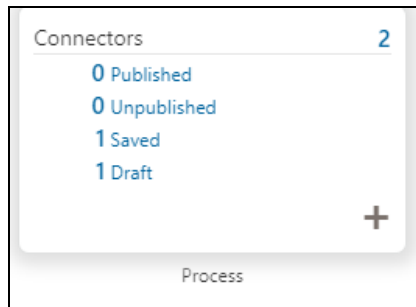
The **Insert - Connectors** window is displayed.

Figure 44: Insert Connectors Window



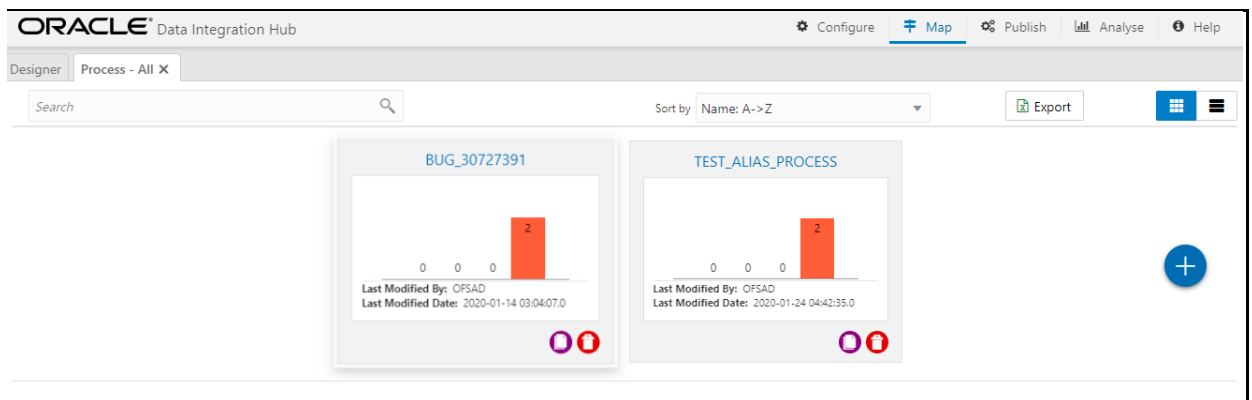
3. To manage **Process Connectors**, click the links under Process.

Figure 45: Process Connectors



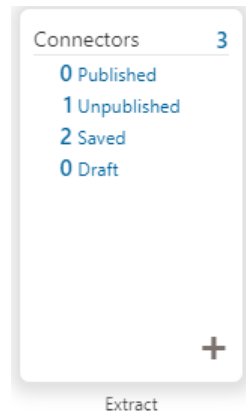
The **Process - Connectors** window is displayed.

Figure 46: Process Connectors Window



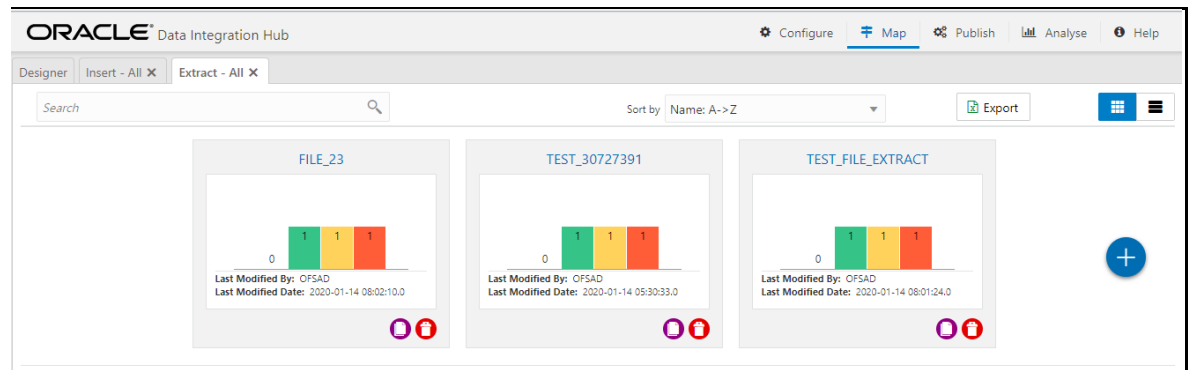
- To manage **Extract Connectors**, click the links under Extract.

Figure 47: Extract Connectors



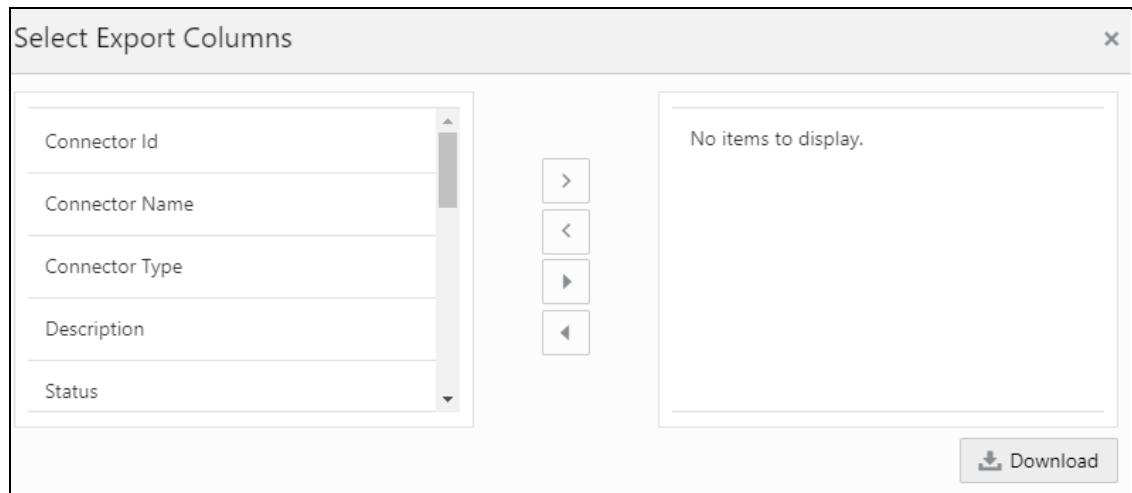
The **Extract - Connectors** window is displayed.


Figure 48: Extract Connectors Window

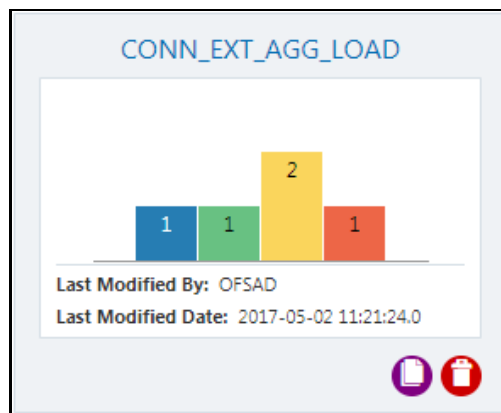


- The Connectors window lists all connectors that are defined in the setup. It displays the entire insert, process, and extract type connector details. It gives information about the number of parameters, EDS, EDDs, and ADI used in a specific connector.
- Click **Export**. The **Select Export Columns** window is displayed.

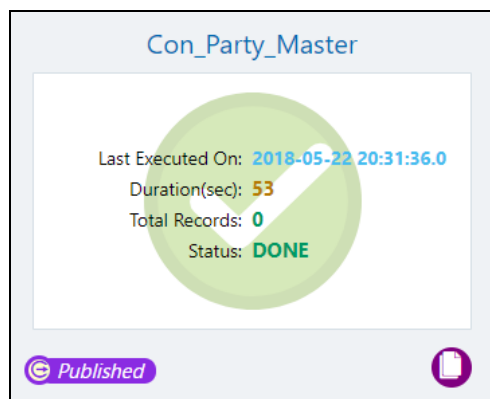
Figure 49: Select Export Columns



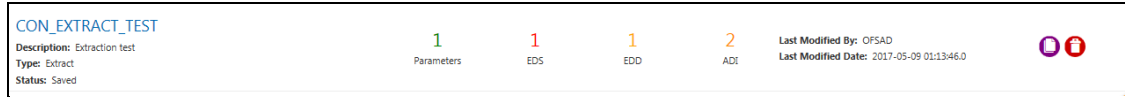
7. Select the required ID and click Download. The list of connectors is exported to an Excel sheet with Connector IDs and Connector Name. This lists both insert and extract type connector details.
8. Click  to view the connectors in the card view. It gives information about the number of parameters, EDS, EDDs, and ADI used in a specific connector.



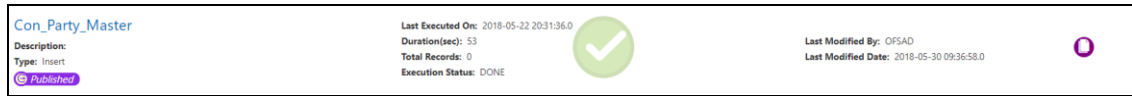
In case, the connector is published:



9. Click the Navigation icon . The **Connectors** are displayed in the list view.




In case, the connector is published:



6.3.4 Understanding the Components

6.3.4.1 Using Filter

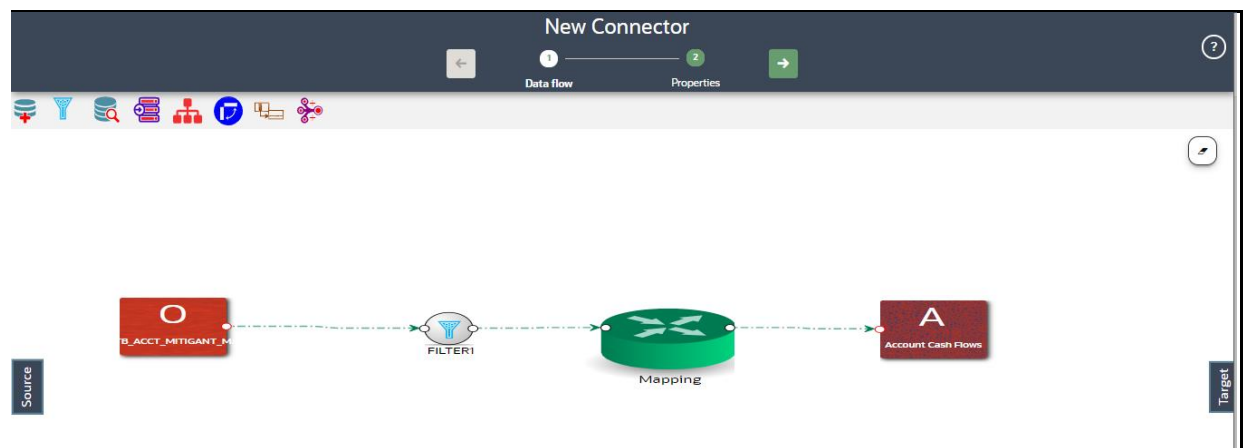
To use the filter component, follow these steps:

1. Drag and drop the **Filter**  component on the canvas to define a filter on an entity. For example, EDD (insert connector) / ADI (Process and Extract Connector).

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

2. It accepts input only from an entity and it can have only one output.
3. If you have multiple entities selected, and you want to have a filter for more than one entity, then you must select as many numbers of filters, connect to the respective entity, and then define their expressions.
4. For example, to add a filter to three entities, drag three filters.

Figure 50: Filter New Connector Window



5. At any given time, right-click the filter component to either delink or remove inlinks / outlinks or delete the filter component.
6. Double-click filter component. The **Filter Expression** window is displayed.
7. The selected entities and parameters are displayed in the **Filter Expression** window.

8. Specify the required filter expression using columns and parameters.
9. Click **Validate** to verify the correctness of the SQL expression.
10. Click **OK**.

NOTE

You do not need to add the 'WHERE' clause for the filter.

1. For File data loading, use the filter expression of the Number type along with single quotes. For example: `N_DRAWN_AMOUNT = '40000'`.
2. For the Date field, see `To_CHAR` function for comparison.
3. Parameters can also be used in the filter expression. The date format must be a valid SQL date format.

For **Example**:

```
[EDD_GL_DATA].[EXTRACTION_DATE] =
TO_DATE(#DIHDEV.MIS_DATE,'dd-MM-yyyy')
```

11. If the Source type is Hive, the filter expressions must conform to the following restrictions:

- Must be valid HiveQL
- Does not include Oracle built-in or user-defined functions
- Does not include Subqueries
- Includes Hive built-in functions only
- Parameters can also be used in the filter expression. `MISDATE` can also be passed dynamically so that it is loaded from the Batch Execution window. The date format specified must be valid Hive Date format i.e. `yyyy-MM-DD`
- **For Example:**
- Filter Expression in Connector: `-[EDD_GL_DATA].[EXTRACTION_DATE] = #DIHDEV.MIS_DATE`
- `MISDATE` Parameter in Batch:
- `MISDATE=$MISDATE:yyyy-MM-dd`

6.3.4.2 Using Join

To use the join component, follow these steps:

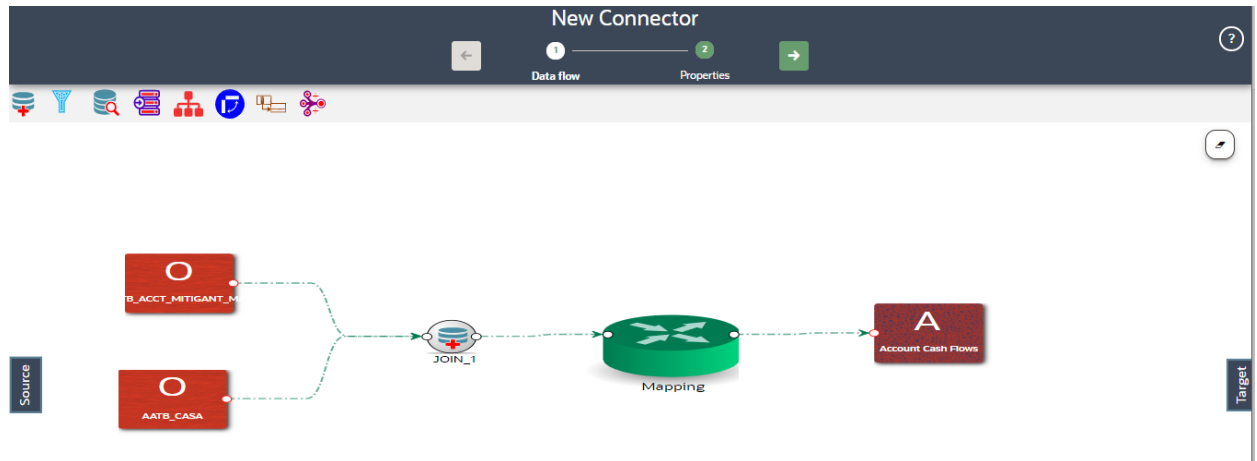


1. Drag and drop the **Join** component on the connector window to link multiple entities. For example, EDDs (insert connector) / ADIs (Process and Extract Connector).

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

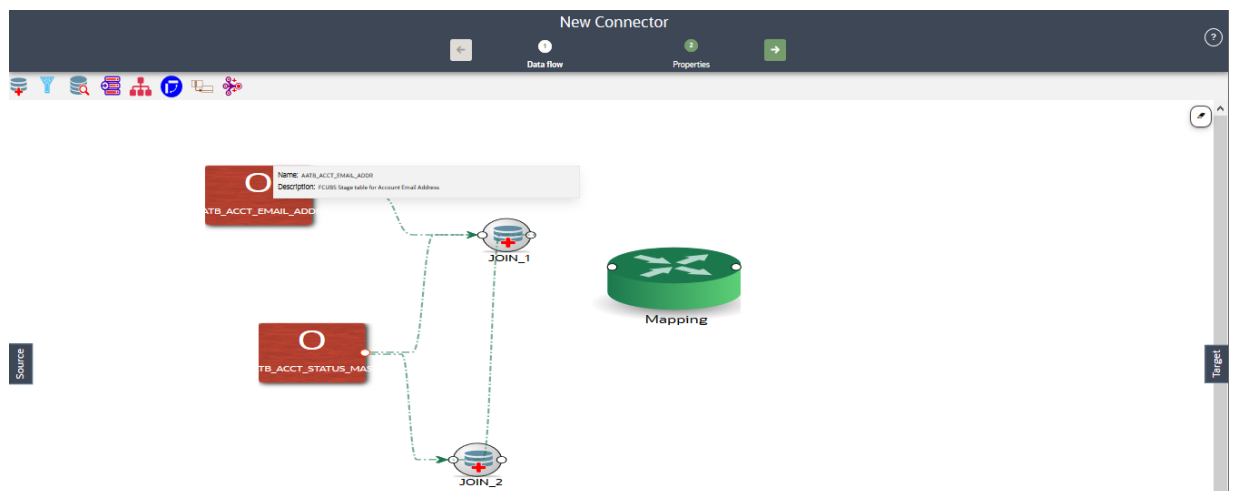
2. The Join component accepts input from two entities.

Figure 51: Join New Connector Window



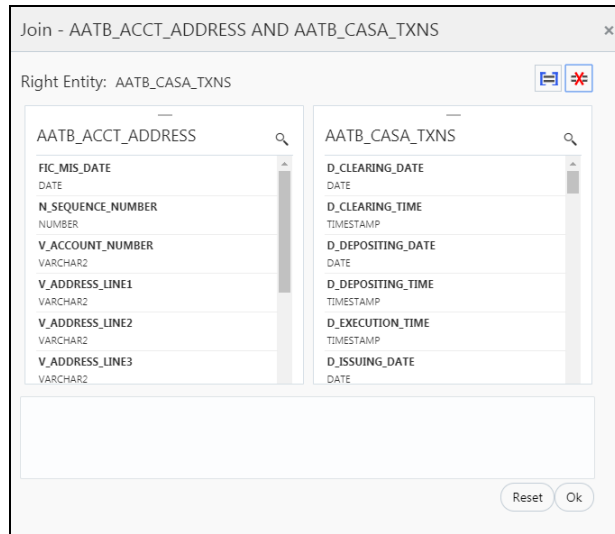
- To join more than two entities, drag another join component. Link the output of the first join to the input of the second join and then connect the other entities. You can repeat this for multiple entities. Select the Source Entity and click **Ok**.


Figure 52: Multiple Join Entities




- At any given time, right-click the join component to either delink or remove inlinks / outlinks or delete a join component.
- Double-click the join component to define a join condition. The Join window is displayed:
- Here you see the selected entities in the left and right tab.

Figure 53: Join Window



7. You can drag and reorder the left and right tab to choose the right/left entity in a join condition.
8. To join entities, the select column from the left and right tab and click **Add Join** . This displays the joined entities. You can join multiple entities.


```
[AATB_ACCT_ADDRESS].[V_ACCOUNT_NUMBER]=[AATB_CASA].[D_ACCT_OPEN_DATE] AND
[AATB_ACCT_ADDRESS].[V_ADDRESS_LINE2]=[AATB_CASA].[D_ACCT_CLOSED_DATE] AND
[AATB_ACCT_ADDRESS].[V_ACCOUNT_NUMBER]=[AATB_CASA].[D_BILLING_CYCLE_DATE]
```

9. To remove two joined conditions, select two columns from the left and right tab, and click **Remove Join** . The joined condition is removed from the list.
10. Click **Reset** to reset all the joined conditions.
11. Click **Ok**.

NOTE This creates an inner join between the connected EDDs.

6.3.4.3 Using Lookup

To use the lookup component, follow these steps:

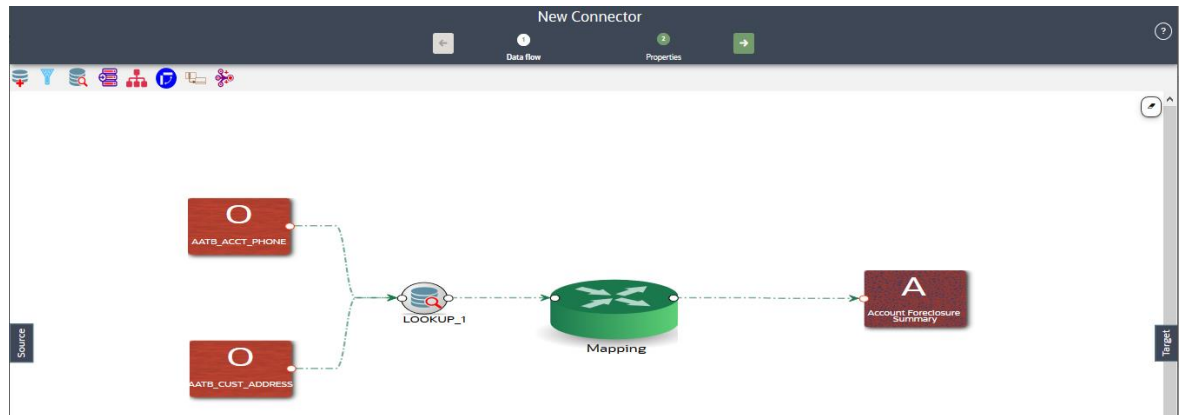
1. Drag and drop the **Lookup**  component on the canvas to define a filter on an entity. For example, EDD (insert connector) / ADI (Process and Extract Connector).

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

2. You can lookup values from an entity using this component.

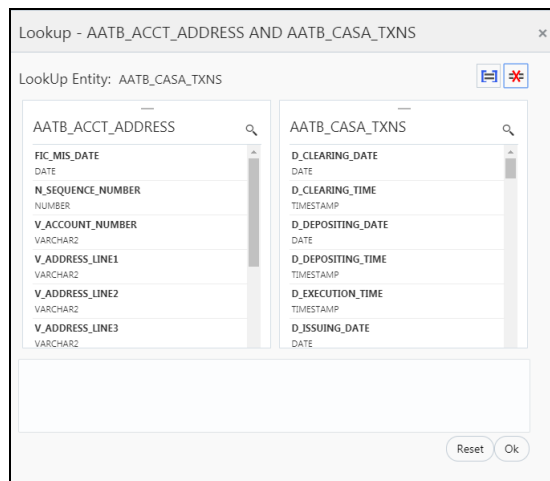
NOTE The lookup component accepts input from two entities. One from Value Entity and the other one from the Lookup Entity.

Figure 54: Lookup New Connector Window



3. At any given time, right-click the lookup component to either delink or remove inlinks / outlinks or delete a lookup component.
4. Double-click the lookup component to define a lookup condition. The Lookup window is displayed:
5. Here you see the connected entities in the left and right tab.

Figure 55: Lookup Window




6. The entity that is on the right side of the window is the lookup entity. You can change the lookup entity by moving it to the right side. The "LookUp Entity" field displays the entity specified for lookup.
7. To specify lookup condition, select data elements from left and right entities and click Add Join



. Lookup condition is displayed as follows:


```
[AATB_ACCT_ADDRESS].[V_ACCOUNT_NUMBER]=[AATB_CASA_TXNS].[D_CLEARING_TIME]
```

8. To remove a lookup condition, select data elements from left and right entities and click Remove Join . The lookup condition is removed from the list.
9. Click **Reset** to reset the lookup condition.
10. Click **Ok**.

NOTE This creates a left outer join between the connected entities.

6.3.4.4 Using Aggregation for an EDD

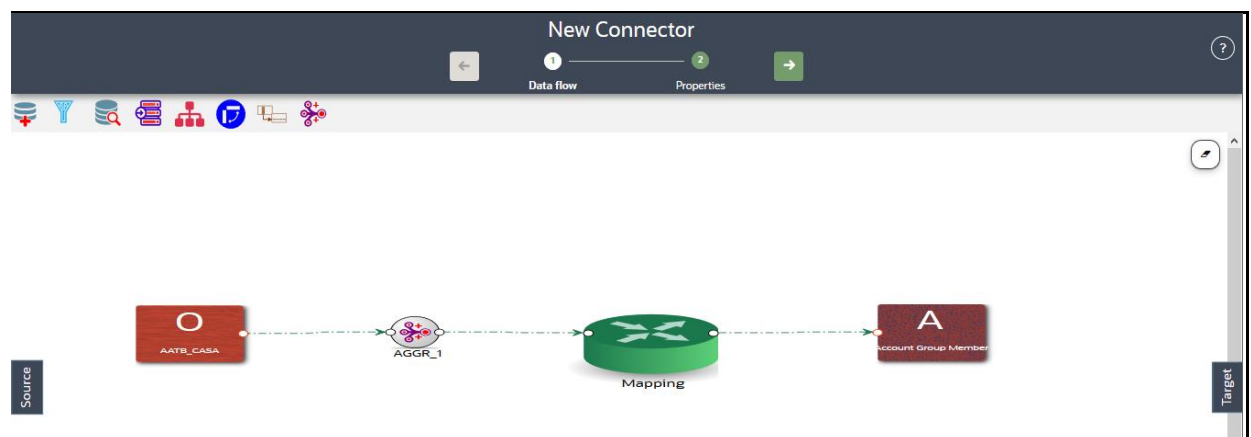
To use the aggregation component, follow these steps:

1. Drag and drop the **Aggregation**  component on the canvas to define an aggregation on an EDD.

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

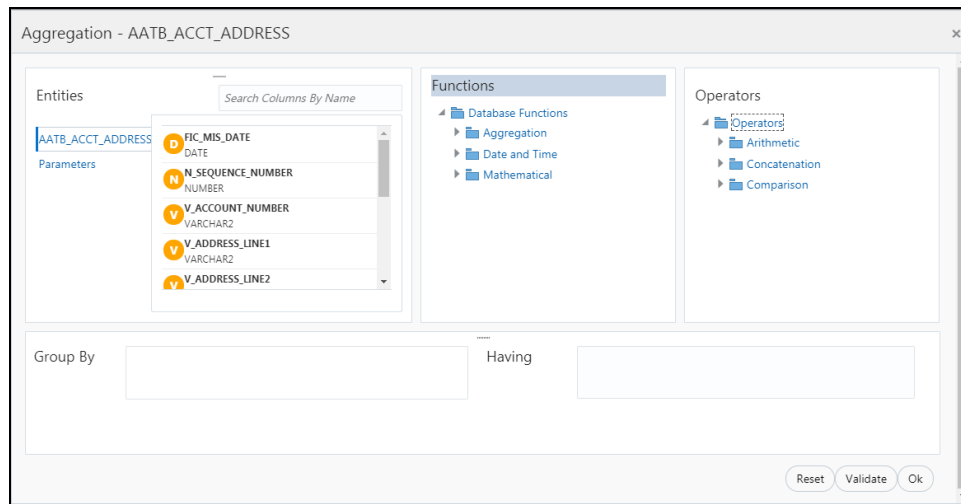
2. It accepts input only from an EDD and it can have only one output.
3. If you have multiple EDDs to be aggregated then you must select as many numbers of aggregation components, connect to the respective EDD, and then define their group by and having clauses.
4. For example, to add aggregation to three EDDs, drag three aggregation components.

Figure 56: Aggregation for an EDD New Connector Window



5. At any given time, right-click the aggregation component to either delink or remove inlinks / outlinks or delete the aggregation component.
6. Double-click the aggregation component to define an aggregation condition. The Aggregation window is displayed:
7. Here you see the selected EDD under the entity tab.

Figure 57: Aggregation Window



8. Select the group by columns and specify an expression for the having clause.
9. Click **Reset** to reset all the aggregation conditions.
10. Click **Validate** to verify the correctness of the SQL expression.
11. Click **Ok**.

6.3.4.5 Using Aggregation for Entire Dataset

To use the aggregation component, follow these steps:

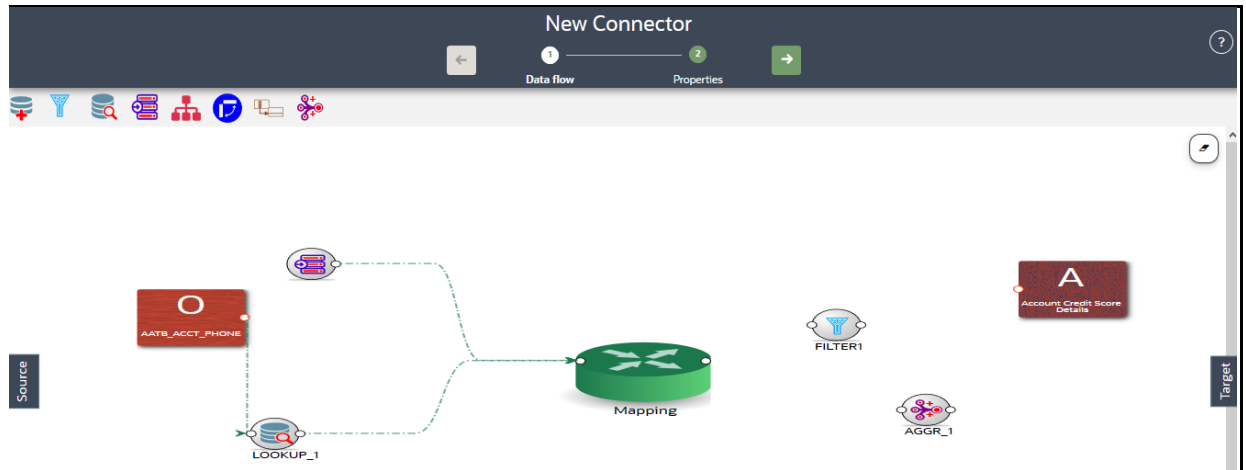


1. Drag and drop the **Aggregation** component on the canvas to define an aggregation on the entire dataset.

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

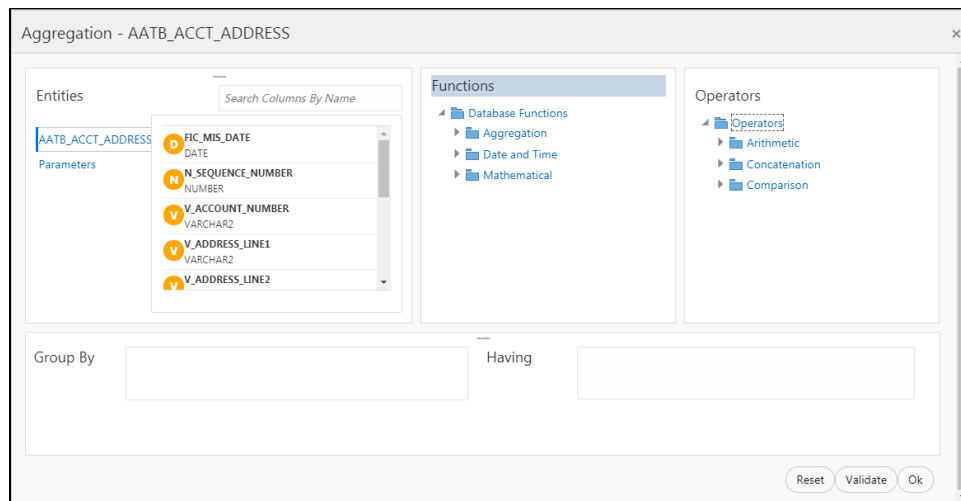
2. It accepts input only from the mapping component or filter that is connected to the mapping component.

Figure 58: Aggregation for Entire Dataset New Connector Window



3. At any given time, right-click the aggregation component to either delink or remove inlinks / outlinks or delete the aggregation component.
4. Double-click the aggregation component to define an aggregation condition. The Aggregation window is displayed:
5. Here you see the selected EDD under the entity tab.


Figure 59: Aggregation Window



6. Select the group by columns and specify an expression for the having clause.
7. Click **Reset** to reset all the aggregation conditions.
8. Click **Validate** to verify the correctness of the SQL expression.
9. Click **Ok**.

6.3.4.6 Using Transpose (Rows to Columns) for an EDD

To use the Transpose (Rows to Columns) component, follow these steps:

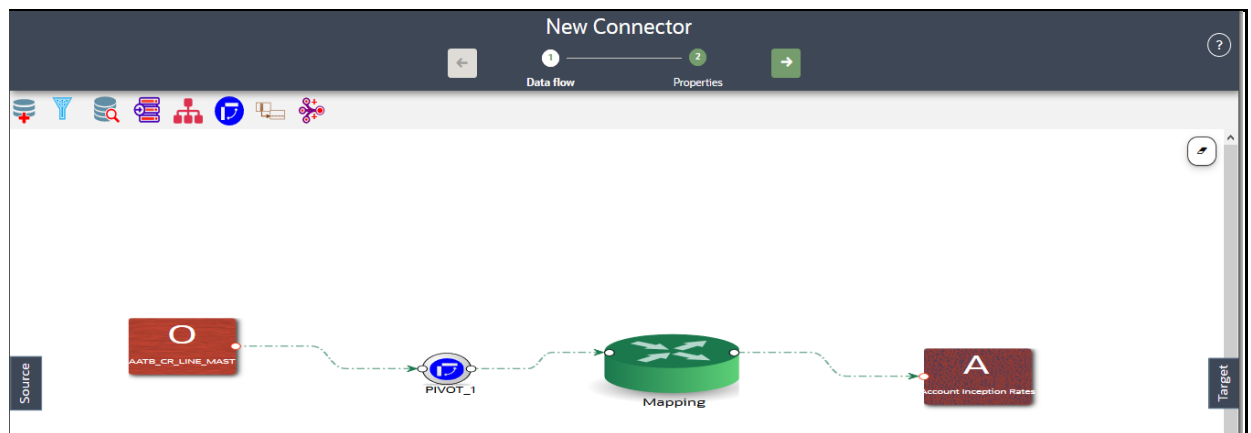
1. Drag and drop the **Transpose (Rows to Columns)**  component on the canvas to define a Transpose (Rows to Columns) component on an EDD.

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

2. It accepts input only from an EDD and it can have only one output.
3. If you have multiple EDDs selected, and you want to have a Transpose (Rows to Columns) component for more than one EDD, then you must select as many numbers of Transpose (Rows to Columns) components, connect to the respective EDD, and then define their expressions.

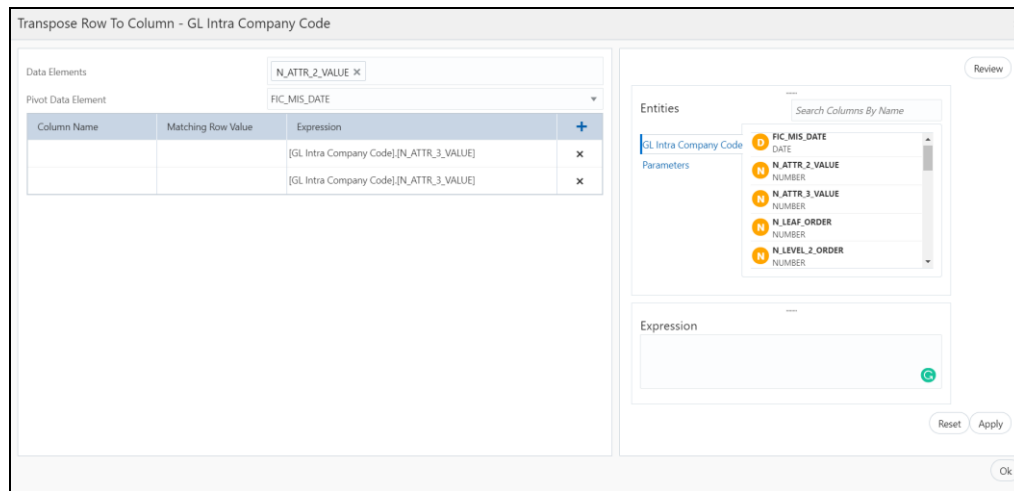
NOTE The output can be connected to Join, Lookup, and Mapping.
For example, to add the Transpose (Rows to Columns) component to three EDDs, drag three Transpose (Rows to Columns) components.

Figure 60: Transpose (Rows to Columns) for an EDD New Connector Window



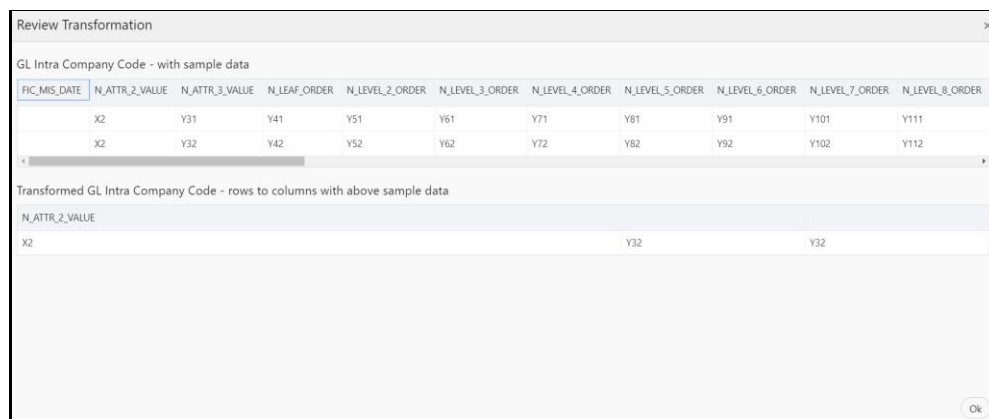
4. At any given time, right-click the Transpose (Rows to Columns) component to either delink or remove inlinks / outlinks or delete a Transpose (Rows to Columns) component.
5. Double-click the component to transpose the entity rows into columns. The Transpose Row to Column window is displayed.
6. Here you see the selected EDD and parameters.

Figure 61: Transpose (Rows to Columns) Window



7. Specify the pivot data element to transpose rows into columns.
8. Specify the Row Value – Transposed Column – Expression combination. You must have a minimum of two combinations.
9. Click Review to review the transformation. The Review Transformation window displays the sample of the transformation data.


Figure 62: Review Transformation Window



10. Click **Ok**.

6.3.4.7 Using Transpose (Columns to Rows) for an EDD

To use the Transpose (Columns to Rows) component, follow these steps:

1. Drag and drop the **Transpose (Columns to Rows)**  component on the connector window to define a Transpose (Columns to Rows) component on an EDD.

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

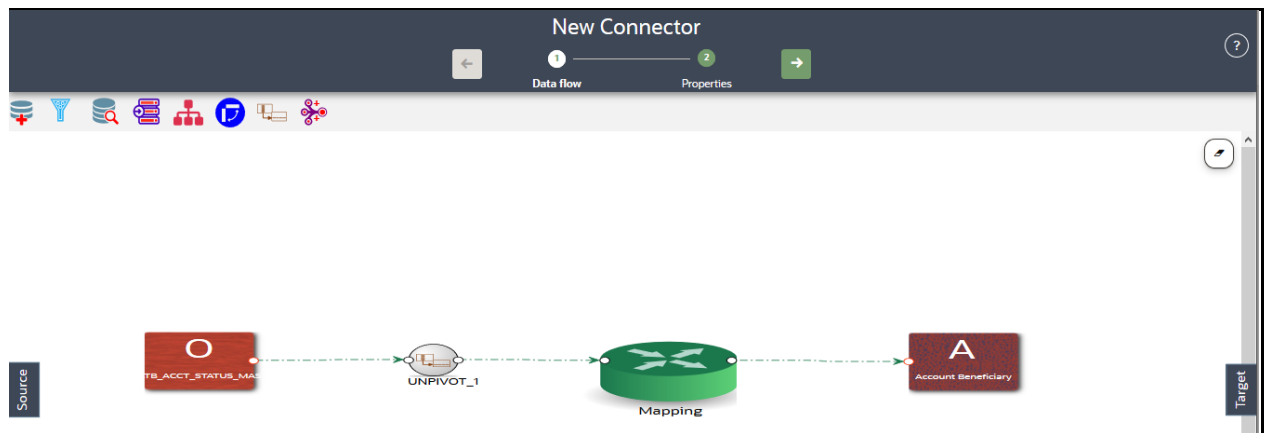
2. It accepts input only from an EDD and it can have only one output.

3. If you have multiple EDDs selected, and you want to have Transpose (Columns to Rows) component for more than one EDD, then you must select as many numbers of Transpose (Columns to Rows) components, connect to the respective EDD, and then define their expressions.

NOTE The output can be connected to Join, Lookup, and Mapping.

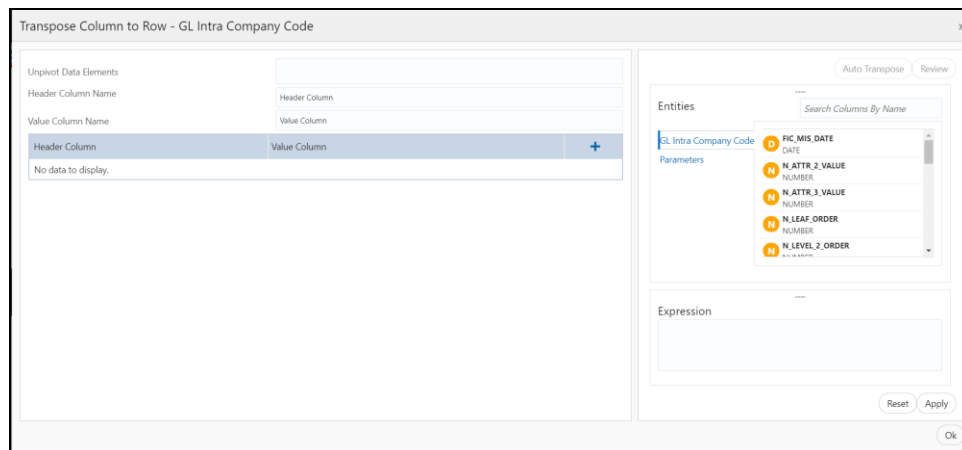
4. For example, to add the Transpose (Columns to Rows) component to three EDDs, drag three Transpose (Columns to Rows) components.

Figure 63: Transpose (Columns to Rows) for an EDD New Connector Window



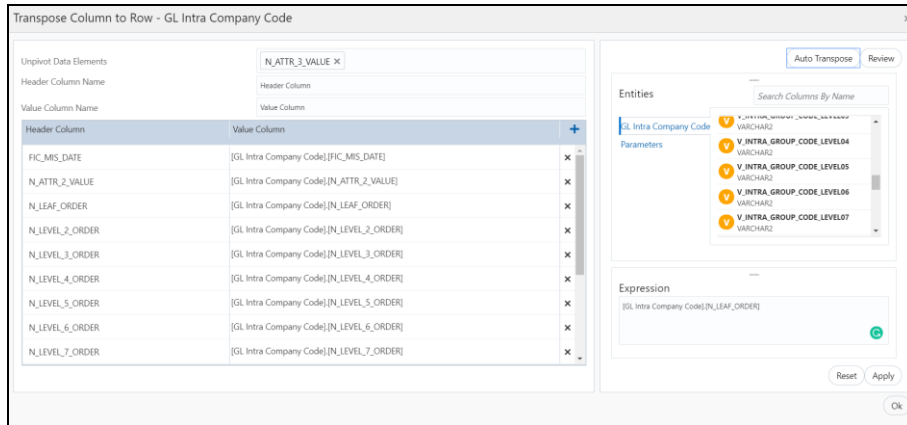
5. At any given time, right-click the Transpose (Columns to Rows) component to either delink or remove inlinks / outlinks or delete a Transpose (Columns to Rows) component.
6. Double-click the component to transpose the entity columns into rows. The Transpose Column to Row window is displayed.
7. Here you see the selected EDD and its parameters.

Figure 64: Transpose (Columns to Rows) Window



8. Specify the Unpivot Data Element to transpose columns into rows.
9. Specify the Header Column Name and Value Column Name.
10. Specify the Column Value (Header column) and Expression Pair (Value column) for each transposed row. You must have a minimum of two pairs.
11. After specifying the Unpivot Data Elements, click **Auto Transpose**. This will transpose columns into rows based on the unpivot data elements selected.

Figure 65: Transpose (Columns to Rows) Window




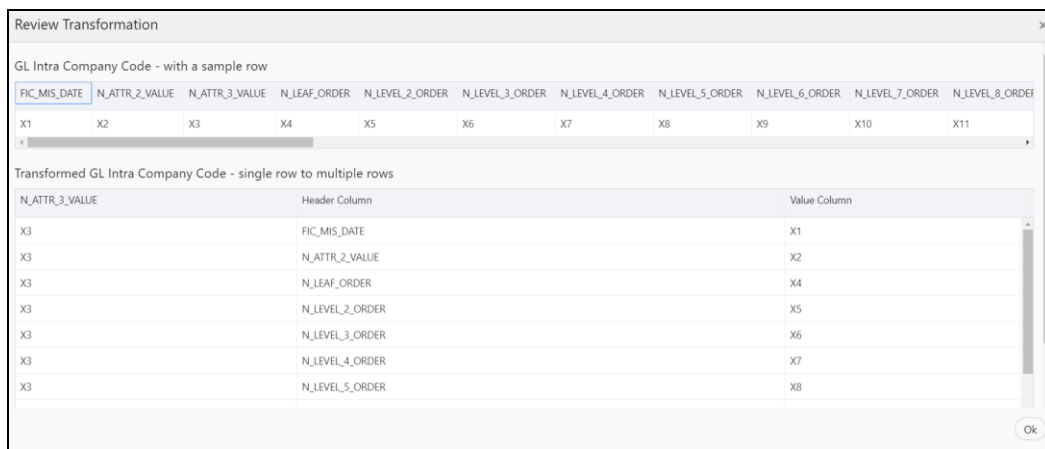
12. You can also click  drag and drop the columns.
13. Click **Review** to review the transformation. The Review Transformation window displays the sample of the transformation data.

Figure 66: Review Transformation Window



14. Click **Ok**.

6.3.4.8 Using Derived Column

To use the Derived Column component, follow these steps:

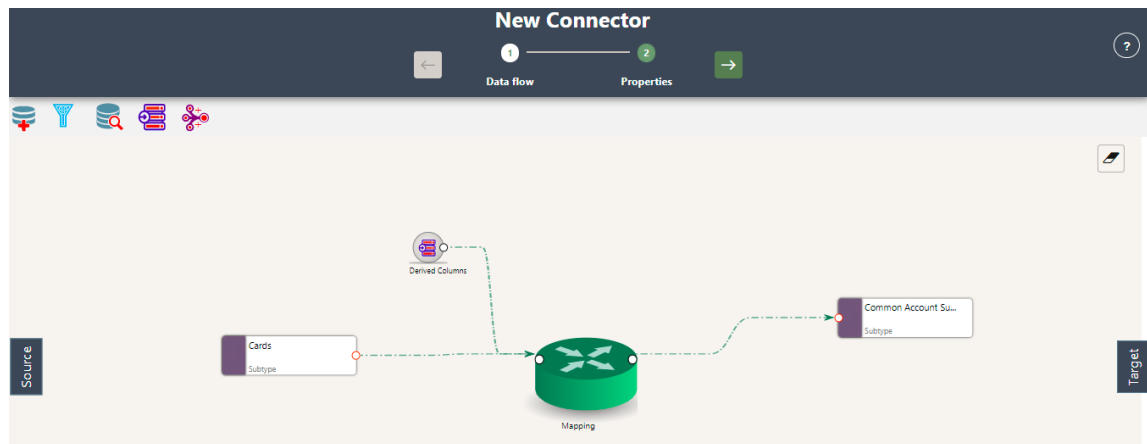
1. Drag and drop **Derived Column**  component on the canvas.

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

2. Connect the Derived Column to the mapping.

NOTE The output must be connected to the mapping.

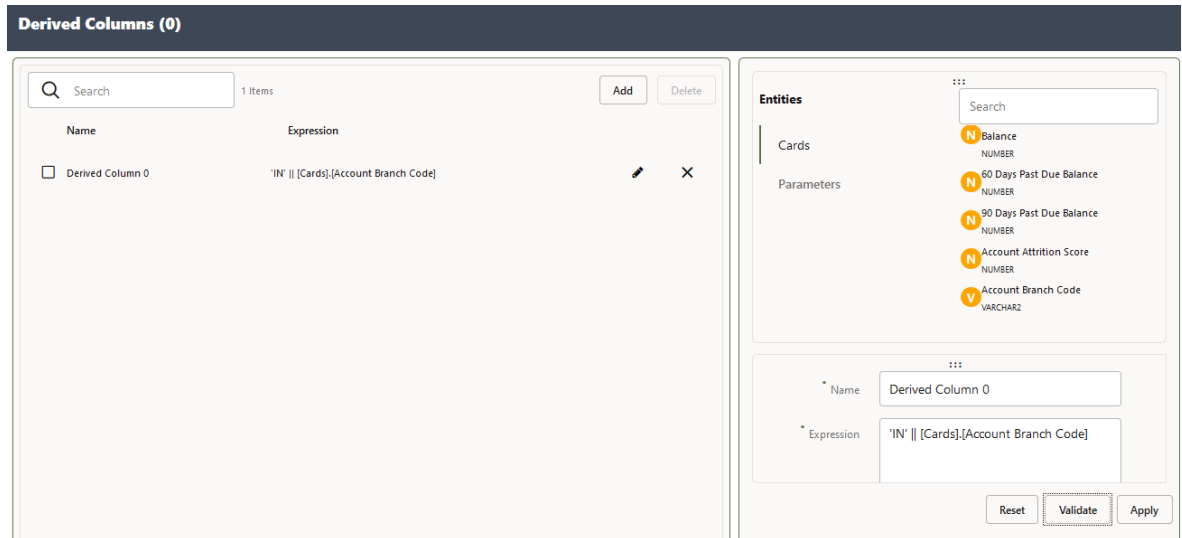
Figure 67: Derived New Connector Window



3. At any given time, right-click the expression component to either delink or remove outlinks or delete an expression component.
4. Double-click the Derived Column component.

The Derived Expression window for Derived Column is displayed.

Figure 68: Derived Column Window



5. Click **Add** to define a new Derived Column.
6. Click **Edit** to specify the name and expression of the Derived Column. You see the selected EDDs in the right tab.
7. Click **Validate** in case you wish to verify the correctness of the SQL expression.
8. Click **Apply**.
9. Repeat the steps for as many Derived Columns.
10. Click **Ok**.

6.3.4.9 Using Mapping

1. Double-click **Mapping**. The Mapping window is displayed.

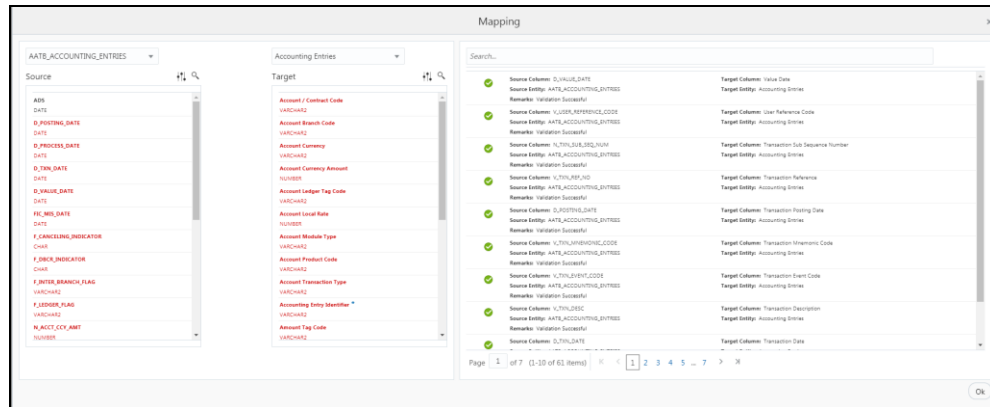
Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.





NOTE

The input and output for the Mapping component must be connected before specifying the mappings.

2. The mapping window displays the EDDs and ADIs and their respective data / derived data elements.

Figure 69: Mapping Window




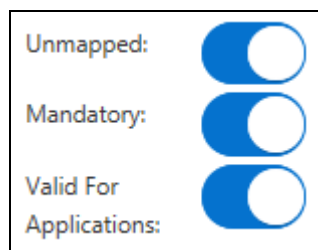
- Click a Data Element under Source, Attribute under Target, and then click **Map** . On the RHS, the column mapping is displayed.
- The following validations are done for the mapping:
 - Data Type Validation
 - Data Length Validation
 - Data Precision Validation
- If validation is successful, it displays Successful  next to the mapping.
- If any of the validations fail, it displays Warning  next to the mapping.
- At any given time, you can select **Unmap** to unmap the source and target.
- Click **Auto-Map**  to auto map a source and target.

NOTE Auto-mapping is done by matching the logical/physical column name of both the source or target.


- In the Source column, click **Filter** . Enable it to view the unmapped items.


NOTE The mapped columns are displayed in red.


- In the Target column, click **Filter** . Enable it to view the unmapped, mandatory, and valid for applications.





11. Under the Target column, you can hover over each item to see the details. It provides the description, length, and scale information.

12. Click **Search**  to search for a column name under the Source or Target column list.

13. Click **Delete**  to delete all the mappings. You can also delete individual mappings by selecting the cross symbol next to the column mapping.

14. Click **Import Mapping**  to import a mapping Excel. Choose mapping Excel from the file browser.

15. Click **Export Mapping**  to export the mapping information. This downloads an Excel file.

16. Click **Search**  to search for a column mapping. You can search for an item based on the source column name, target column name, source or target entity, or a remark.

6.3.4.10 Using the Flatten Table to PC Hierarchy Transformation for an EDD

To use the Flatten table to PC Hierarchy Transformation, follow these steps:

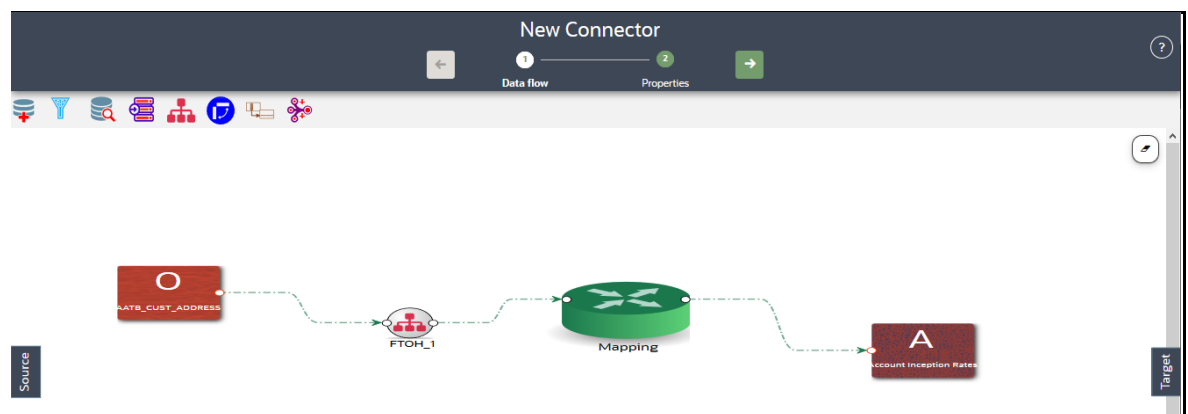
1. Drag and drop the **PC Hierarchy**  component on the canvas.

Alternatively, you can navigate to the component using the Tab key and use key board shortcut Ctrl C (Copy) and Ctrl V (Paste) to copy and paste the component on the canvas. To position the component, select the component and use arrow keys on the keyboard to reorder.

2. Connect the Derived Column to the mapping.

NOTE The output must be connected to the mapping.

Figure 70: PC Hierarchy New Connector Window



3. At any given time, right-click the expression component to either delink or remove outlinks or delete an expression component.
4. To define the expression, double-click the Flatten table to the PC hierarchy. The Flattened Table to Hierarchy window is displayed.

Figure 71: Flattened Table to Hierarchy Window

5. Choose the Hierarchy Type. The types of hierarchy supported are Balanced, Ragged, and Skipped. Click to view the details and understand how the hierarchies are defined.

Figure 72: Flattened Table to Hierarchy Window

6. Specify the Number of levels in the hierarchy. This field accepts only numbers.
7. Specify the Parent Node Column name and Child Node Column name which are used in the mapping.
8. Select the **Key Elements** from the drop-down list.
9. Select all nodes. You can change the date and or other details from the drop-down list.

10. Click **Review** to view the transformation changes.
11. Click **Ok**.

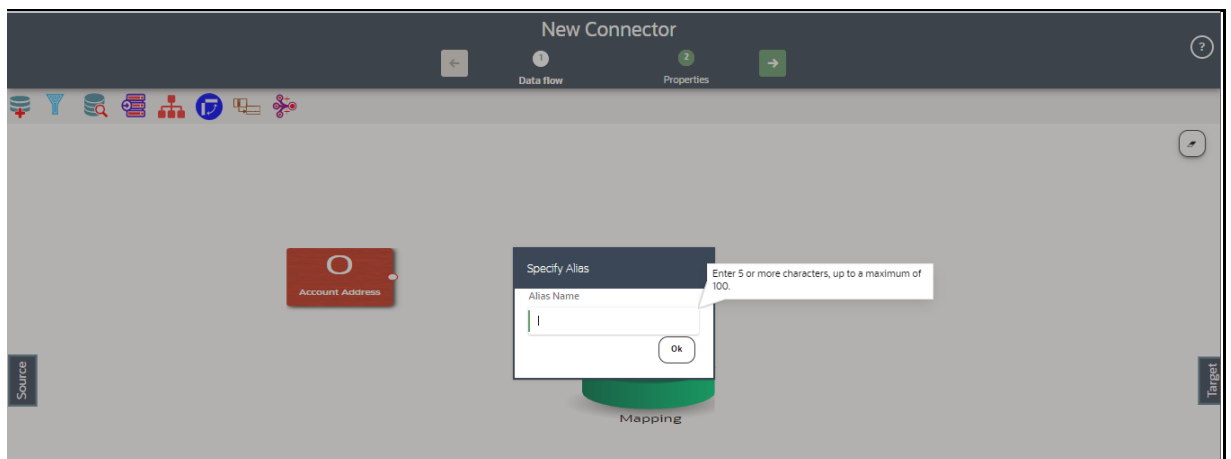
6.3.5 Specifying Alias in Connector

Alias refers to an assumed name or pseudonym assigned to an EDD or an ADI as you define Connectors, much like table aliases used in SQL statements. Aliases allow an EDD or an ADI to be referred in multiple joins, lookups, or both, within the same Connector definition, each in a distinct context.

Aliases are automatically initiated by DIH as and when required while defining Connectors. The following scenario explains the mechanism:

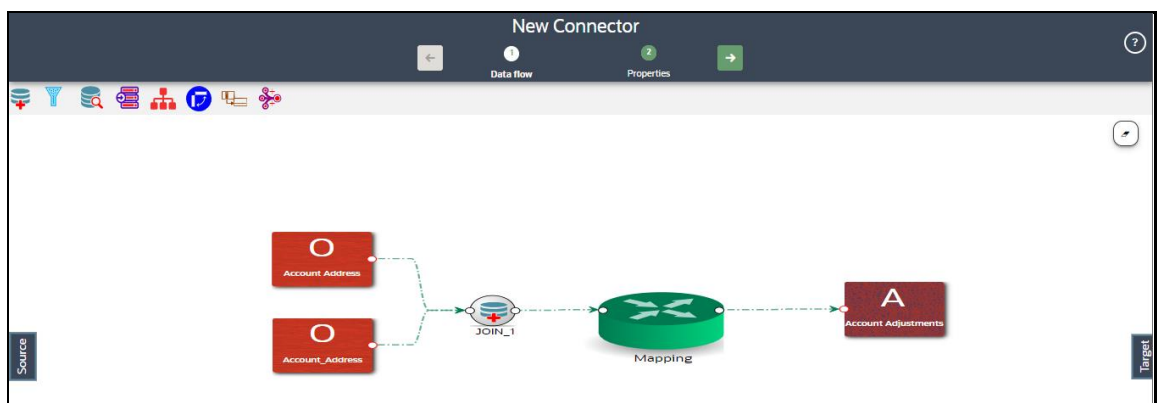
1. Drag and drop an ADI or EDD more than once, for use in separate joins or looks-up, on the New Connector canvas. DIH initiates the specification of an alias by displaying the **Specify Alias** dialog box.

Figure 73: Specify Alias



2. Enter a name of your choice under Alias Name and click Ok. Note that the Alias Names must be unique within a Connector.

Figure 74: New Connector Window



6.3.6 Connector Properties

The properties by default have some values. For every connector, it needs to be reviewed. Generally, no change is required.

The following table describes the fields in the connector properties.

Table 27: Connector Properties

| Fields | Applicable Values | Description | Default Value |
|--------------------|--|---|---------------|
| Loading mechanism | <ul style="list-style-type: none"> • External Table • SQLLDR | <p>There are two options External Table and SQLLDR.</p> <ul style="list-style-type: none"> • External Table - If the loading mechanism is selected as an External table, then the file-landing zone must be located/mounted on the database server. • SQLLDR - This option is only applicable when OFSAA is hosted in Oracle Database. The file-landing zone must be located or mounted on the server where the ODI agent is running. Oracle Database Client must be installed in the server where the ODI agent is running. <p>NOTE: If the loading mechanism is selected as External Table, the file must be located in the same place as the database server.</p> <p>If the target database type is HDFS, only the External Table option is enabled.</p> <p>If the target database type is Oracle, provide CREATE DIRECTORY role to the Atomic schema. Also, the path/folder used in the directory must have read and write permissions.</p> | SQLLDR |
| DIRECT | <ul style="list-style-type: none"> • True • False | Direct path load of SQLLDR. Values can be True and False. | TRUE |
| Parallel | <ul style="list-style-type: none"> • True • False | Parallel option. True means the loading happens with parallel option and False means it happens sequentially. | TRUE |
| Degree of Parallel | Integer | Decides the degree of parallelism. The values specified here must be a number. | 0 |

| Fields | Applicable Values | Description | Default Value |
|--|---|---|---------------|
| No: of Errors | Integer | Shows the number of errors allowed for the SQLLDR and External Table to proceed. That means a single record fails the job fails. The values specified here must be a number. | 0 |
| Maximum Discard | Integer | Discarded records allowed for SQLLDR. This must be set to a very high number when using multiple subtypes under an ADI. That means it is multiple targets. | 1 |
| XML date Format | Valid XML Date format | In this field, you can define the format of the XML Date. Example: MMDDYYYY. | MMDDYYYY |
| Avoid Partition Exchange | <ul style="list-style-type: none"> • Yes • No | <p>The property option decides whether the Partition Exchange method is used during the data load.</p> <p>NOTE: If the connector execution fails during partition exchange for some reason, then it can be avoided by choosing the option "Yes".</p> | No |
| Do you want to use Data Pump? | <ul style="list-style-type: none"> • Yes • No | <p>There are two values 'Yes' and 'No'. If the value is 'Yes', it indicates that the Oracle Database source is loaded into OFSAA using the Data Pump method. Alternatively, the standard way of using the DBLink method is followed.</p> <p>NOTE: The following access is required for the data pump option.</p> <p>Grant create any directory to Source schema</p> <p>Grant create any directory to the target schema</p> <p>Grant execute on DBMS_FILE_TRANSFER to the target schema</p> <p>Grant execute on utl_file to the source schema</p> | No |
| Source and Target in the Same Environment? | <ul style="list-style-type: none"> • Yes • No | This parameter is used only if the Data Pump is used. If the value is, 'Yes' then the file transfer step is not performed during loading. Alternatively, it will transfer files from source to target folder using DBLink. | Yes |
| Source Dump Location | /src/tmp | Specify a folder/path that is accessible by the Source Oracle Database to create the dump file. | /src/tmp |

| Fields | Applicable Values | Description | Default Value |
|--------------------------------------|---|--|---------------|
| Target Dump Location | /target/tmp | Specify a folder/path that is accessible by Target Oracle Database to read the dump file. | /target/tmp |
| Number of Splits for Dump File | Integer | Specify a number to transfer the files in parallel chunks. For example, 3 indicates that the dump file is split into three and transferred separately. This is to improve the performance of file transfer. The values specified here must be a number. | 0 |
| Effective Dated Key for Result Area? | <ul style="list-style-type: none"> • Yes • No | <p>This parameter is used for loading data into the result area. It decides whether it will perform a lookup into a dimension for the latest record, or an effective dated record. To get the latest record, it appends f_latest_record_indicator='Y' and for effective dated it appends mis_date between d_record_start_date and d_record_end_date.</p> <p>NOTE: For extraction data or any date attribute, effective dating does not work. If this parameter is selected as "Yes" and any date field needs to be loaded into an Extraction date or any other date field, then do not rely on surrogate key generation. Use the derived column and enter the value in the format "YYYYMMDD" as a number.</p> | No |
| Do you want to use DBLink? | <ul style="list-style-type: none"> • Yes • No | <p>This parameter is used to specify the source database connection method. There are two values 'Yes' and 'No'.</p> <p>If the value is 'Yes', it indicates a connection source database is created using the DBLink method.</p> <p>If the value is 'No', it indicates a connection to the source database is created using the JDBC URL.</p> | No |
| Hive Date Format | yyyy-MM-dd | This parameter is used to specify the date format for date columns in Hive source. | yyyy-MM-dd |

| Fields | Applicable Values | Description | Default Value |
|----------------------------------|---|--|---------------|
| Do you want to use Big Data SQL? | <ul style="list-style-type: none"> • Yes • No | <p>This parameter is used to specify the loading mechanism from the HIVE source.</p> <p>If you have selected Yes, Oracle Big Data SQL is used.</p> <p>NOTE: The following are the pre-requisites for this method:</p> <ul style="list-style-type: none"> • The Oracle Big Data SQL is installed and configured on Oracle Big Data Appliance and Database (OFSAA) machines. • In the OFSAA database, a directory named DEFAULT_DIR is created. • The folder/system path specified for the above directory must be accessible to the database (OFSAA). <p>For more information see: https://docs.oracle.com/bigdata/bda45/BDSUG/installing.htm#BDSUG-GUID-5CDA332E-8CB8-42CC-8922-09295E22B0E5</p> <p>If you have selected No, Oracle Loader for Hadoop is used.</p> <p>NOTE: The following are the pre-requisites for this method:</p> <ul style="list-style-type: none"> ▪ Oracle Loader of Hadoop connectors is installed and configured in the source system. ▪ The ODI agent is also configured for OLH. <p>For more information see: https://docs.oracle.com/cd/E72987_01/odi/odi-big-data/setup.htm#ODIBD117</p> | No |

6.3.7 Modifying and Viewing a Connector

To edit or view a connector, follow these steps:

1. To edit or view a connector, you can select the required connector from the connector summary.

NOTE

You can edit an existing connector, other than connectors in Published status. To edit a connector in Published status, you must first un-publish it and then open it.

2. The connectors are created open in their respective view mode.

3. The details of the selected connector are displayed. You can modify or view the details.
4. The Connector Name cannot be edited. Update the other required details.
5. Click **Save** to save the changes made.
6. To make changes to a published connector, click 'Unpublish'. The 'Unpublish' option clears the ODI metadata that has been created during publishing. Update the required changes and then click Publish. The updated changes are synced in ODI.

6.3.8 Copying a Connector

To copy an existing Connector, follow these steps:


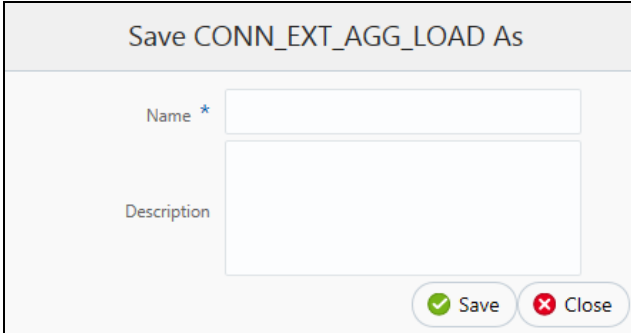
1. Click **Copy**  for the required connector. A **Save As** dialog box is displayed.

Figure 75: Save Connector As




Depending on the view in which the original connector is created, the copied connector will have the same view.

2. Enter the name and description.
3. Click **Save**. The Connector details are saved with a new specified connector name. The existing connector remains unmodified.

6.3.9 Deleting a Connector

To delete an existing Connector, perform the following steps:

1. Click **Delete**  for the required connector. A confirmation dialog box is displayed.
2. Click **Yes** to delete a connector. The Connector is deleted. If you do not wish to delete, click No.

6.3.10 Search and Filter

The Search and Filter option in the UI helps you to find the required information. You can enter the nearest matching keyword to search, and filter the results by entering information on the search box. You can search for a connector with either the name, description, or status of the connector.

For example, enter the keyword as 'CON_DRM_GL' in the search box. All the connector names with 'CON_DRM_GL' are listed.

You can sort the list by connector name or modified date (ascending or descending order).

6.3.11 Parameters in Connector

Parameters are used while defining the mapping, between EDD to ADI. While mapping the ADI to EDD, the fields or columns within the ADI needs to be mapped to the fields in EDD. If there are no corresponding extracts in EDD, parameters can be used to identify the default values for certain ADI elements. Also, parameters can be used while defining derived columns during mapping.

For example, if you want to use the Runtime MIS date as the parameter, then it must be converted to date first. The following is the expression:

```
To_char(to_date(#DIHDEV.MIS_DATE,'dd-MON-YYYY'),'MM')
```

NOTE

Runtime batch MIS date is in string format. It must be converted to DATE. The date format used here must be a valid SQL date format.

7 Processing DIH Definitions in Preparation for Executing Data Movement

As the standard Oracle tool for data extraction, loading, and transformation, ODI is employed by DIH to perform data movements based on EDS, EDD, ADI, and Connector definitions. Details of the ODI instance to be used are captured as detailed in the section Setting up ODI Connectivity. This section details how definitions are processed, published and registered in ODI by DIH in preparation for executing data movement.

Topics:

- [Refreshing Target Datastore](#)
- [Publishing Connectors and Underlying Metadata](#)

The first step in this sequence is to Refresh Target Datastore, as follows:

7.1 Refreshing Target Datastore

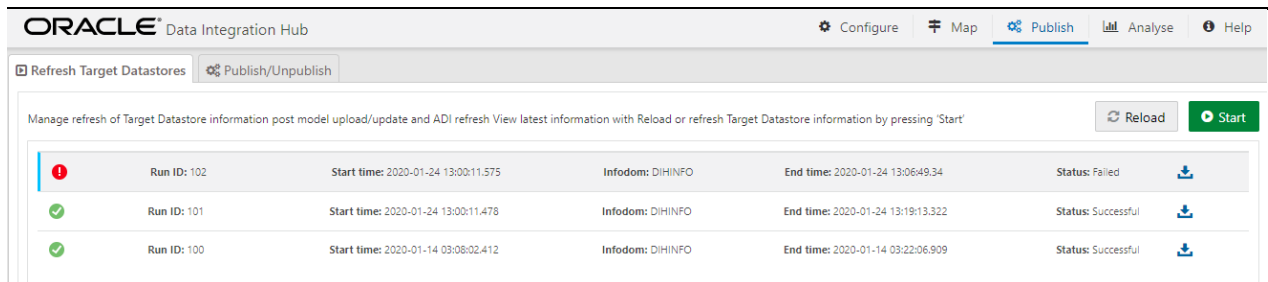
It represents underlying physical tables of every available ADI as a data store in ODI. This process also works incrementally.

Post data model upload, if the changes are to be made available in DIH, then first perform Refresh ADI. Once **Refresh ADI** is done, perform **Refresh Target Datastore**.

To refresh the Target Datastore list, follow these steps:

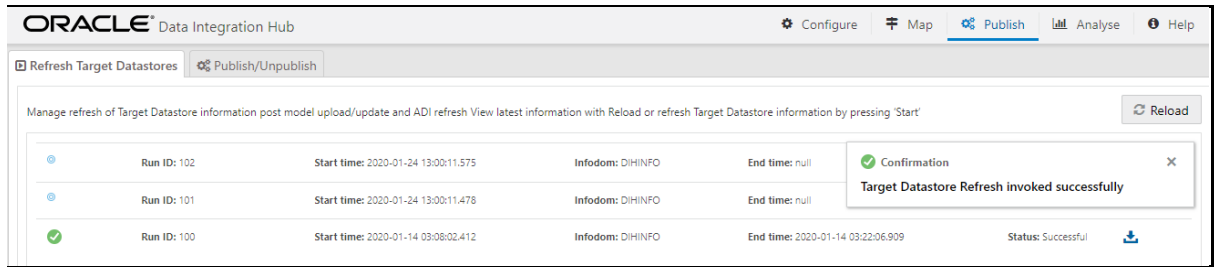
1. From the **Data Integration Hub Designer** window, select **Publish** and then select **Refresh Target Datastores**. The **Refresh Target Datastore Interface** summary is displayed.

Figure 76: Refresh Target Datastore

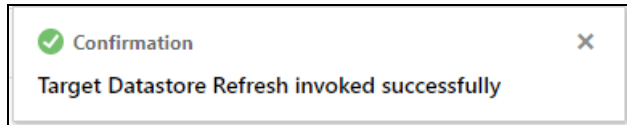



2. Click **Start** to start the refresh of the Target Datastore. The ongoing Target Datastore refresh is displayed as follows:

Figure 77: Target Datastore Refresh Confirmation



- On successful invocation of TDS refresh, a message is displayed.



- If you need a detailed running log, click  download the log.
- A zip file is downloaded containing the detailed log for the execution.
- To view the log details, extract the log file from the zip folder.

NOTE At any given time, click **Reload** to check if the execution is complete or still in progress.

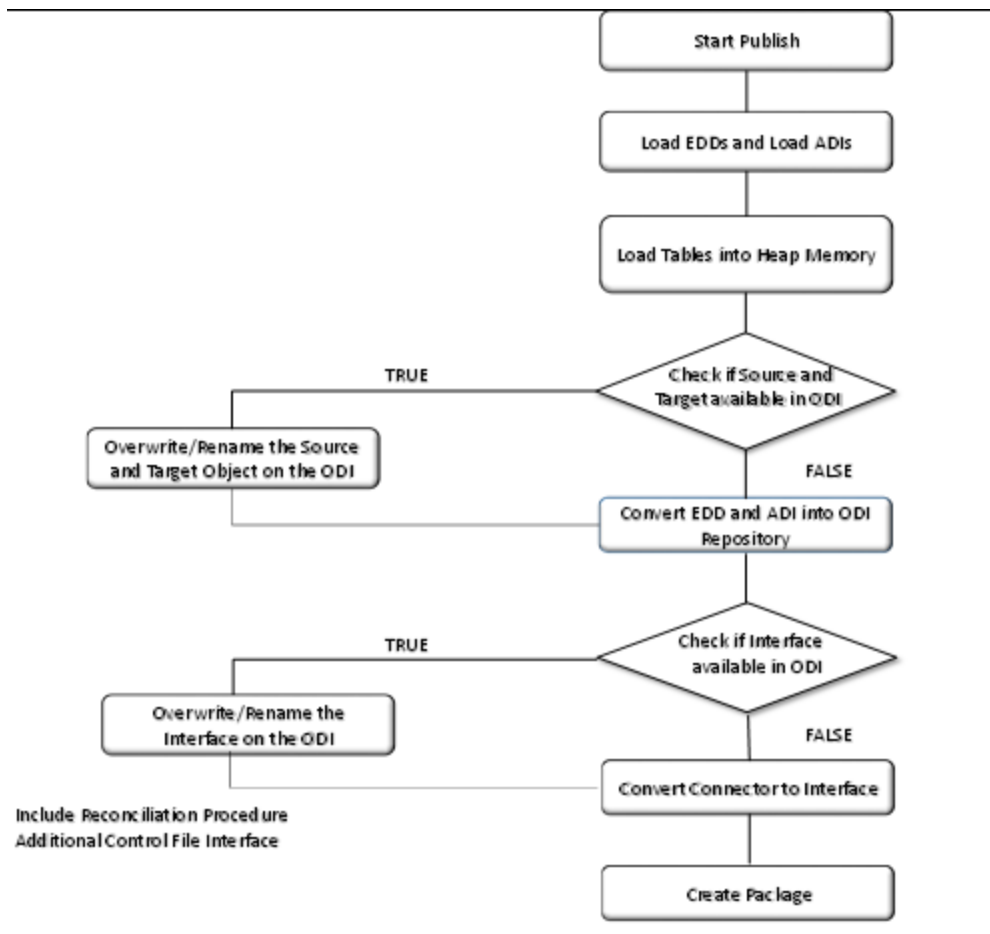
NOTE This is not a day-to-day activity.

7.2 Publishing Connectors and Underlying Metadata

Publishing is a process of transferring the DIH metadata to Oracle Data Integrator (ODI). It converts the DIH metadata into ODI metadata by creating required ODI objects such as DataServer, Logical/Physical Schema, Datastores, Interface, Package, and Scenario. You cannot edit any DIH object after publishing. To edit, you need to unpublish, make the necessary changes, and then re-Publish.

NOTE The flow diagram represents the activities undertaken upon publication. No action is required apart from triggering the Publish task.


Figure 78: Publish Connectors



7.2.1 Publish and Unpublish

You can batch publish/unpublish the selected connectors to ODI at any given point. You can also unpublish all the objects using this option.

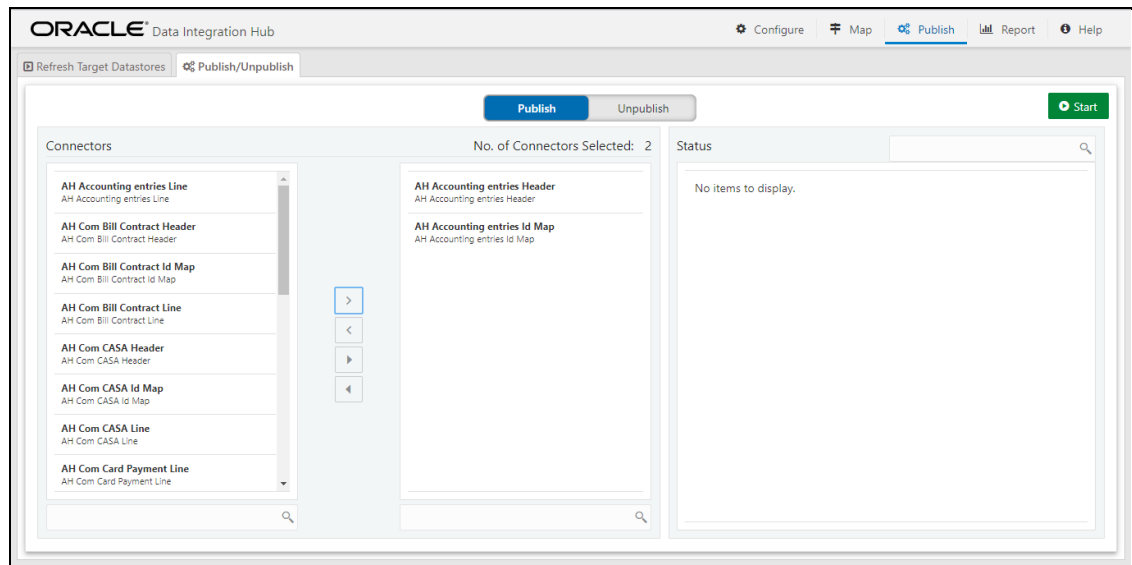
To publish, follow these steps:

1. From the **Data Integration Hub Designer** window, select **Publish** and then select **Publish/Unpublish**. The **Publish/Unpublish** window is displayed.
2. When you select the option Publish, all the saved and unpublished connectors are displayed.
3. Double-click the connectors or press Ctrl to select multiple connectors and click  to move it to the next column.

NOTE

To deselect a connector click .


Figure 79: Publish/ Unpublish Window



4. Click  and select all the connectors.

NOTE

To deselect all the connectors click .

5. Click Start after the items are selected. The status of the published connects are displayed. You can view the connectors, which are successfully connected or failed, under the Status field.
6. When you select the option as Unpublish, all the published connectors are displayed.
7. Double-click the connectors or press Ctrl to select multiple connectors and click  to move it to the next column.

NOTE


To deselect a connector click .

8. Click  and select all the connectors.

NOTE

To deselect all the connectors click .

9. If you select All Objects, then by default all the connectors and their corresponding EDD, EDS and parameters are unpublished.
10. Click Start after the items are selected. The status of the selected connectors is displayed. You can view the connectors, which are successfully published/unpublished or failed under the Status section.

11. Search  the connector to check the status of the publish/unpublished process.

8 Executing Data Movement Based on Definitions

A published DIH Connector can be executed using the following methods.

Topics:

- [External Batch Management Tool](#)
- [OFSAA Batch](#)
- [Process Modelling Framework](#)
- [Specifying Runtime Parameters](#)

8.1 External Batch Management Tool

A DIH connector, with ETL job instructions, is published in ODI as a Package object. The ODI package object shares the same name as a DIH connector. These packages can be executed by generating a scenario out of it. External Batch Management tools can be used to execute these scenarios. See ODI documents for more information on generating and executing an ODI scenario.

8.2 OFSAA Batch

DIH creates a typical OFSAA Batch and executes it. For more information, see [OFS AAI Process Modelling Framework Orchestration Guide](#). The component is DIH Connector. You can select all the published connectors during task creation. DIH Connectors can be configured in the Run Rules Framework (RRF) also with the same component.

NOTE

- This can be done only on a published connector and not a saved connector.
- Currently, there is a limitation of the passing parameter in the Run Rule Framework for DIH connectors. The batch framework can be used when the runtime parameter must be passed.

If the connector contains any Runtime parameter, see section [Specifying Runtime Parameters](#) for more details.

8.3 Process Modelling Framework

Process Modelling Framework (PMF) is a design and execution framework that enables Process Pipeline developers to implement various Pipelines modeled by business analysts. Process Pipeline developers use the framework to orchestrate the Business Pipelines and Run Pipelines within OFSAA, and also to design the artifacts that participate in the Pipelines, to complete their implementation. The Process Modeling Framework consists of Process Modeling components for modeling Pipelines and Process Monitor components for monitoring instantiated Pipelines of OFSAA applications. For more details, see [OFS AAI Process Modelling Framework Orchestration Guide](#).

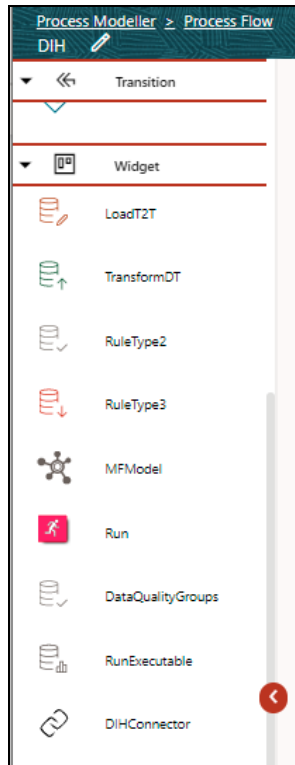
To add a connector task in PMF Pipeline, perform the following steps:

1. Open a process pipeline in which you want to add a connector task.

- Click  to expand the left menu.

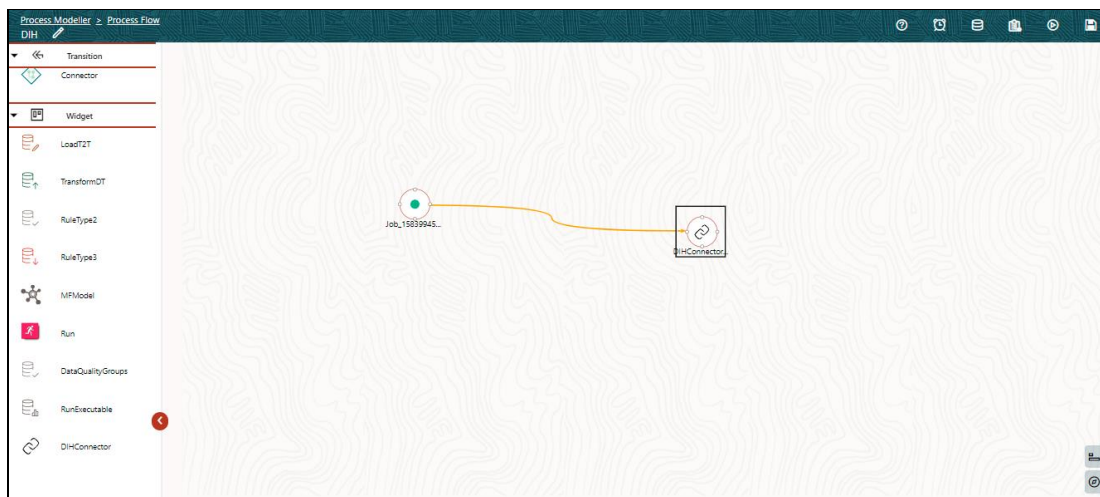
You can see **DIH Connector** under **Widget**.

Figure 80: Process Modeller Process Flow Window



- Select **DIHConnector** and drag and drop it on the canvas.



Figure 81: Process Modelling Framework Tile



- Double-click **DIHConnector** to select the connector to be executed.

Figure 82: DIHConnector

5. Specify the **Activity Name** and **Activity Description**.
6. Under the **Dynamic Parameters for DIHConnector**:
 - a. Select the required **Connector**.
 - b. Enter the Runtime **Variables**. If the connector contains any Runtime parameter, see section [Specifying Runtime Parameters](#) for more details.
 - c. Select the **Agent**.

7. Enter all the details and click  to Save the details. You can also click  **Save** icon to save the details.

A confirmation message is displayed as Successfully Saved.

8.4 Specifying Runtime Parameters

The following are the runtime parameters:

- If the connector contains any Runtime parameters, they can be set in the Variables input field of the Task Definition window.

Example: MISDATE='10-Jan-2015'

If there are multiple parameters, they can be passed by separating them with a comma.

Example: MISDATE='10-Jan-2015', BATCHID=22015

- MISDATE and BATCH ID can also be passed dynamically.

Example: MISDATE=\$MISDATE:dd-MM-yyyy, BATCHID=\$BATCHID

- In this example, the date format appended to MISDATE has to conform to the Simple Date Format. If no date format is specified, the default date format used is yyyyMMdd.
- If variables are used as part of connector mappings or filter expressions, they must be passed within single quotes as follows:

Example: MISDATE=' \$MISDATE:dd-MM-yyyy' , BATCHID=' \$BATCHID'

- If the date format is expected in dd-MON-yyyy format, then it must be specified in Batch Task in the following format. Note the difference in month format in the following example:

Example: MISDATE=' \$MISDATE:dd-MMM-yyyy'

- If the parameter is used in connector filter expression for an EDD of source type Hive, the date format is expected in yyyy-MM-dd format.

Example: MISDATE=' \$MISDATE:yyyy-MM-dd'

9 Analyzing Operational Details, Utilization of Available Artifacts and User Activity

DIH provides built-in facilities for you to analyze details of data movement operations performed under its auspices. The following sections provide details on how this may be accessed and used. Also, DIH provides information on the utilization of the data model in use, split by applications deployed. Utilization information also covers Application Data Interfaces and Connectors.

Topics:

- [Execution History](#)
- [Utilizing Available Artifacts](#)
- [DIH Activity](#)

9.1 Execution History

Execution History provides the status of Connector executions, covering details of volume, outcome, and performance.

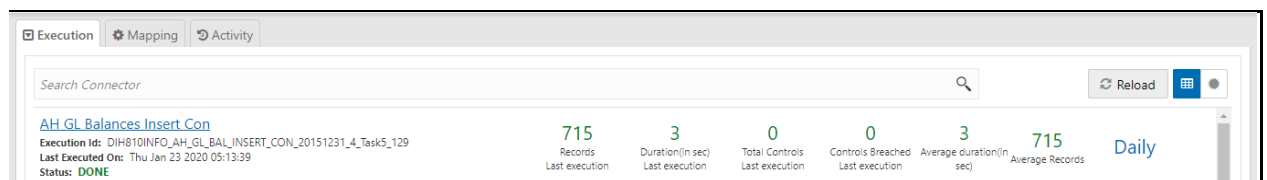
To view the Execution History of a Connector, follow these steps:

1. From the Data Integration Hub Designer window, select Analyse, and then select Execution. The **Execution** window is displayed. The Connectors are listed in the sequence of their last execution. It also provides information on the following.
 - Number of records loaded in the last execution
 - Duration
 - Total Controls defined for all EDDs in the connector
 - Number of controls breached for the last execution
 - Average Duration
 - Average Records
 - Frequency

NOTE

The frequency is classified as Daily, Weekly, Monthly, Bi-Monthly, and Quarterly. This value is decided automatically depending on the frequency of the connector execution in the specified environment.

Figure 83: Execution Window





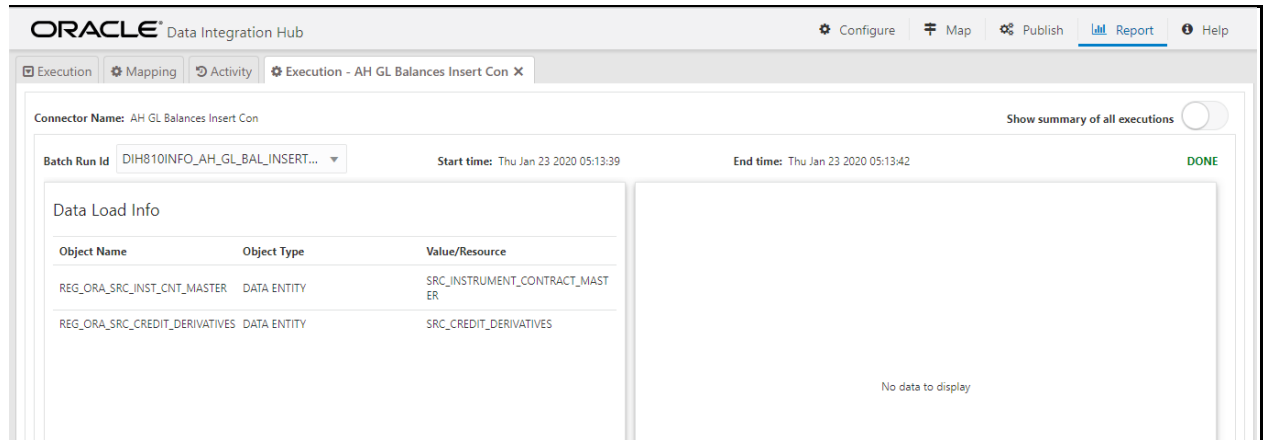
2. You can view the summary details of all the connectors that are executed in either the **Execution Report**  or **Performance View**  to view the connector performance heat map.

Figure 84: Execution Performance View



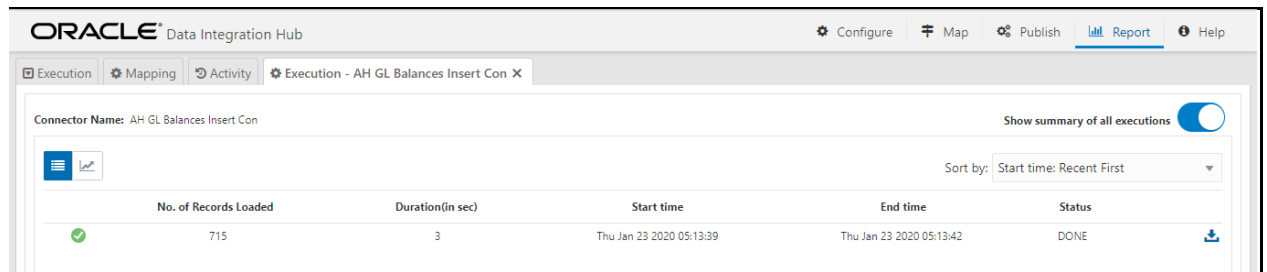
3. The search bar helps you to find the connector for which you can view the executions. You can enter the nearest matching keywords to search and filter the results by entering information on the search box. You can search for a connector using either the name or description.
4. Click the required Connector under Execution History.
5. Select the Batch Run ID from the drop-down list to view the executions. You can view the Data Load information.

Figure 85: Execution Window



- To view the summary of all executions, select Show summary of all executions.

Figure 86: Execution Summary Of All Executions



- Here you can view the number of records loaded, duration, start time, end time, and the status of the last execution.

NOTE Due to limitations in ODI, the number of records while loading data into Hive might display as zero.

- If the status displays as Error, click **the link** to view details about the execution. The error details failed source command and failed target command is displayed.

Figure 87: Execution Details

Execution Details

VARIABLE0 RUNTIMEVARIABLE0 RUNTIMEVARIABLE1 AH GL Balances Insert Con_100

Start time: Thu Jan 31 2019 02:13:05 End time: Thu Jan 31 2019 02:13:08 Status: **ERROR**

Error Failed Source Command Failed Target Command

ODI-1226: Step AH GL Balances Insert Con_100 fails after 1 attempt(s).
 ODI-1227: Task SERIAL-MAP_MAIN- fails on the source connection <Empty Value>.
 ODI-1298: Serial task "SERIAL-MAP_MAIN"- (10) failed because child task "SERIAL-EU-INTF_AH_OIDF_TAR_FILES_UNIT (20)" is in error.
 ODI-1298: Serial task "SERIAL-EU-INTF_AH_OIDF_TAR_FILES_UNIT (20)" failed because child task "Call sqldr-LKM MultiFiles to Oracle (SQLLDR)- (100)" is in error.
 ODI-1590: The execution of the script failed.
 Caused By: org.apache.bsf.BSFException: exception from Jython:
 Traceback (most recent call last):
 File "<string>", line 19, in <module>
 Load Error: See /scratch/software/ODI/data_files/AH_Files/L_FAH_OFSA SR_Balance_480.log for details

at org.apache.bsf.engines.jython.JythonEngine.exec(JythonEngine.java:146)
 at com.sunopsis.dwg.codeinterpreter.SnpScriptingInterpreter.execInBSFEngine(SnpScriptingInterpreter.java:396)
 at com.sunopsis.dwg.codeinterpreter.SnpScriptingInterpreter.exec(SnpScriptingInterpreter.java:247)
 at oracle.odi.runtime.agent.execution.interpreter.SessionTaskScriptingInterpreter.scripting(SessionTaskScriptingInterpreter.java:190)
 at oracle.odi.runtime.agent.execution.SessionTask.scripting(SessionTask.java:105)
 at oracle.odi.runtime.agent.execution.cmd.ScriptingExecutor.execute(ScriptingExecutor.java:49)
 at oracle.odi.runtime.agent.execution.cmd.ScriptingExecutor.execute(ScriptingExecutor.java:21)
 at oracle.odi.runtime.agent.execution.TaskExecutionHandler.handleTask(TaskExecutionHandler.java:52)
 at oracle.odi.runtime.agent.execution.SessionTask.processTask(SessionTask.java:206)
 at oracle.odi.runtime.agent.execution.SessionTask.doExecuteTask(SessionTask.java:117)
 at oracle.odi.runtime.agent.execution.AbstractSessionTask.execute(AbstractSessionTask.java:886)
 at oracle.odi.runtime.agent.execution.SessionExecutor\$SerialTrain.runTasks(SessionExecutor.java:2225)



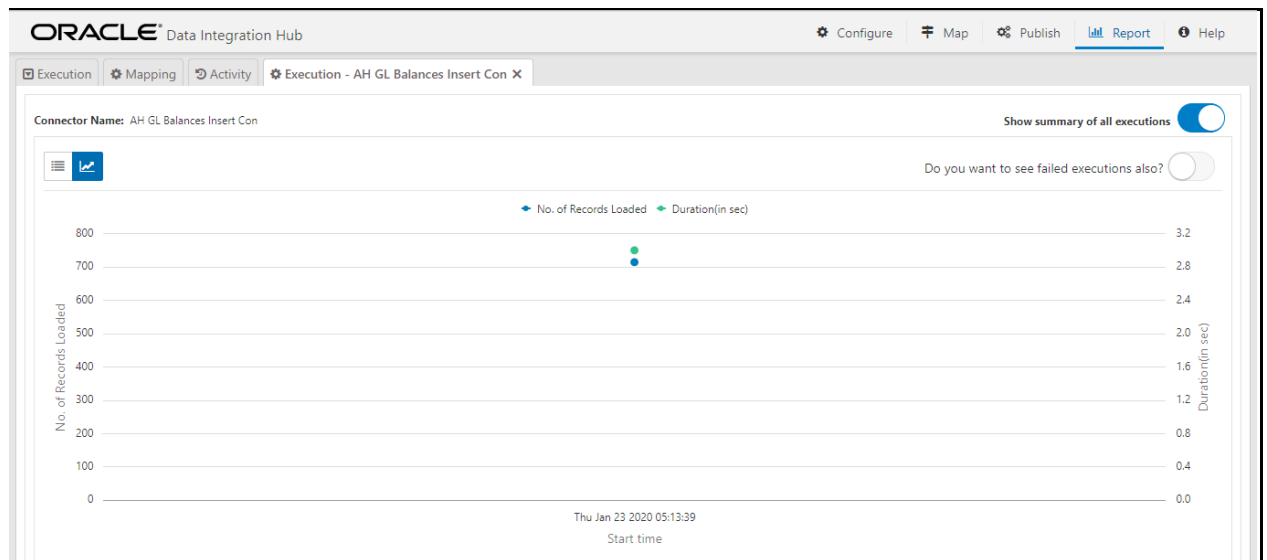
9. If you wish to see the detailed execution report, click **Download Log** . A zip file is downloaded containing the detailed log for the execution.
10. To view the log details, extract the log file from the zip folder.
11. If the control definition is defined for file type EDD, the report of the expected and actual results loaded is displayed.
12. Click  to view the report for the required connector.

Figure 88: Connector Report



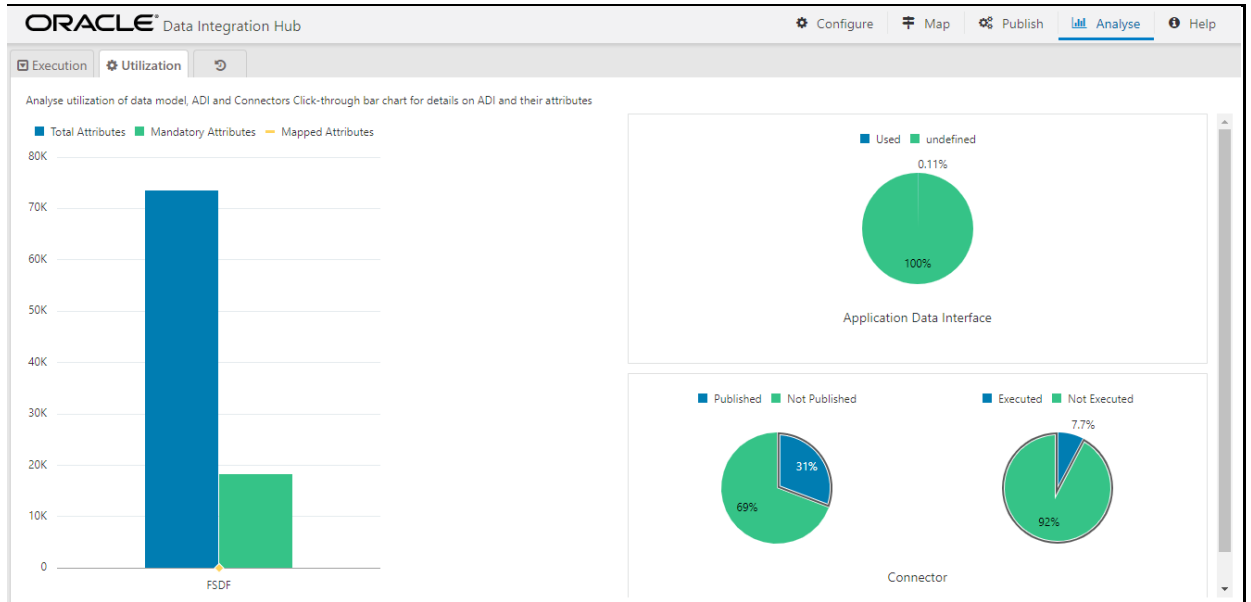
13. If you wish to view the failed executions, select “Do you want to see failed executions also?”

9.2 Utilizing Available Artifacts

To analyze the utilization of data model, ADI, and Connectors, follow these steps:

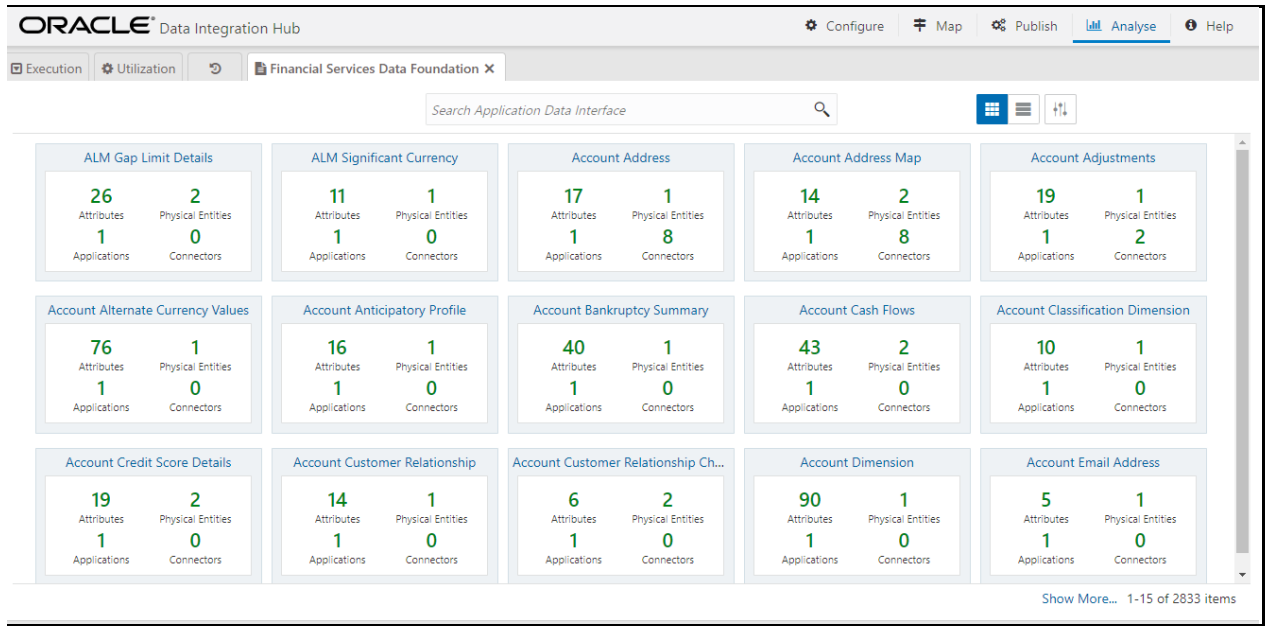
1. From the Data Integration Hub Designer window, select **Analyse** and then select **Utilization** to view the **Mapping Report** for the ADI and Connectors.
2. The **Utilization** window displays the graph of total attributes, mandatory attributes, mapped attributes along with Connector details.

Figure 89: Utilization Window



3. Click **Total** or **Mandatory** or **Mapped Attribute** graph. This opens the **ADI Summary** window.

Figure 90: ADI Summary Window



4. Select the required ADI. The **Mapping Report** window is displayed. This displays the number of attributes mapped for the ADI (subtype) in each connector created for the same.

Figure 91: Mapping Report

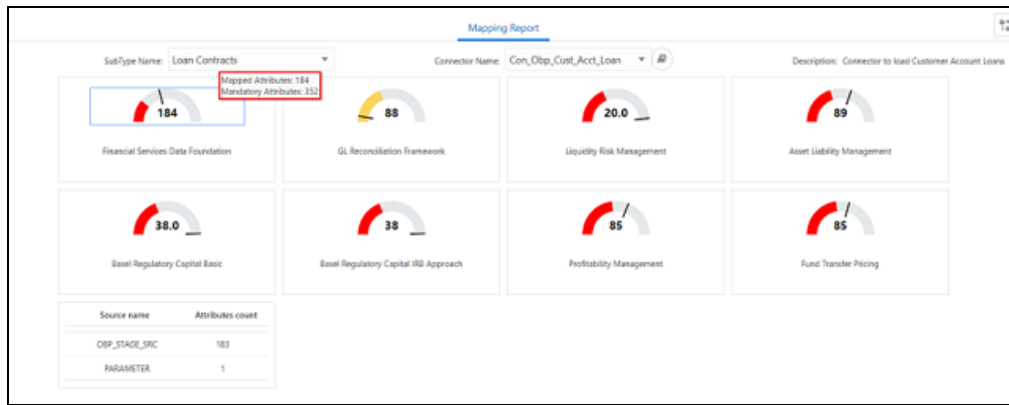
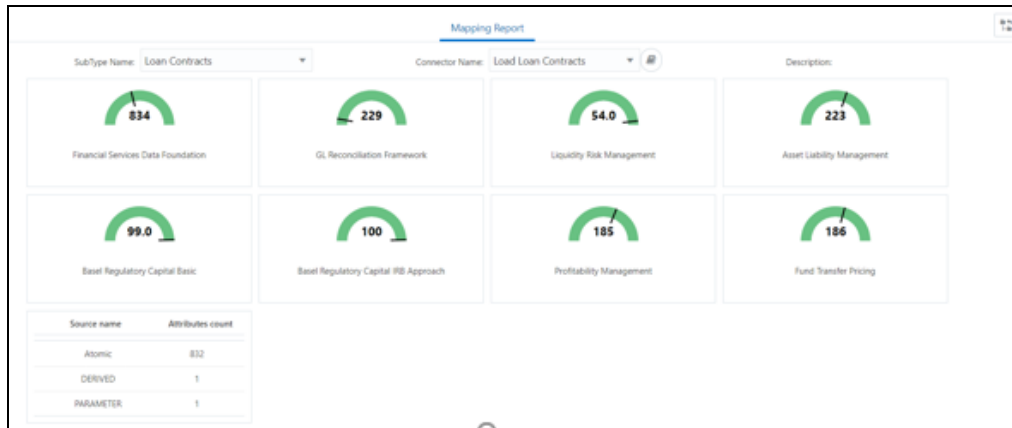
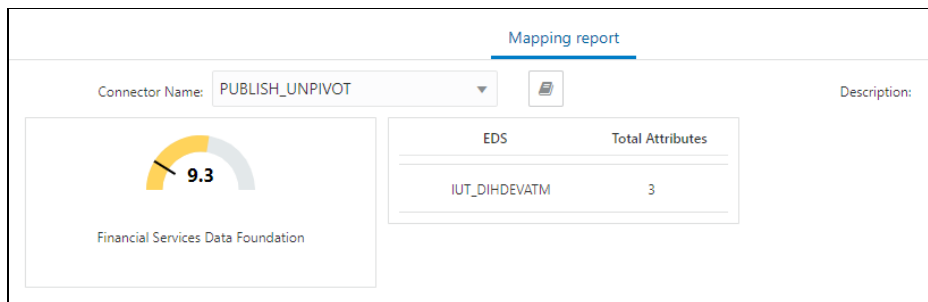
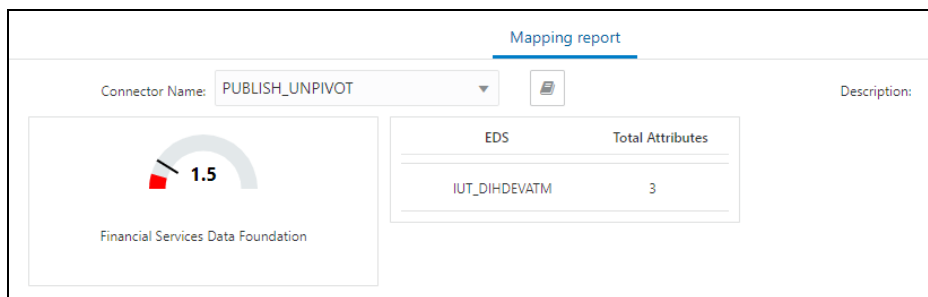
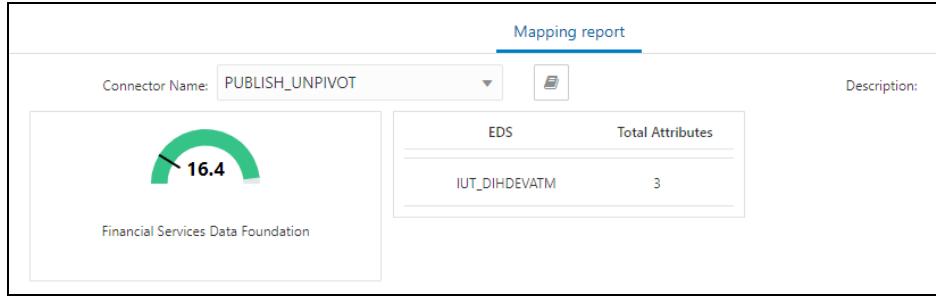


Figure 92: Mapping Report



5. Select the **Connector Name** from the list of connectors, which are populating that particular ADI.
6. Depending on the connector and subtype you select, the attribute report is displayed.
7. The report displays the total number of mapped attributes and a total number of mandatory attributes for that particular application.
 - Mapped attributes which are less than mandatory attributes display in **Red**.
 - Mapped attributes which are greater than mandatory attributes display in **Yellow**.
 - Mapped attributes are the same as the total number of attributes displayed in **Green**.





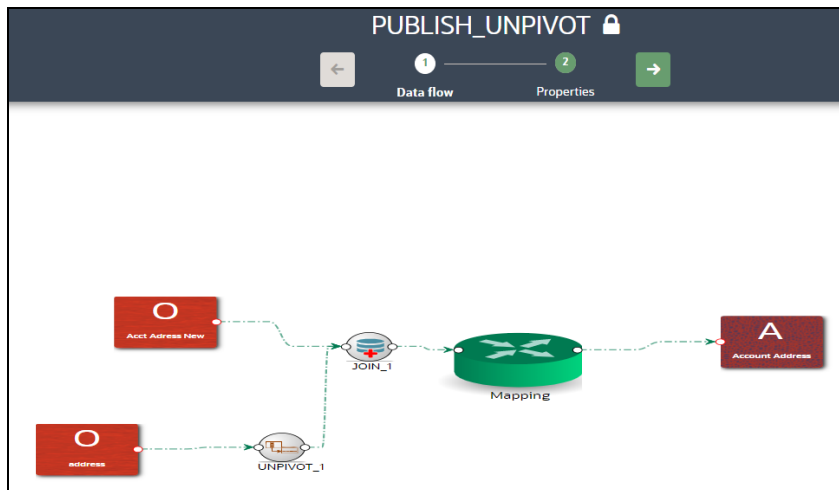
It also displays the EDS name, the number of attributes sourced from that particular EDS.

| Source name | Attributes count |
|-------------|------------------|
| Atomic | 832 |
| DERIVED | 1 |
| PARAMETER | 1 |

- Click  to View Connector.



Figure 93: Publish Unpivot



9.3 DIH Activity


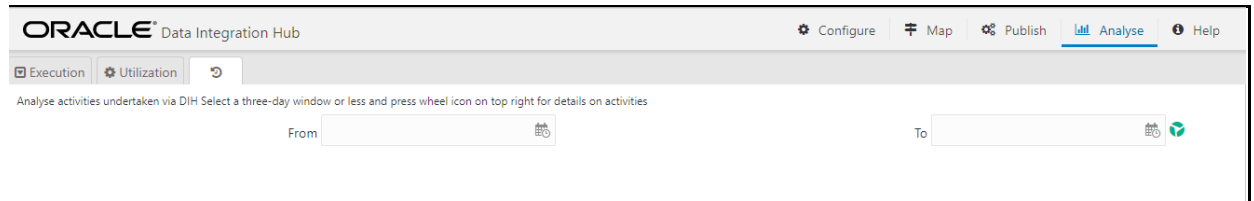

1. From the **Data Integration Hub Designer** window, select **Analyse**, and then select . The **Analyze Activity** window is displayed.

Figure 94: Analyze Activity



2. You can analyze activities undertaken through DIH over a maximum range of three calendar days. You can view the activities of any creation, update, and deletion of the metadata. You can view the publish/unpublish or refresh activity in this window.
3. In **From** and **To** fields, select **Date** and **Time** and click  **Fetch** info to view the results.

10 Command Line Utilities

Using the command-line utility, you can invoke both Refresh Application Data Interface and Refresh Target Datastore process.

Topics:

- [Prerequisites](#)
- [Invoking the Command Line Utilities](#)

10.1 Prerequisites

- You must have access and execute permission to the following directory:
\$FIC_HOME/ficdb/bin
- If the secured protocol is enabled for accessing the OFSAA application then “CURL_CA_BUNDLE” environment variable must be set where the application is installed. The variable points to the path where the CA certificate is available that is generated during application deployment.
- For example: CURL_CA_BUNDLE=/usr/share/ssl/certs/ca-bundle.crt.

10.2 Invoking the Command Line Utilities

To invoke the command-line utility for refreshing ADI and Target Datastore, follow these steps:

1. Once the prerequisites are met, navigate to \$FIC_HOME/ficdb/bin path and execute RefreshADI.sh and then RefreshDS.sh. This expects two mandatory parameters:
 - a. The information Domain name where DIH is installed. It is mandatory to enter this in uppercase.
 - b. OFSAA Login ID, which can be specified in either lower or uppercase.
2. For example, see the following window.
 - a. For **Refreshing Application Data Interface**, use the following command:
 - b. `./RefreshADI.sh <INFODOM_NAME> <OFSAA_USER>`

Figure 95: Refreshing Target Datastore command

```

/scratch/dihodi/DIHHOME/ficdb/bin>./RefreshADI.sh OFSDIINFO OFSAD
-
infodomNameOFSDIINFO
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
   0      0     0     0         0     0         0     0    0:00:08  --:--:--    0
  
```

- c. For **Refreshing Target Datastore**, use the following command:
- d. `./RefreshDS.sh <INFODOM_NAME> <OFSAA_USER>`

Figure 96: Refreshing Target Datastore command

```

/scratch/dihodi/DIHHOME/ficdb/bin>./RefreshDS.sh OFSDIINFO OFSAD
F
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total      Spent    Left     Speed
0         0         0         0         0         0         0         0  --:--:--  0:00:04  --:--:--  0

```

NOTE

If the webserver is Websphere, an error message is displayed in the console as “Refresh Target Datastore as fail”.

Verify this from the application window, and ignore the message in the command prompt and proceed.

If the webserver is Weblogic, an error message is displayed "curl: (35) SSL connect error". In this case, invoke the process from the application window and proceed.

11 Metadata Object Migration

Metadata objects created by DIH may also be migrated leveraging this mechanism, with the list including DIH Parameters, EDS/EDD/Connector definitions, and batches.

Topics:

- [Overview](#)
- [Performing Object Migration](#)
- [Exporting Object from Source Environment](#)
- [Importing Object into Target Environment](#)

11.1 Overview

Using the object migration utility, you can migrate (export/ import) DIH metadata objects across different setups. You can specify one or more objects within an object type or multiple object types. You can choose from where the Object Migration utility reads the data, that is, from CSV files or the OBJECTMIGRATION.xml file. For migrating objects using CSV files and OBJECTMIGRATION.xml file, see section Command Line Utility to Migrate Objects from [OFSAAI User Guide](#).

The following sections detail how such Metadata Object migration can be achieved.

11.2 Performing Object Migration

Offline Object Migration is a two-step process:

- Exporting Objects from Source Environment
- Importing Objects into Target Environment

For both of these steps, see the sample file **OBJECTMIGRATION.xml**, which is also present at `$(MIGRATION_HOME)/conf/` in the OFSAAI setup.

Assumptions and Pre-requisites for this process are listed as follows:

- Both the Source and Target environment must be on the same data model version.
- Ensure that the source and target are at the same DIH application level as well as at the same model level before migrating objects from source to target.
- ADI refresh is completed before this process.
- ODI settings are correctly specified in the target environment before importing objects.
- Target datastore refresh is also performed and successful.
- If the exported object already exists in the target environment or an object with the same name already exists in the target environment, then it is overwritten with new definition details.

If the exported Parameter already exists in the target environment, it is not overwritten unless the new definition is of a different Parameter Type. If exported EDS already exists in the target environment, it is not overwritten unless the new definition is of different EDS Types.

- Any exported object, if already exists in target or an object with the same name exists in target then that object and all its dependent objects must be unpublished for migration to go through successfully.

Dependent objects for a Connector are EDS, EDD, and Parameter. Dependent objects for EDD are EDS and Parameter. Parameter and EDS do not have dependent objects.

11.3 Exporting Objects from Source Environment

To export objects from the source environment, follow these steps:

1. Replace placeholders of UserID, Infodom with Source UserID, and Infodom.
2. For \$Folder put segment name for the infodom provided. Enter locale as 'en_US'.
3. \$FILE_NAME: Specify the file name which is created under "metadata/archive" folder. For example, mention 'rules' in place of \$FILE_NAME and you get **rules.dmp** in the archive folder.

Fail On error: Fail on any error that occurs while restoring metadata. Mention 'Y' for Yes and 'N' for No.

OVERWRITE: If Metadata exists in the system then Overwrite while restoring metadata. Enter 'Y' for Yes and 'N' for No.

In Mode tag: mention EXPORT.

For **FAILONERROR** and **OVERWRITE**, it's recommended to mention 'Y'.

4. In the OBJECT tag, enter "*" for Code property, to export all definitions. If not, for each definition put an equal number of OBJECT tags with map ID and version number in comma-separated format.

Type: Use the following for definitions:

- 70 for Parameter Definitions
- 71 for EDS Definition
- 72 for EDD Definition
- 74 for Connector Definition

5. The format for the All OBJECTS tag is as follows:

```
<OBJECTS TargetFolder="DIHUSERS"><OBJECT Code="*" Type="73" /></OBJECTS>
```

6. For three definitions, the OBJECTS tag is:

```
<OBJECTS TargetFolder="DIHUSERS">
<OBJECT Code="123221" Type="73" />
<OBJECT Code="321331" Type="73" />
<OBJECT Code="131213" Type="73" />
</OBJECTS>
```

7. Execute \$MIGRATION_HOME/bin/ObjectMigration.sh after providing executable permissions.

8. A file \$FILE_NAME.dmp (rules.dmp) is created in \$MIGRATION_HOME/metadata/archive

9. Move this file to `$MIGRATION_HOME/metadata/restore` folder. You can copy the file in the Target environment by creating a "restore" folder under the "metadata" directory (if not available).
10. Exporting definitions from the Source environment is done successfully.

11.4 Importing Objects From Target Environment

To import objects to the target environment, follow these steps:

1. Repeat 1-3 steps as followed in export mode.
2. In the Mode tag: mention IMPORT.
3. In the OBJECT CODE property, mention "312321".

NOTE

Everything that is exported is imported. You cannot choose specific definitions to move.

4. The format for the OBJECTS Tag is:

```
<OBJECTS TargetFolder="DIHUSERS">  
<OBJECT Code="312321" Type="73" /></OBJECTS>
```

5. Execute `$MIGRATION_HOME/bin/ObjectMigration.sh` after providing executable permissions.
6. Check **DIH.log** for logs. It provides details such as the number of definitions that have successfully moved and other errors. Importing objects to the target environment is done successfully.

12 Metadata Browser

The Metadata Browser function allows you to view and analyze all aspects of the metadata used in OFSAAI.

Topics:

- [Overview](#)

12.1 Overview

Metadata Browser provides extensive browsing capabilities of metadata, helps in tracking the impact of changes to metadata, and trace through to the source of the originating data.

The DIH metadata or objects which are available in the Metadata Browser (MDB) are as follows.

- Application Data Interface
- External Data Descriptor
- Connector

For detailed usage information on Metadata Browser, see the [OFSAA Metadata Browser User Guide](#).

13 Data Domain Browser

Data Domain Browser (DDB) is intended to help logically visualize the underlying Oracle Data Foundation instance and navigate its content in a structured manner. The information represented in Data Domain Browser, abbreviated to DDB henceforth, is curated and maintained by Oracle, aligned with the release and version of Data Foundation it is shipped alongside. Data Domain Browser cannot be used independently of Oracle Data Foundation – Financial Services Data Foundation or Insurance Data Foundation.

DDB facilitates structured navigation of data foundation, using Data Domains, Data Sub-Domains, Business Segments, Logical Entities, Entities, and Attributes, all of which are described in detail in the following sections.

DDB also supports searching for information on Data Foundation in common business terms and exporting results thereof – these are also detailed in the following sections.

Data Domain Browser is currently available in Data Integration Hub for Oracle Insurance Data Foundation.

13.1.1 Prerequisites for Data Domain Browser

13.1.1.1 Required Software Licenses

The following software must be licensed and installed to access and use the Data Domain Browser:

- Oracle Insurance Data Foundation Pack v8.0.8.1.0 ML release patch **30257021** available at [My Oracle Support](#)

13.1.1.2 Audience

The following user roles are expected to leverage Data Domain Browser for their business functions, and will benefit from a detailed read of this guide:

- **Business Users:** DDB will support visualizing the data foundation in business terms such that its details are easily navigated and understood.
- **Data Office, Technologists, and Operators:** Categorized information on data content will be made available to Data Office through DDB's structured navigation and curated information content.

13.2 About Oracle Insurance Data Foundation

Oracle Insurance Data Foundation is a data platform to facilitate a comprehensive set of quantification and analysis functions across multiple business use-cases, as relevant to insurers. OI DF combines an industry-specific data model for insurance with a set of data management and infrastructure tools/facilities towards this goal. It is intended to enable insurers to develop, deploy, and operate quantification/analysis solutions covering functional areas including risk management, performance management, actuarial processing, actuarial-risk-finance integration, and customer insight. OI DF facilitates the management of the whole analytical data lifecycle from sourcing to reporting, via its data model, platform, and infrastructure tools.

Functionally, OI DF supports a comprehensive set of insurance business segments under the Direct Insurance Contracts domain, including life, health, property & casualty, and retirement policies, alongside annuities. Additionally, under the Indirect / Reinsurance domain, it supports reinsurance held and reinsurance issued business segments.

While working with OI DF, you will continually interact with and use Staging and Results models. OI DF Staging Model provides data structures for ingesting and hosting data that is typically sourced from external sources in raw form. OI DF Results Model provides data structures for processed information in the form of a cluster of entities with factual information alongside a set of conformed dimensions that retains a common, consistent meaning across all factual information.

Data Domain Browser will pick, process, and render properties including entity names/descriptions, entity attributes/descriptions / data-types, and keys, across Staging and Reporting Models. OI DF data model entities that exclusively support processing, configuration, and related parameters are not covered by DDB.

For more information on OI DF, see the [Oracle Insurance Data Foundation Application Pack](#) user guide.

13.3 Using Data Domain Browser

This chapter explains how to use Data Domain Browser using the DIH application.

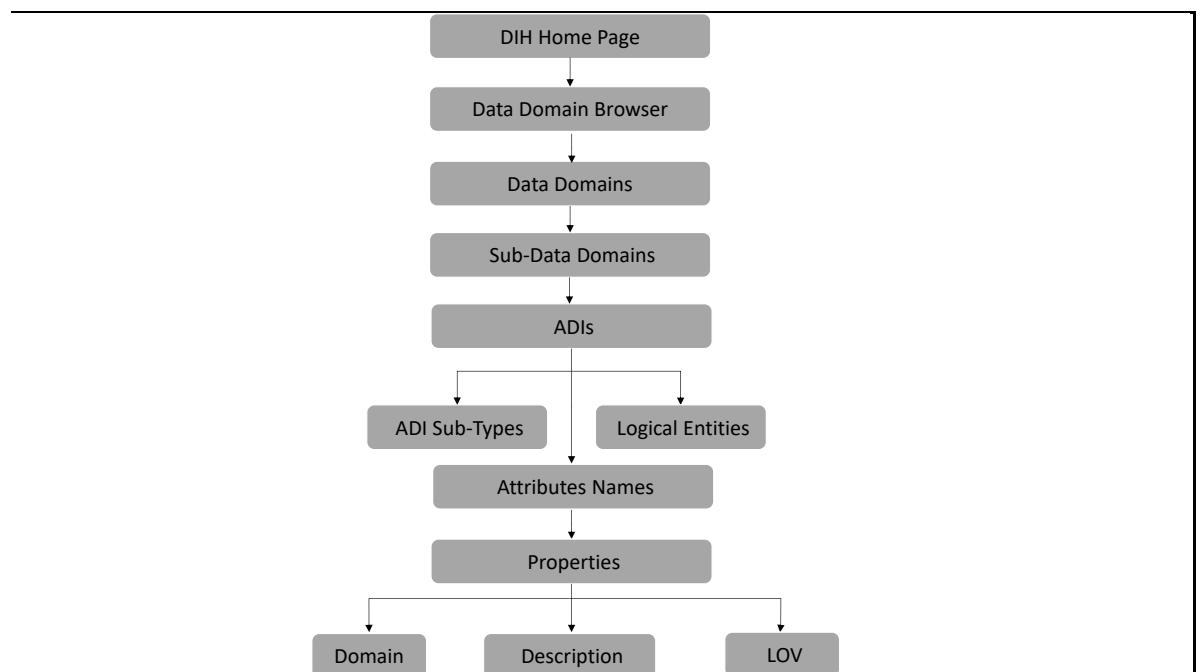
13.3.1 Launching DDB

After logging into the application, select **Financial Services Data Integration Hub**. The DIH window appears. Click the **Data Domain Browser**.

13.3.2 Data Domain Browser Navigation

The Data Domain Browser Navigation is as follows:

Figure 97: Data Domain Browser Navigation Path



DDB employs the following navigation path to render its information content in a structured manner – Data Domains, Data Sub-Domains, Application Data Interfaces (Sub Types and Logical Entities), and Attributes. You will read details about each of these in the sections below.

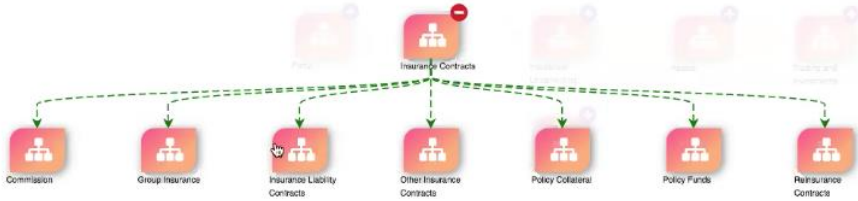
Each Data Domain is relevant across multiple Segments and could be one of two types - Download or Results. Segments and Types are intended for filtering results presented by DDB. The list of ADIs, when filtered by Segment “Life Insurance Contracts” and type “Results” will show only those logical entities that hold processed results relevant to life insurance contracts, for example.

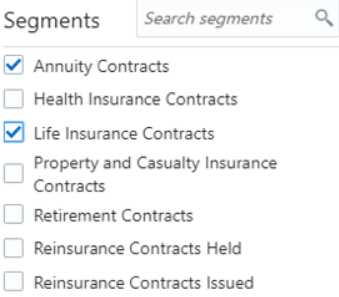
The finest grain of detail rendered by DDB is Attributes and their properties, including Description and List of Values, the latter, only when seeded by the application.






13.3.3 Information Rendered by DDB

The tabulation below lists key information content/constructs rendered by DDB and provides a definition or description of use, with examples:

Table 28: Information Rendered by DDB

| | |
|-------------------------------|--|
| <p>Data Domain</p> | <p>Data Domains are akin to Subject Areas in Data Foundation and a group of related entities thereof.</p> <p>These refer to key actors (Party, for example), activities (Accounting), or business functions (Insurance Underwriting), typical at insurers.</p> <p>The following Data Domains are covered as of this version:</p> <ul style="list-style-type: none"> • Accounting and General Ledger • Actuarial • Insurance Contracts • Insurance Underwriting • Party <p>For detailed information on Subject Areas, Data Domains, and Data Sub-Domains, see the latest version of the Oracle Insurance Data Foundation Application Pack user guide.</p> |
| <p>Data Sub Domain</p> | <p>Data Sub Domains expand Data Domains into finer sections, each self-contained and functionally complete in itself but inter-related in the context of the Data Domain it belongs to.</p> <p>For example, Insurance Contracts Data Domain expands into Insurance Contracts, Re-insurance Contracts, Group Insurance, Policy Funds, Policy Collateral, and Commission Sub Domains. Each of these is functionally complete, allowing users to navigate underlying data assets relevant to them. In the context of Insurance Contracts, these are related in a specific way.</p> <p style="text-align: center;">Figure 98: Data Sub Domain</p>  |

| | |
|-----------------------|--|
| | |
| <p>Segment</p> | <p>Each Segment refers to a typical business segment at insurers, with a group of products that are differentiated from those of other business segments based on their attributes, customers, geographic relevance, or marketplaces.</p> <p>The following Business Segments are covered as of this version:</p> <p style="text-align: center;">Figure 99: Segment</p>  <p style="text-align: center;"> <ul style="list-style-type: none"> • Annuity Contracts • Health Insurance Contracts • Life Insurance Contracts • Property and Casualty Insurance Contracts • Retirement Contracts • Reinsurance Contracts Held • Reinsurance Contracts Issued </p> <p>Each domain consists of multiple Segments. Each Segment, in turn, will be relevant to one or more Domains.</p> |
| <p>Tag</p> | <p>Tags refer to labels attached to data artifacts (entities or attributes) uniquely indicating similarity or commonality, for purposes of easy identification and search.</p> <p>Some entities related to a specific Domain may be assigned a common tag. For example, “Insurance Contract Monetary Values” may be used to tag all entities related to the “Insurance Contracts” Domain that contain monetary values.</p> <p>Each tag may go across multiple Segments. In the above example, “Insurance Contract Monetary Values” will apply to entities in the following Segments: Annuity Contracts, Health Insurance Contracts, Life Insurance Contracts, Property and Casualty Insurance Contracts, Retirement Contracts, Reinsurance Contracts Held and Reinsurance Contracts Issued.</p> <p>Tags may also go across multiple Domains, in line with entities that do so.</p> |
| <p>Details</p> | <p>Details refer to information including column description, logical name, and list of values. This information, as with Domains and Segments, is sourced from Data Foundation.</p> |

| | |
|--|---|
| | |
| Logical Entities | Logical Entities refer to the label attached to the data element for identifying similar data elements together across entities. For example, monetary data elements are available across all seven insurance product processors. A logical entity called Monetary Amount brings all premiums, commissions, expenses, the sum insured together under one heading thus expediting the search. |
| Search  | It is possible to search within the data elements using specific search criteria. See section Using Search Utility for details. |
| Filter  | It is possible to find out a particular entity or group of entities using specific filter criteria. See section Using Filter Utility for details. |
| Zoom In  | Helps to zoom in and view the details. |
| Zoom Out  | Helps to zoom out and view the details. |
| Show Logs  | Displays the log details in ascending order. |

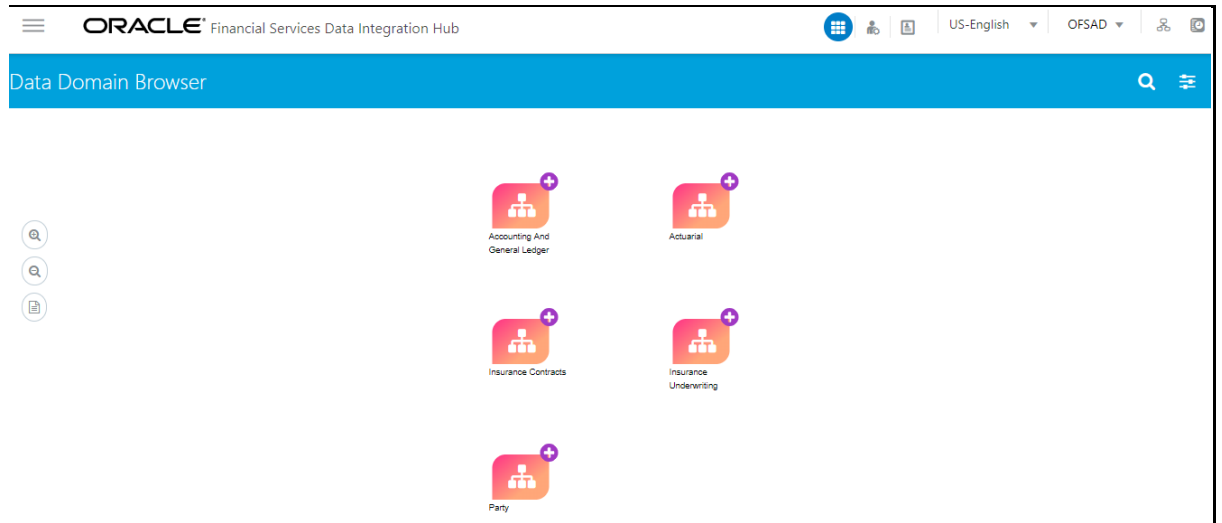
13.3.4 Navigating Data Domain Browser

DDB navigation helps to display a huge list of entities available in the Data Foundation in a pre-grouped format. It is possible to view the data elements within a given entity as well as the properties of each data element.

To navigate and use DDB, perform the following steps:

1. Click **Data Domain Browser** under the navigation list. This displays DDB components.

Figure 100: Data Domain Browser




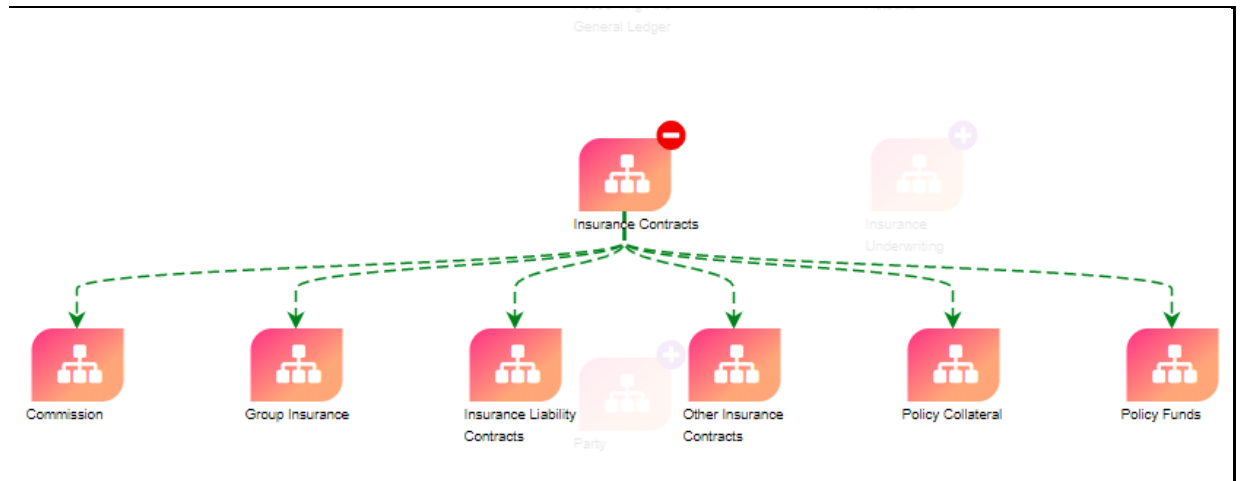
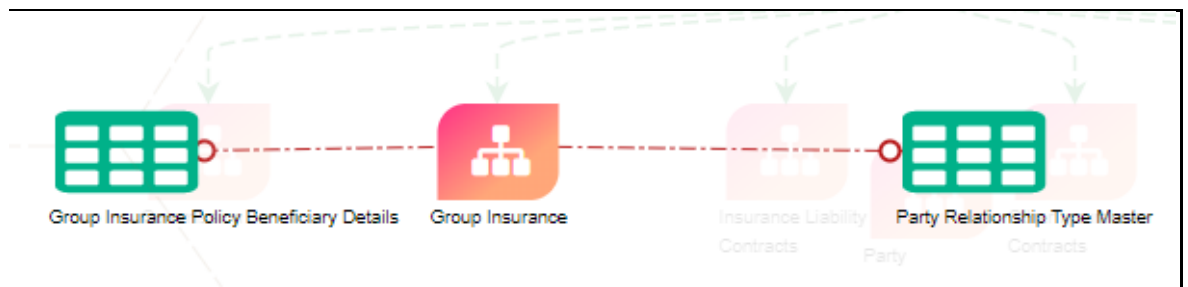
2. Click Filter  utility to view and filter the segments and then proceed. You can also search the segments to view the results. For details, see sections [Using Filter Utility](#) and [Segment](#) respectively.
3. Select the required **Data Domain**. This displays sub-data domains.

Figure 101: Data Domain Browser



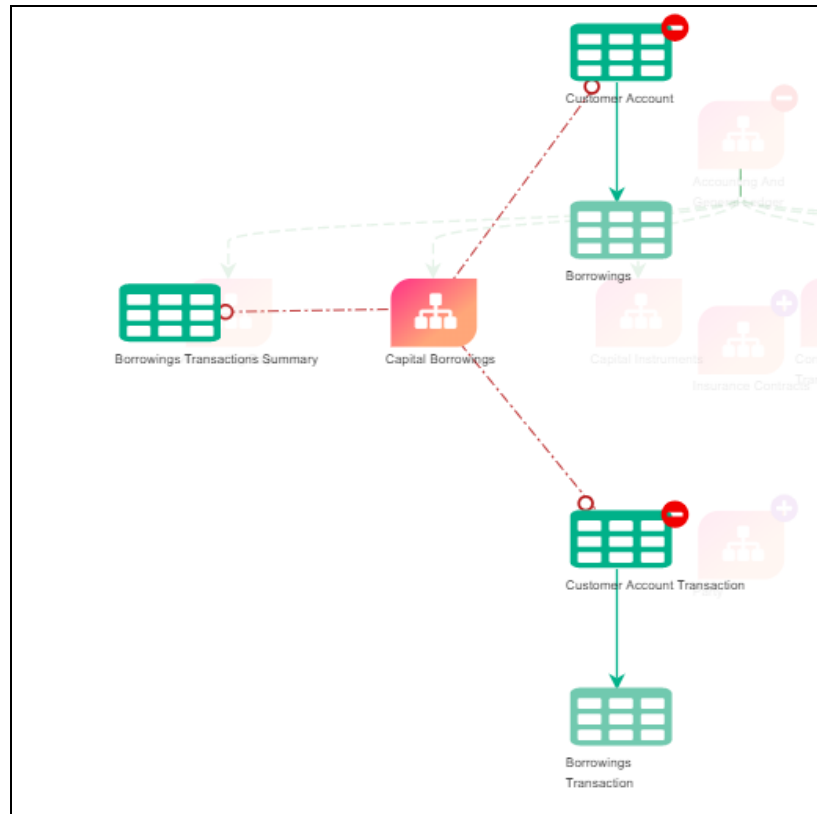
4. Click the required **Sub-Data Domain**. This displays the ADIs associated with it.

Figure 102: Sub-Data Domain



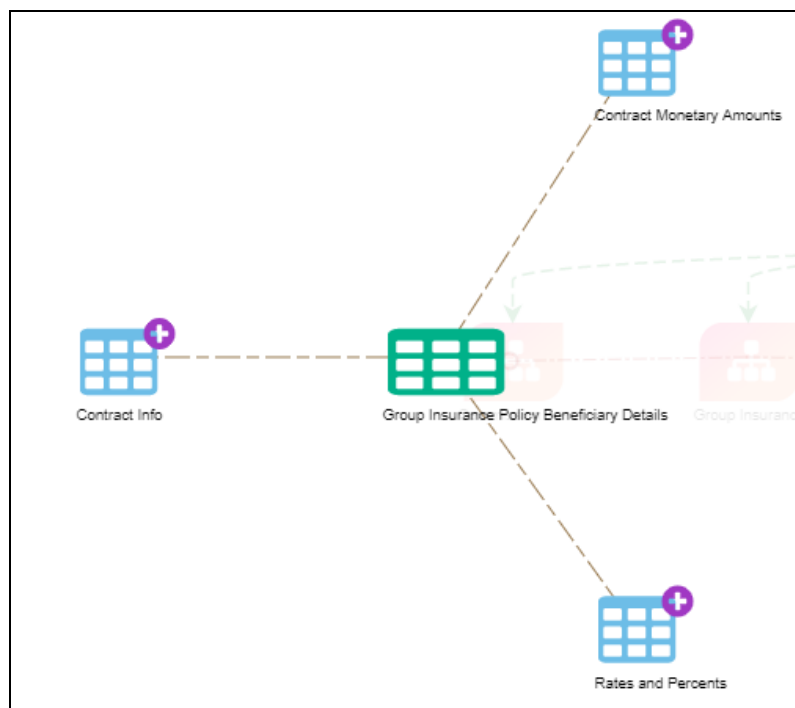
5. Click the required **ADI**.
 - a. If available, for some of the elements, it displays the **ADI sub-type** associated with it.

Figure 103: ADI sub-type



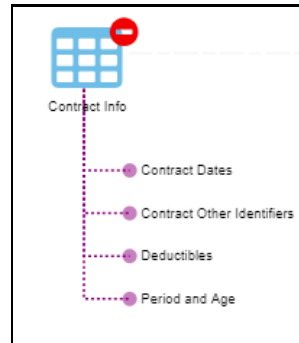
- a. For some of the elements, it displays the **Logical Entities** associated with it. **Logical Entities** refer to the label attached to the data element for identifying similar data elements together across entities.

Figure 104: Logical Entities



- b. Expand the **Logical Entities** which contains attribute groups or complex attributes.
- c. Click complex attributes. In the RHS attribute list is displayed. Click individual attribute to view the properties - Domain, description, LoVs.

Figure 105: Complex Attributes

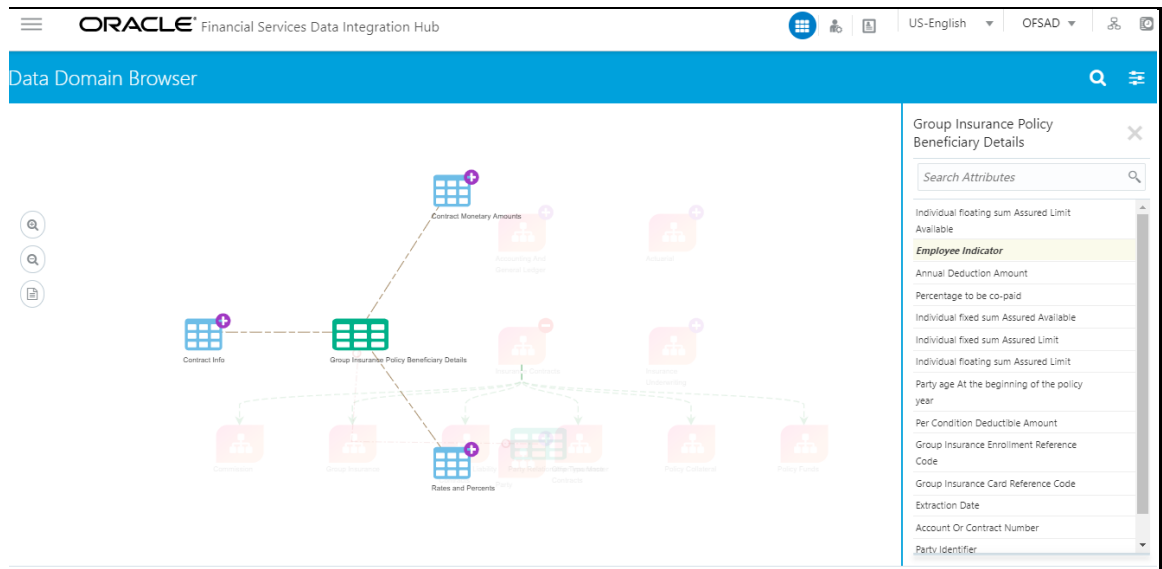


6. Click **ADIs** to view the names of the attributes on the left panel. The attributes displayed are either download or reporting entities depending on the selection.

Figure 106: ADI - Attributes

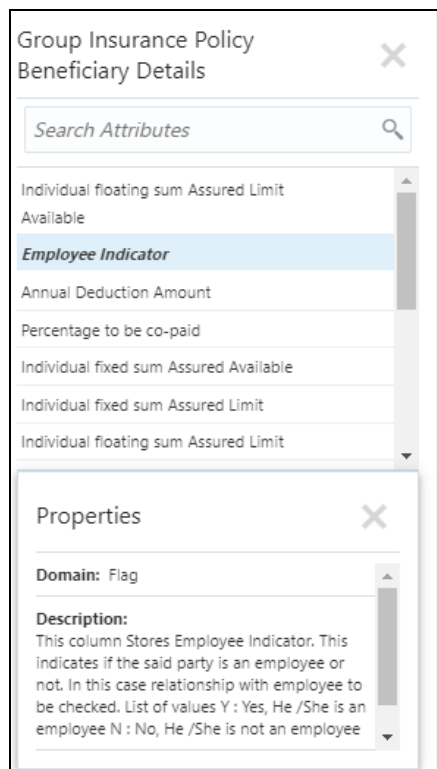
The screenshot shows the Oracle Financial Services Data Integration Hub interface. The top navigation bar includes the Oracle logo and 'Financial Services Data Integration Hub'. Below this is the 'Data Domain Browser' header. The main area displays a hierarchical tree of entities. The 'Borrowings' entity is selected, and its attributes are listed in a panel on the right. The attributes listed are: Current Payment Amount, Put Option Date, Base Value Of Index, Load Run Identifier, Market Value, Administered By US Agent Flag, Arms Length Transaction Indicator, Automatic Renewal Flag, Brokerage Account Indicator, Call Exercised Flag, Embedded Options Flag, Exposure Held At Fair Value Indicator, External Customer Branch Ibf Indicator, Fiduciary Relationship Flag, Guaranteed By Sovereign Indicator, and Held By Client Or Proprietary Flag.

Figure 107: ADI - Attributes



7. Click the attributes or column names to view their **Properties**. The Properties panel displays the **Domain Name**, **Attribute Description**, and **List of Values** sourced from the respective data model.

Figure 108: Attribute - Properties



13.3.4.1 Using Filter Utility

Using Business Segments and Domain Types, filter utility can be performed. This utility helps to find out a particular entity or group of entities using specific filter criteria.

NOTE

The segment window retains filters. Apply the filter in the Data Domain view itself and then proceed to view the sub-data domains and so on. In case you do not apply a filter in the Data Domain view and move from data domain to sub-data domain view, it displays all entities.

Perform the following steps to use the **Filter** utility:


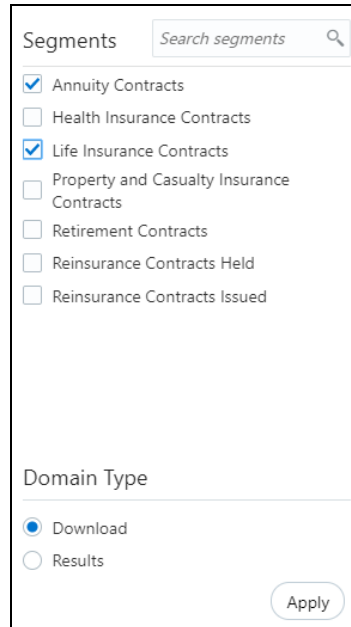
1. In DDB, click **Filter**  icon. The **Segments** and **Domain Type** are displayed on the right panel.

Figure 109: Segments and Domain Type



The screenshot shows a filter utility window. At the top, there is a 'Segments' section with a search bar labeled 'Search segments' and a magnifying glass icon. Below the search bar is a list of contract types, each with a checkbox:

- Annuity Contracts
- Health Insurance Contracts
- Life Insurance Contracts
- Property and Casualty Insurance Contracts
- Retirement Contracts
- Reinsurance Contracts Held
- Reinsurance Contracts Issued

 Below the list is a 'Domain Type' section with two radio buttons:

- Download
- Results

 At the bottom right of the window is an 'Apply' button.

2. Under **Segment**, select the required Segment to filter.
3. Under **Domain**, select the required Domain Type to filter. Here, it is either **Download** or **Results**.
4. In case you select Domain Type as Results, you can export the Dimension tables to an excel sheet and view the details. For more information, see the section [Exporting Tables](#).
5. Click **Apply**.

The following are the cases where filter utility is used:

- In case the filter is applied at parent subject area level, only those parent subject area is displayed which have at least one entity in its child Subject Area.
- Also, in case the filter is applied at the parent subject area level, on expansion only that child subject area is displayed which have at least one entity as per Business Segment filter.
- In case you do not apply the filter and proceed from the subject area window to the entity window, the filter shows all entities.
- Apply a filter in the subject area window, and then click any subject area to view the entities and then go to the next level. Here filter, which is applied to the subject area, is carried forward.

13.3.4.2 Using Search Utility

Data Domain Browser user interface helps to direct and expedite the search. It is possible to search within the data elements using specific search criteria.

For multiple entities displayed on the DDB window, a search facility within the entities displayed highlights the entities containing search criteria and removes other entities from the display. This helps to view the desired results easily.

13.3.4.2.1 Data Domain Search

To search for a particular functionality or a feature search in the Data Domain:

1. In DDB, search for specific data. For example, Party Contacts.

Figure 110: DDB - Search

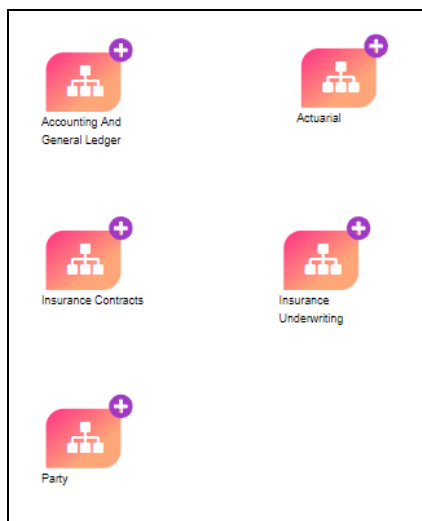
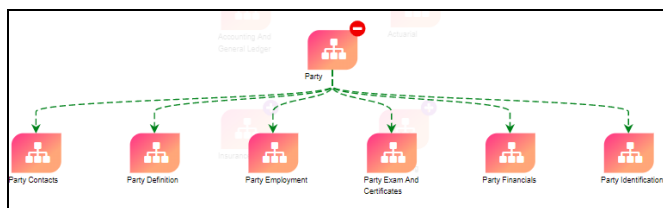
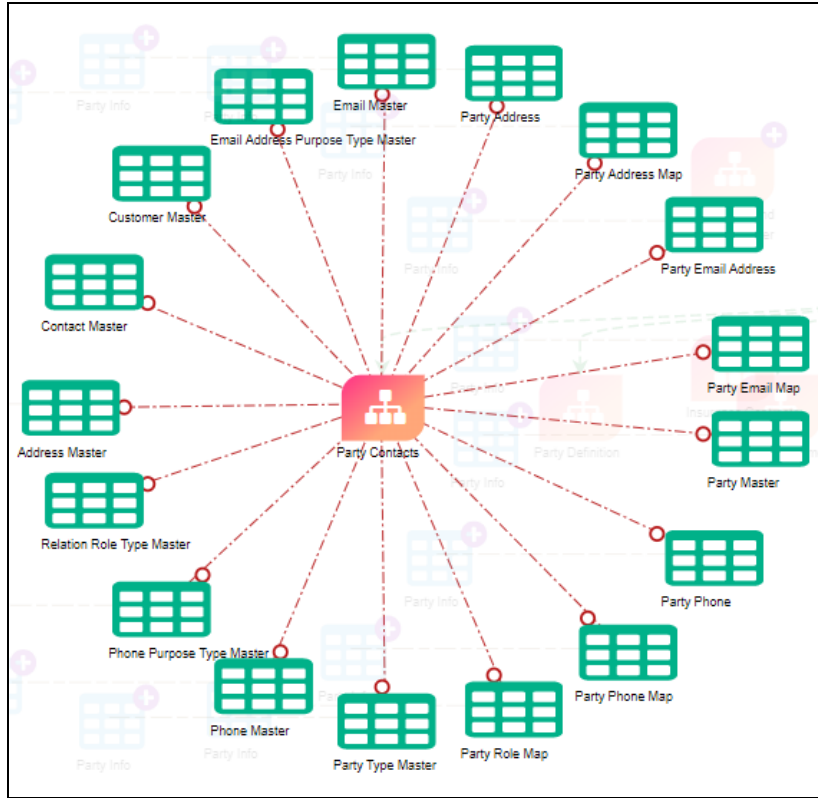


Figure 111: DDB - Search



2. Click Party Contacts in Sub Data Domain area screen. This displays the ADIs under the Sub Data Domain area.

Figure 112: Party Contacts




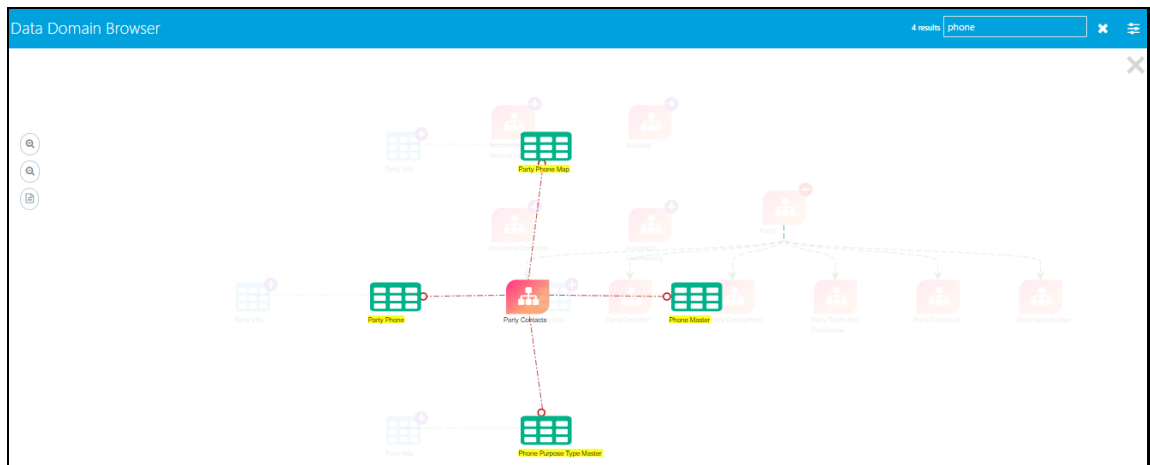
3. Click **Search** . The search option in the property window provides a toggle to use a direct logical column or tags to search the particular data element. The search results are highlighted in yellow and it filters and displays only those tables with the keyword “phone”.

Figure 113: Search Results in Yellow



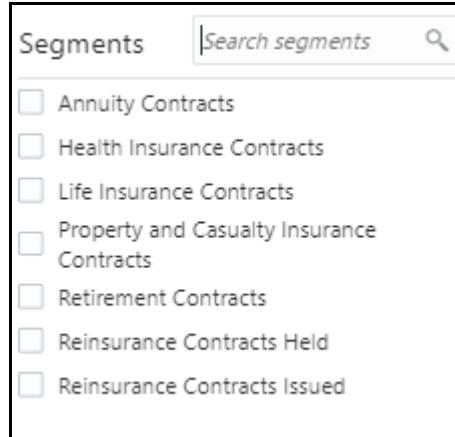
13.3.4.2.2 Segment Search

It is possible to filter entities based on segments for the domain type.

To search for commonality across different business units or departments you must search it in Business Segment.

1. In Segments, search for a specific segment. For example, Life Insurance Contracts.

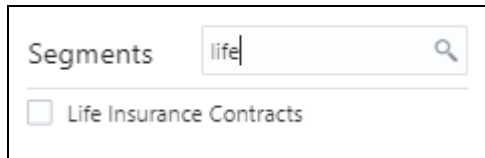
Figure 114: Search - Segments



The screenshot shows a search interface for segments. At the top, there is a search bar with the placeholder text "Search segments" and a magnifying glass icon. Below the search bar, there is a list of segments, each with a checkbox to its left:

- Annuity Contracts
- Health Insurance Contracts
- Life Insurance Contracts
- Property and Casualty Insurance Contracts
- Retirement Contracts
- Reinsurance Contracts Held
- Reinsurance Contracts Issued

Figure 115: Search - Segments

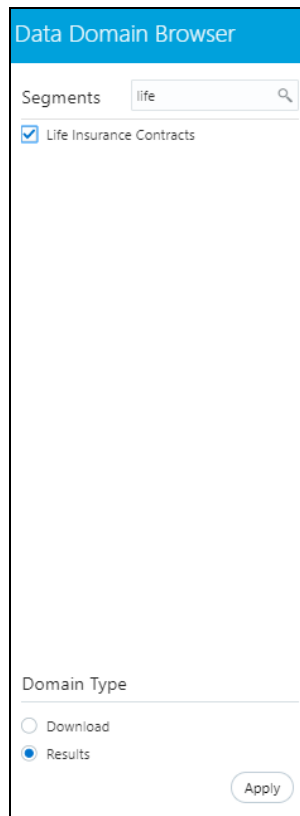


The screenshot shows the same search interface as Figure 114, but with the search term "life" entered in the search bar. The search bar now contains "life" and the magnifying glass icon. Below the search bar, the list of segments is filtered to show only one result:

- Life Insurance Contracts

2. Select Life Insurance Contracts in Segments and the Domain Type.

Figure 116: Segments and Domain Type



3. Click **Apply**. This filters only the Life Insurance Contracts' result ADIs. The commonality across different business units or departments is displayed.

13.3.4.2.3 Attribute Search

All the attribute information is available in the property window. The property window provides Domain Name, Description, and List of Values sourced from the respective data model.

The search pane in the property window provides a toggle to search the particular data element. You can enter the logical entity name in the search pane and view results.

1. In Attributes, search for a specific element. For example, Flag. This displays all the attributes with the name Flag.

Figure 117: Attribute - Search

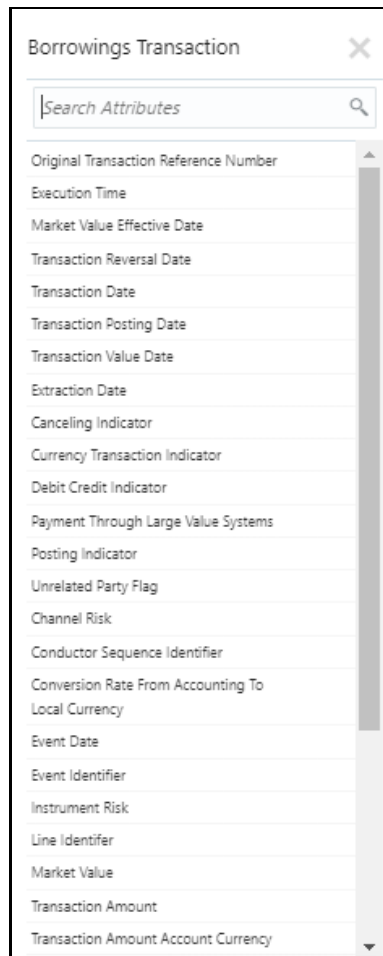
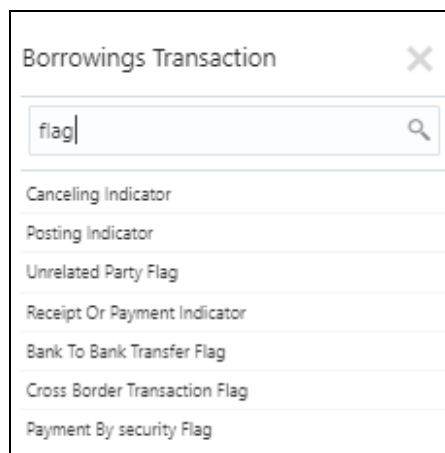
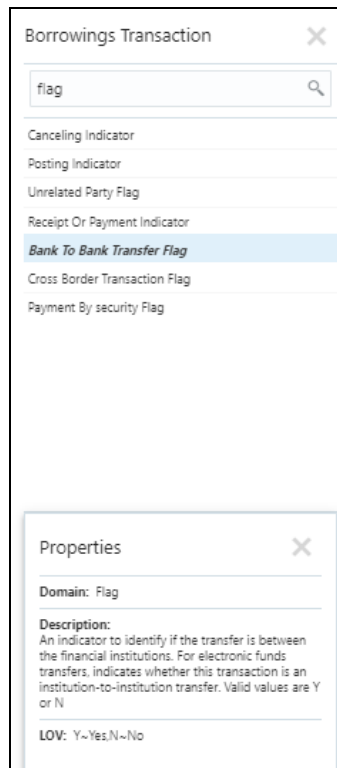


Figure 118: Attribute - Search



2. Click the individual attribute to view the property details.

Figure 119: Attribute - Properties



3. Click **Apply**. This filters only the Life Insurance Contracts' result tables. The commonality across different business units or departments is displayed.

13.3.4.3 Exporting Tables

It is possible to export dimension (result) tables. While we are viewing any dimensions as part of DDB, there is an option to show and hide the relationships of the specific dimension and export it to an excel sheet.


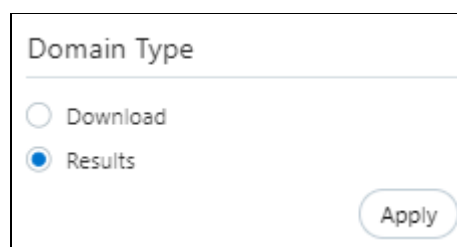
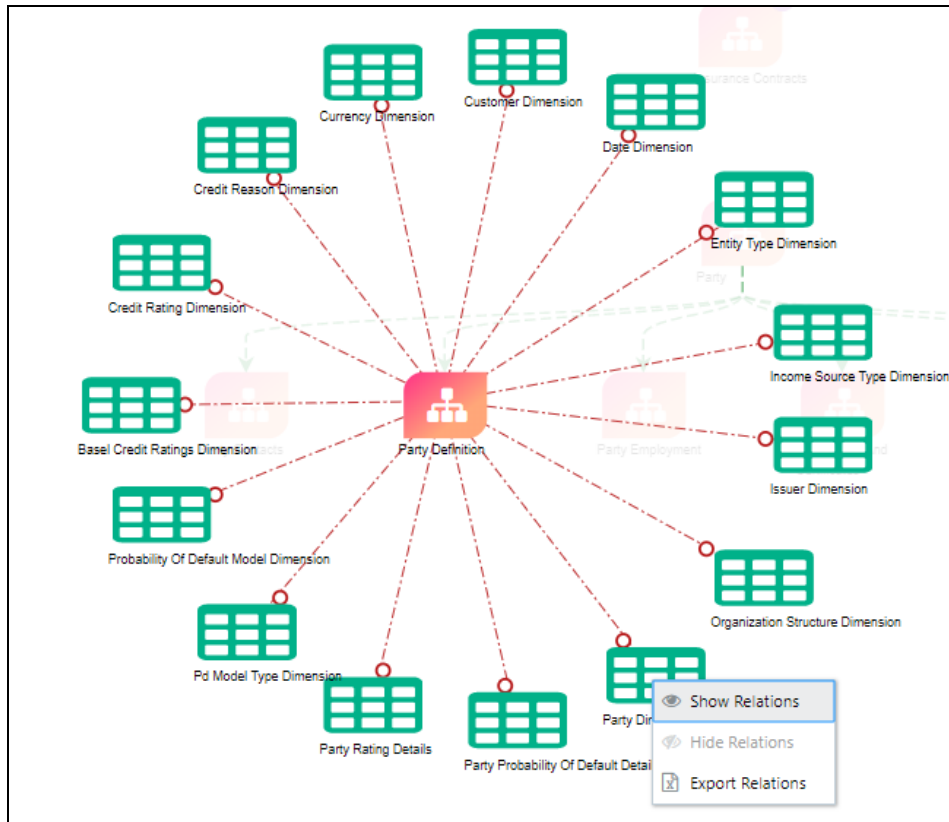
1. Click **Filter**  and select the Domain Type as **Results**.

Figure 120: Domain Type



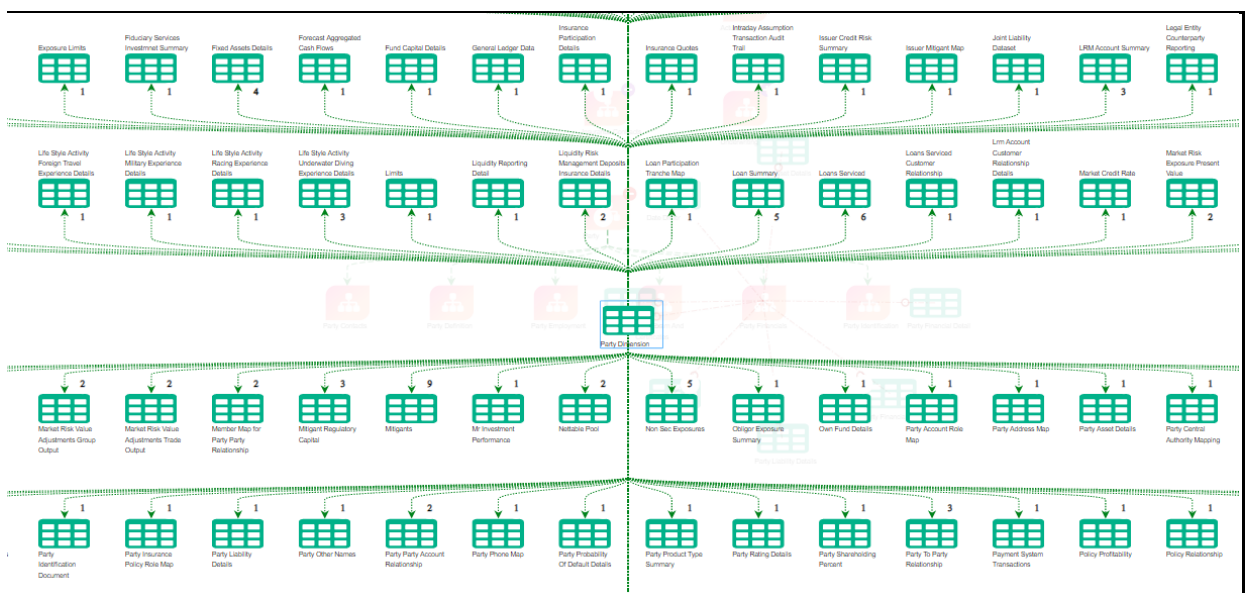
2. Navigate through data domains and sub-data domains to select a dimension table.

Figure 121: Party Definition



3. Right-click the dimension table and select **Show Relations**. This displays the relationships of each entity dimension.

Figure 122: Party Definition - Show Relations



- Right-click the dimension table and select **Export Relations**. This downloads the table details to an excel format.

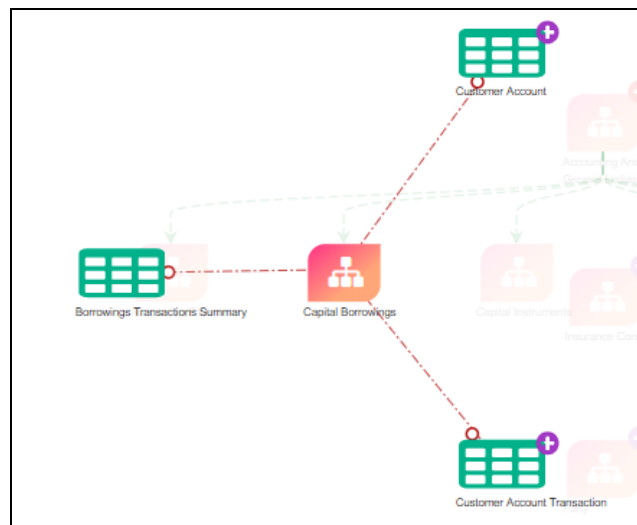
13.3.5 Use Cases

The following are some of the DDB Use Cases with examples.

1. Some ADIs have subtypes

The data domain and the sub-data domain can be filtered depending on the available segments. Expand the sub-data domain, the list of entities around it is displayed.

Figure 123: Data Domain and Sub-data Domain



Some entities have subtypes. Click the subtypes to view the attributes and properties.

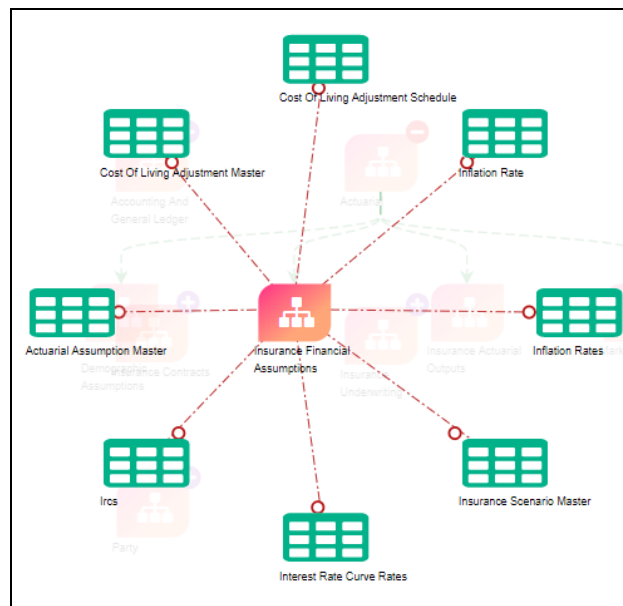
For example, click Customer Account and Customer Account Transactions in the above example:

Figure 124: Subtypes - Attributes and Properties

2. Subtypes have logical entities

The data domain and the sub-data domain can be filtered depending on the available segments. Expand the sub-data domain, the list of entities around it is displayed.

Figure 125: Sub-data Domain



Some entities have logical entities. Click the subtypes to view the logical entity details under properties.

For example, click the Inflation Rate in the above example:

Figure 126: Subtypes - Logical Entities

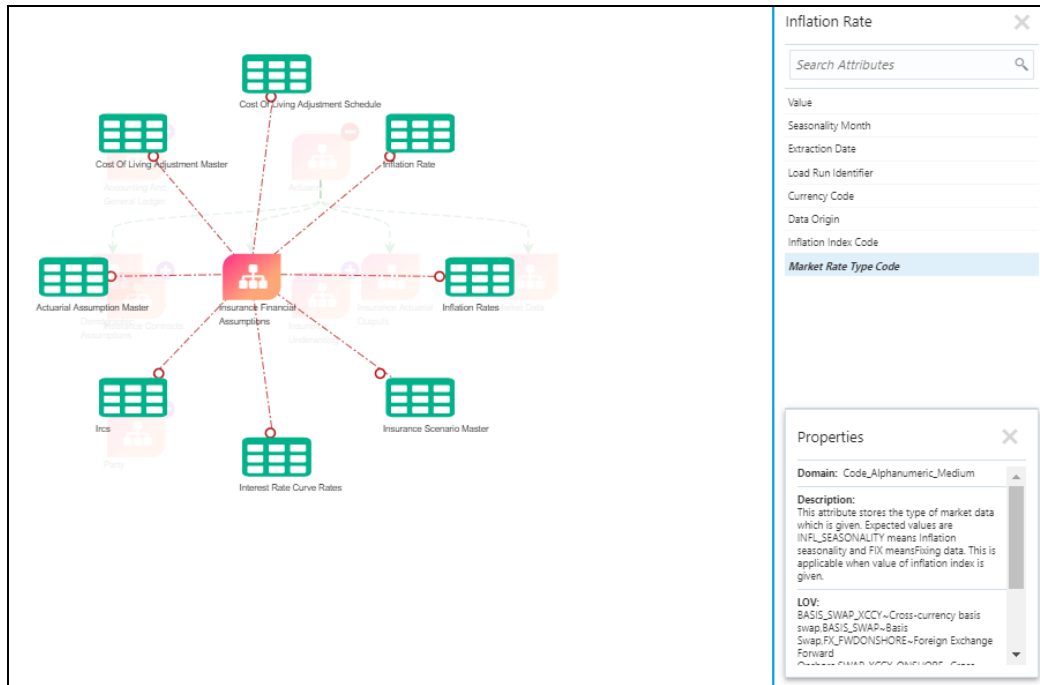
This figure shows the 'Inflation Rate' entity selected in the diagram. A 'Properties' panel is open on the right side, displaying the following attributes:

- Value
- Seasonality Month
- Extraction Date
- Load Run Identifier
- Currency Code
- Data Origin
- Inflation Index Code
- Market Rate Type Code

The panel also includes a search bar labeled 'Search Attributes' and a close button (X).

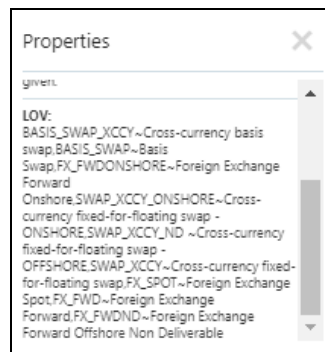
Click the attributes to view the logical entities. For example, click the Market Risk Type Code.

Figure 127: Logical Entities



The property window displays the Domain Name, Description, and List of Values or logical entity information.

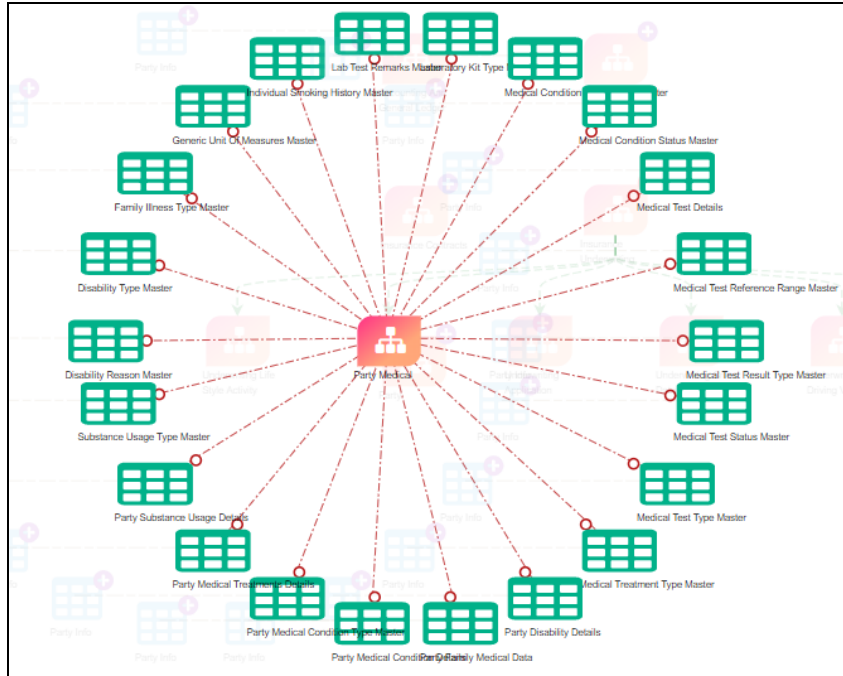
Figure 128: Properties



3. Some logical entities have a group or complex attributes

The data domain and the sub-data domain can be filtered depending on the available segments. Expand the sub-data domain, the list of entities around it is displayed.

Figure 129: Logical Entities



Some subtypes have logical entities.

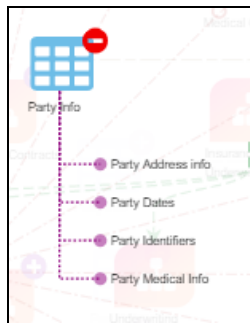
For example, click Medical Test Details in the above example. Only Medical Test Details are displayed in a new window and all other tables are hidden.

Figure 130: Logical Entities



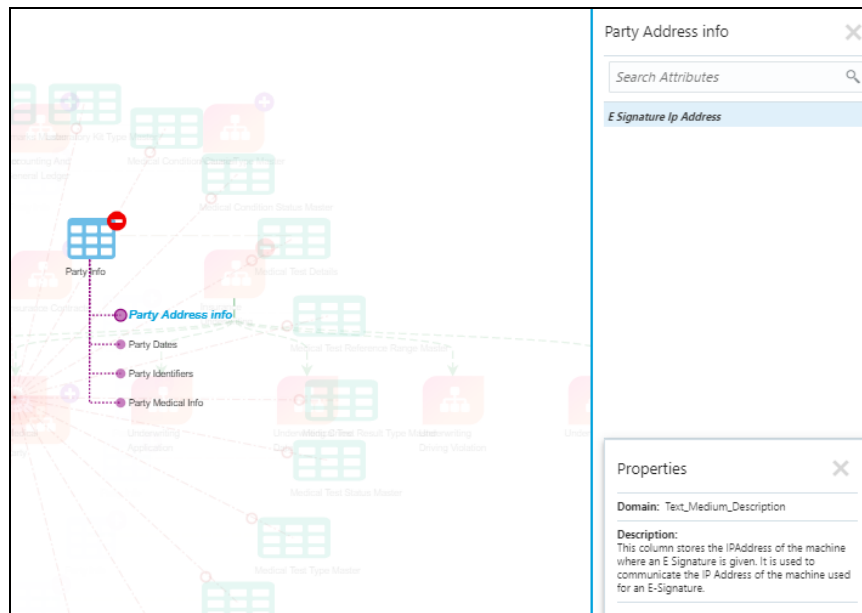
Subtypes are expanded to smaller logical entities. When you expand the logical entities, a group or complex attribute list is displayed.

Figure 131: Complex Attributes



4. Click the attributes to view the property details. Each complex attributes have a list of attributes.

Figure 132: Complex Attributes – List of Attributes



5. Identifying fact tables where the dimension is referred

When a result dimension is selected under all fact entities, the associated entities are displayed in the DDB application. The relationship between a Dimension Entity and Fact Entity is in line with Foreign Key relationship display standards. When relationships are displayed in the DDB application, all other entities displayed before the relationship display are minimized. It is possible to navigate back to the most recent or earliest user interface screen from Relationship Display.

6. Exporting a list of fact tables

It is possible to export dimension tables. While we are viewing any dimensions as part of DDB, there is an option to show and hide the relationships of the specific dimension and export it to an excel sheet.

The columns display result dimension name, result dimension surrogate key, fact entity, and fact entity surrogate key.

For details, see section [Exporting Tables](#).

7. Identifying Similar Logical Entities

You can use search utility in the data domain, segments, and attributes to find similar logical entities. For details, see [Using Search Utility](#).

14 Appendix A: Frequently Asked Questions (FAQs)

This appendix provides answers to frequently asked questions.

Topics:

- [General FAQs](#)
- [Upgrade FAQs](#)

14.1 General FAQs

Loading Data from One File into Staging

To load data from one file to Staging, follow these steps:

1. Create an EDS of the type **File**.
2. Create EDD by selecting pre-defined EDS. Provide all required information while creating the EDD. If post-loading reconciliation is required, then go to the Control tab and provide a control record.
3. Create a Connector for loading data into staging. Select both the EDDs, establish a join, and click Lookup. If the SQLLDR option is enabled, then the file must be available in the server where the ODI agent is running. If the External Table option is selected, then the file must be available in the target database server.
4. Publish the Connector.
5. Execute the Connector.

Loading Data from Two Files Joined into Staging

To load data from two files that are joined to Staging, follow these steps:

1. Create EDS of type **File**.
2. Create two EDD by selecting pre-defined EDS. Provide all required information while creating EDD. If post-loading reconciliation is required, then go to the Control tab and provide a control record.
3. Create a Connector for loading data into staging. Select both the EDDs, establish a join, and click Lookup. If the SQLLDR option is enabled, then the file must be available in the server where the ODI agent is running. If the External Table option is selected, then the file must be available in the target database server.
4. Publish the Connector.
5. Execute the Connector.

Loading Data into Staging from File and Performing Lookup into a Table

To load data from a file to Staging, and perform a lookup into a table, follow these steps:

1. Create one EDS of type **File** and another EDS of type **Database**.

2. Create two EDD by selecting predefined EDS. Provide all required information while creating the EDD. If post-loading reconciliation is required, then go to the Control tab and provide a control record. Post loading reconciliation is only applicable for file type Data Loading.
3. Create a Connector for loading data into staging. Select both the EDDs, establish a join, and click Lookup. If the SQLLDR option is enabled, then the file must be available in the server where the ODI agent is running. If the External Table option is selected, then the file must be available in the target database server.
4. Publish the Connector.
5. Execute the Connector.

Loading Data into Staging from an XML File

To load data to Staging from an XML file, follow these steps:

1. Create an EDS of type **XML**.
2. Create EDD by selecting predefined EDS. Provide all required information while creating the EDD. To get the XML structure, use the XSD file. If post-loading reconciliation is required, then go to the Control tab and provide a control record. The control record must be in a separate file.
3. Create a Connector for loading data into staging. Select both the EDDs, establish a join, and click Lookup. If the SQLLDR option is enabled, then the file must be available in the server where the ODI agent is running. If the External Table option is selected, then the file must be available in the target database server.
4. Publish the Connector.
5. Execute the Connector.

Loading Data into Staging from EBCDIC

To load data to Staging from EBCDIC, follow these steps:

1. Create an EDS of type EBCDIC.
2. Create EDD by selecting predefined EDS. Provide all required information while creating the EDD. To define the EBCDIC structure, use the Cobol Copybook file. If post-loading reconciliation is required, then go to the Control tab and provide a control record. The control record must be in a separate file.
3. Create a Connector for loading data into staging.
4. Publish the Connector.
5. Execute the Connector.

Loading Data from File into Multiple ADI/Subtypes

To load data from the File into multiple ADI/Subtypes, follow these steps:

1. Create an EDS of type **File**.

2. Create EDD by selecting predefined EDS. Provide all required information while creating the EDD. To define the file structure, you can use an Excel template. If post-loading reconciliation is required, then go to the Control tab and provide a control record.
3. Create a Connector for loading data into staging. Select multiple ADIs/Subtypes. Set filter against each ADI/Subtype selected to identify record status.
4. Publish the Connector.
5. Execute the Connector.

Loading Data into Results from File

To load data into Results from File, follow these steps:

1. Create an EDS of type **File**.
2. Create EDD by selecting predefined EDS. Provide all required information while creating the EDD. To define the File structure, you can use the Excel template for defining the columns. If post-loading reconciliation is required, then go to the Control tab and provide a control record. The control record must be in a separate file.
3. Create a Connector for loading data into results. Select EDD and ADI (is displayed only relevant for the Result area). During mapping, code values that are available in the file must be mapped to the attributes (that are internally number columns). While loading it converts the code value to the surrogate key by performing a lookup in the related dimension table. Dimension data must be populated before this loading occurs using the SCD process.
4. Publish the Connector.
5. Execute the Connector.

Loading Data from HIVE

The following Jars must be copied to /ficweb/WEB-INF/lib:

1. commons-logging-1.1.3.jar
2. guava-15.0.jar
3. hadoop-auth-2.6.0-cdh5.8.0.jar
4. hadoop-auth.jar
5. hadoop-common-2.6.0-cdh5.8.0.jar
6. hadoop-common.jar
7. hive-common-1.1.0-cdh5.8.0.jar
8. hive-jdbc-1.1.0-cdh5.8.0.jar
9. hive-metastore-1.1.0-cdh5.8.0.jar
10. hive-service-1.1.0-cdh5.8.0.jar
11. HiveJDBC4.jar
12. httpclient-4.3.jar

13. httpcore-4.3.jar
14. libthrift-0.9.0.jar
15. slf4j-api-1.7.5.jar

For example, to connect to the Cloudera Hive server with JDBC 4.0 data standards, specify “com.cloudera.hive.jdbc4.HS2Driver” as the driver. See the [Cloudera](#) document For more information about Cloudera JDBC drivers.

NOTE Contact Oracle Support for additional queries.

Loading data into Partitioned Hive Table

If data is loaded into the partitioned table in Hive, the following configuration must be done in ODI by logging into ODI studio:

1. Navigate to **Designer Tab > Models > <INFODOM_NAME>**.
2. Right-click the target table to be loaded and click Open.
3. Navigate to Attributes Tab.

Select the “Partition by” checkbox for the column on which partition is defined.

Extracting Data from Staging into File

To extract data from Staging into File, follow these steps:

1. Create an EDS of type **File**.
2. Create EDD by selecting predefined EDS. Provide all required information while creating the EDD. To define the File structure, you can use the Excel template for defining the columns. If post-loading reconciliation is required, then go to the Control tab and provide a control record. The control record must be in a separate file.
3. Create a Connector for extracting data from staging. ADI becomes the source and EDD becomes a target. The file structure is according to EDD.
4. Publish the Connector.
5. Execute the Connector.

Extracting Data from Result into File

To extract data from Results into File, follow these steps:

1. Create an EDS of type **File**.
2. Create EDD by selecting the predefined external data store. Provide all required information while creating the EDD. To define the File structure, you can use the Excel template for defining the columns. If post-loading reconciliation is required, then go to the Control tab and provide a control record. The control record must be in a separate file.

3. Create a Connector for extracting data from results. ADI becomes the source and EDD becomes a target. The file structure is according to EDD. During extract, internal surrogate keys are converted into code values by performing a lookup into the dimension table.
4. Publish the Connector.
5. Execute the Connector.

OFSAA out of the box dimension tables as lookup entity

Out of the box, dimension tables are used in the following scenarios:

- Loading Data into results
- Extracting data from results

While defining mapping, if the results ADI surrogate key column is mapped, then the corresponding dimension table is used as a lookup.

For Example:

- Loading data into Results

If N_Product_Skey of Common Account Summary ADI is mapped to an EDD product code column, then DIM_PRODUCT Table is used as a lookup for getting the N_Product_Skey value and V_PROD_CODE is used in the join expression. The EDD does a left outer join with the DIM_PRODUCT.

- Extracting Data from Results

If N_Product_Skey of Common Account Summary ADI is mapped to an EDD product code column, then DIM_Product Table is used as a lookup for getting the V_PROD_CODE value and N_Product_Skey is used in the join expression. The Common Account Summary lookup does a left outer join with DIM_PRODUCT.

NOTE

The lookup is established by identifying the Surrogate key relationship between Results ADI and Dimension table automatically. You do not need to define the lookup in the connector.

How to connect to External Source Server such as Db2/SQL Server/Teradata/Sybase?

DIH uses the JDBC mechanism to connect to source such sources. Hence the JDBC driver for the source server must be made available in the deployed location for DIH and ODI agents.

Navigate to \$FIC_WEB_HOME/webroot/WEB-INF/lib and copy the JDBC driver.

14.2 Upgrade FAQs

Is it necessary to reconfigure existing Connectors and EDDs after upgrading DIH to 8.1.0.0.0?

No. The upgrade process of the application remodels the existing Connectors and EDDs to the latest version, automatically. Also, the enhanced ADI refresh process takes care of reconfiguring existing connectors as per the latest data model changes, if any. However, the 8.1.0.0.0 version comes with

strict encryption policies. Hence, you need to follow these steps to ensure all credentials are protected with the latest encryption.

1. Specify ODI Passwords and Save.
2. Trigger ADI Refresh.
3. Save EDSs after specifying the password.
4. Trigger Target ADI Refresh.

Publishing the existing connector fails with “javax.crypto.BadPaddingException” error in the DIH.log, after upgrading to 8.1.0.0.0. How can this be resolved?

Follow the steps mentioned in the FAQ “Is it necessary to reconfigure existing Connectors and EDDs after upgrading DIH to 8.1.0.0.0?”

Can the new features in the existing connectors be used?

Yes, you can use the new features, added in 8.1.0.0.0, for existing connectors.

There are Data model changes in the 8.1.0.0.0 version of other applications. Is it required to reconfigure existing connectors since there are changes in ADIs?

No. ADI refreshes abstracts model changes from mapping. See section Abstraction of Model Changes for Data Movement / ETL Processing and Handling Model Changes with Impact on Data Movement / ETL Processing.

A message appears "Migrate <CONNECTOR_NAME> to Dataflow View" while accessing the existing connectors' built-in Standard View. How can this be resolved?

The Standard View mode in Connector Definition is de-supported from the 8.0.7.0.0 release onwards.

However; the application provides an option to migrate the connectors to the Dataflow View as a one-time activity. Perform the following steps to migrate:

1. Login to the DIH application with a valid User ID and Password.
2. Launch a new tab in the browser and enter the URL:

Information about the total number of connectors in Standard View and the total number of connectors that are migrated successfully to the Dataflow View are displayed as follows.



All the existing connectors in Standard View are now available in the Dataflow View.

NOTE Contact Oracle Support for additional queries.

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.

