Oracle® Financial Services Data Foundation Cloud Service for Banking

Data Services Guide

Release 25C
G30800-13
October 2025

Oracle Financial Services Data Foundation Cloud Service for Banking Data Services Guide, Release 25C

G30800-13

Contents

4 Application Data Services (DFCS Only)

5 Integration with Oracle Applications

6 Data Extraction

7 Dimension Loader

8 Data Pipelines

9 Key Terms and FAQs

1 Data Services

Data Services facilitates seamless data exchange between OFSAA and external systems by providing a logical abstraction of the OFSAA Data Foundation—which includes the Financial Services Data Foundation —through Application Data Interfaces (ADIs). Using the DIH (Data Integration Hub) user interface, users can define External Data Sources (EDS) and External Data Descriptors (EDD). These EDDs can then be mapped to ADIs to create Connectors, enabling structured and efficient data flow across systems.

2 Data Source Templates

The **Data Source Templates** section in the **Data Services** module of the **Data Foundation for Banking** application is designed to manage data exchange between the **Financial Services Data Foundation Cloud Service for Banking** and external systems. This is accomplished through a logical abstraction layer that exposes the Data Catalog as **Application Data Interfaces (ADIs)**.

Users can define **External Data Descriptors (EDDs)** via the interface, and then map these EDDs to ADIs to create **Connectors** that enable smooth and efficient data services Additionally, the interface provides a summary of all existing templates, displaying key details such as the creator and the date each template was created.

To navigate to the Data Source Template, follow the below steps.

1. From the **Data Foundation for Banking** window, click **Data Services** > **Data Source Templates**.
   The Summary page displays the list of External Data Descriptor (EDD).

2.1 External Data Descriptor (EDD)

External Data Descriptors (EDD) are definitions of specific data content from External Data Sources (EDS). **EDD** helps define how data from **External Data Sources (EDS)** should be structured and processed within the **DFCS**.

- It enables:
  - **Ingestion** of external data into DFCS.
    Extraction of data from DFCS.
  - **Control information** management during data ingestion.

Each EDS can have **multiple EDDs**, each tailored to a specific data format or control logic.

1. On the **Financial Services Data Foundation Service for Banking** home page, click **External Data Descriptors**.

   The **Summary** page is displayed.
2. Click **Search** to search the existing Data Descriptor from the available list. Sort by clicking the **Name** drop down list.
   - Name
   - Last Modified Date
3. Click **Add** to configure additional EDDs.
4. Enter the **Specify Data Descriptor name** and **Describe the Data Descriptor** details.
5. **Configure File Settings** (under the *Data* tab):

Table 2-1 Data Tab Description

Fields	Description
Toggle **Is the file archived?**	By enabling this option, you can upload an archived data file to the object store as input to the DFCS data ingestion process. You can also combine multiple files into an archive and use it to load data into DFCS. **Note** The Archive file name field appears only when **Is the file archived** option is enabled. Enter the archive file name. Example: td_contracts%#MISDATE%.zip Only archives with a .zip extension using standard DEFLATE algorithm are supported.
Specify data file names	Click on **Add More Files**. Enter the file name and click **OK**. You can add multiple data files to an EDD. For example, if 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 td_contracts%#MISDATE%_1.csv, and as the next record, add the Term Deposits Contracts data file for corporate accounts.In case the file sizes are large, it is recommended you break the file into smaller files. Optimize file size is 3 GB for each file. Example: td_contracts%#MISDATE%_1.csv
Select File Format	There are two options: • 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 adds 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. For example, select Delimited.
Select Record delimiter	The records are stored differently in different operating systems. The available options are: • MS-DOS • Unix • No Record Delimiter • Other For example, select Unix.

Table 2-1 (Cont.) Data Tab Description

Fields	Description
Specify Text Qualifier (Optional)	A character that identifies text. Generally, double quotes are prefixed and suffixed to identify text. This is optional.
Specify the number of records to skip	Provide the number of records to be skipped. The records are skipped from the top. Generally, this is used to skip headers.
Specify Decimal Separator	The character used to identify the decimal and fractional part. Usually point (.) or comma (,)
Do you want to read from template?	(Optional) Enable **Do you want to read from template?** to auto-populate values in the Excel file format. If the template is not available, create it manually by clicking Add, under the Data Elements. 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. **Note** CSV files can support UTF-8 encoding, which allows for special characters and non-Latin text. To ensure these characters are preserved correctly, make sure to save the file with UTF-8 encoding.
Data Elements	If not using a template, manually **add data elements** using the **Add** option.
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. Example: String, Number, and Date.
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 refers to the **length only**.
Scale	This is used to specify the number of digits after the decimal point. Example: 10.3.

Table 2-1 (Cont.) Data Tab Description

Fields	Description
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** • 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. **Steps to Configure Control Settings in EDD**: Reconcile between source data received and data loaded into Financial Services Data Foundation Cloud Service for Banking, 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. The execution fails only if the difference is more than the threshold value.
7. The threshold can be an absolute or a percentage value. The connector execution process considers the threshold type while performing reconciliation.
8. Generate control information for file extracts from DFCS. 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).

Table 2-2 Control Tab Description

Field	Description
NOTE: The Reconciliation details present in separate file option cannot be modified if the option **Is the file archived** in **Data Tab** is enabled. When Archive option is enabled: If you want to perform reconciliation, control file must always be present in the same archive as input data files and should be a separate file. The control file should have only two columns that are Control Name and Expected Value.	
When **Separate File** is selected as **Yes**.	
File Name	Specify the name of the file.
File Format	There are two options: • **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. In the previous example, select Delimited.
Column Delimiter	If the File Format is selected as Fixed Length, the Column Delimiter would be Other by default. If the File format is selected as Delimited, the following options are available in the drop-down list. • Other • Space • Semicolon • Comma • Tab In the previous example, select Comma.
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 such a 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: • MS-DOS • Unix • No Record Delimiter • Other For example, select Unix.

Table 2-2 (Cont.) Control Tab Description

Field	Description
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. Example: If you specify this as 1, the first row in the file will be ignored.
Text Qualifier	A character that identifies text. Generally, double quotes are prefixed and suffixed to identify 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 Records 400” 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”. Therefore, 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 **Are Reconciliation details present in Separate File** option is disabled, the below fields will be enabled.	
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 such a 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	
Control Name	Specify the name of the control.
Aggregation Method	Select either Aggregation Method or Count. The supported aggregation methods are as follows: • Min • Max • Average • Sum

Table 2-2 (Cont.) Control Tab Description

Field	Description
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 available options for threshold: percentage or absolute. If you select percentage, the reconciliation difference in percent is matched against this threshold value. If you select absolute, the absolute percent difference is matched against this threshold value.
Threshold Value	Specify the difference value in percent or absolute.

> **Note**
>
> For more information on Data Load Dashboard, see Data Load Dashboard section.

9. Click on the **Transformation** tab at the top of the EDD configuration screen.

Table 2-3 Transformation Tab Description

Fields	Description
Name	Name of the derived field in the 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.
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**. Then the newly created field name is listed.
If the Transformation Type is selected as **Aggregation**:	
Derived Data Elements	

Table 2-3 (Cont.) Transformation Tab Description

Fields	Description
Name	Name of the derived field in the 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**. 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.

10. **Save the Descriptor**: Click **Save as Draft** to store your progress or **Save** to finalize the descriptor once all required fields are completed.

2.2 Defining an External Data Descriptor (EDD)

To define a new **External Data Descriptor (EDD)**, follow these steps:

1. Navigate to the **Data Foundation Cloud Services** home page, and then click **EDD**.

   The **Data Ingestion Summary** page is displayed.
2. On the **Data Ingestion Summary** page, click **Add**.

   This will open the **New** tab.
3. Enter the **name** and **description** for the data descriptor, and then select **Datastore** from the drop-down list. Select the data source (example: **FOUNDATION_DATA_STORE**). The values in Defining an External Data Store example are used. The description will be automatically populated.
4. Click **Save as Draft** or click **Save**.
5. Enter the required values in the fields.
6. If data needs to be reconciled post-loading, click the **Control** tab.
7. To transform the EDD, click the **Transformation** tab.
8. You can add derivation to data elements of the EDD.
   - Click **Add** to create derived data elements.
   - To edit the derived data elements, click **Edit**. This will open the **Expression** window, where you can specify the expression using data elements defined in the **Data** tab and functions.
   - To delete the derived data element, click **Delete**.

9. Click the **Transformation** tab and select the **Transformation Type**.
    - Select **Aggregation** and click **Edit** to view **Expression** window.
    - In the expression window, specify the **Group by** clause and **Having** expression, if applicable.
    - Define Derived Data Elements for the field to be aggregated under the previous tab.
10. Click **Save**.

2.3 Modifying and Viewing an External Data Descriptor (EDD)

You can edit or view existing EDDs.

> **Note**
>
> You cannot edit EDDs in **Published** status.

To edit or view an EDD, complete the following steps:

1. From the EDD Summary page, select the required EDD.
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**.

2.4 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 an EDD is used in a Connector. Then, unless the Connector is deleted, the used EDD cannot be deleted.

To delete an existing EDD, complete the following steps:

1. On the EDD Summary, click **Delete** corresponding to the EDD you want to delete.
2. Acknowledge the confirmation message.

    The EDD details are deleted.

2.4.1 Search and Filter

Use the search and filter options to find the required information. You can enter keywords nearest to what you are looking for in the search box. You can search for an EDD using either the name, description, status, or type.

For example, enter the EDD keyword as Loan Data in the search box. The search results show the list of all EDDs containing the text **Loan Data**.

2.4.2 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: Account number and Balance.

Table 2-4 EDD Parameters Example

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.

Parameter 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.

3

Ingest Connectors

Ingest Connectors serve as configurable components that enable the mapping of one or more **External Data Descriptors (EDDs)** to an **Application Data Interface (ADI)**. They are designed to support both inbound and outbound data exchange, ensuring seamless integration between external systems and the **Financial Services Data Foundation**.

1. Navigate to **Home** page, under **Data Services** > **Ingest Connectors**.

3.1 Icons and Description

This section describes the tools Financial Services Data Foundation Cloud Service for Banking provides to create connectors.

Table 3-1 Icons and Description

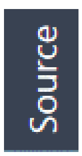
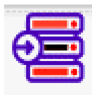
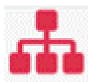
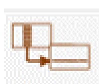
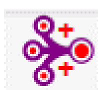
Icon	Description
	Click the Source button to view the list of all External Data Descriptors created in the setup. Use the **External Data Store** field to filter and the search icon to quickly locate the EDD you are looking for. After locating the EDD, you can drag the desired EDD on to the canvas.
	Click the Target button to view the list of all Application Data Interfaces (ADIs). Use the **Applications** field to filter and the search icon to quickly locate the ADI you are looking for.
	The Mapping component is used to map the source columns to target columns.
	The Join component is used to define a join between two entities. Double click this icon and provide the join condition information.
	The Filter component is used to define the filter of a given entity. Double click this icon and provide the filter expression information.
	The Lookup component is used to define the lookup condition. Double click this icon and provide the lookup information.

Table 3-1 (Cont.) Icons and Description

Icon	Description
	The Expression component is used to define the derived columns. Double click this icon to define an expression, which can be mapped to the target column.
	The Flattened Table To Hierarchy component is used to transform flattened hierarchy entities into parent-child hierarchy entities.
	The Transpose (Row to Column) component is used to transpose rows to columns for a given entity. Double click this icon to define the pivot data element and the new columns, which are transposed from multiple rows of the source entity.
	The Transpose (Column to row) component is used to transpose columns to rows for a given entity. Double click this icon to define the unpivot data element and new rows which are transposed from the columns of the source entity.
	The Aggregation component is used to define a group by and having a clause for aggregation. Double click this icon to define a group by and having a clause for aggregation.
	Click this button to remove all the nodes added to the canvas.
	The Reference Identifier Generator component is used to generate unique identifiers for DFCS dimensions even though source systems do not provide it.

3.2 Ingesting Data into Data Foundation Cloud Service for Banking

To create a connector for ingesting data into Data Foundation Cloud Service for Banking, complete the following steps:

1. Navigate to the home page, click **Data Services**, and then click **Add Connector**.

   The connectors which are defined are listed here.

2. From the menu select **Connectors** if it is not already selected.
3. Click **Add**.

The **New Connector** 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 view the list of EDDs defined in the default EDS of the ERP File Extracts.
6. Select the default EDS-ERP File Extracts file type EDD and drag it on to the canvas.
7. Click **Target**.
8. Search for the ADI you are looking for.
9. Select the required ADI and drag it on to the canvas and then link the input and output nodes.
10. To link the nodes, right click on a component and click **Link to** and select the component you want to link it with. You can also use your mouse to link the nodes. Point your mouse on the white circle of the input node and with your left-mouse button pressed, move the cursor to the white circle on the output node and release the mouse button.
11. At any given time, you can right-click a node to either link, delink, remove inlinks/outlinks, or delete a node.
12. To edit or view the properties, on the Connector window, click the Next button on the top of the page.
13. Under **Connector Details**, enter the name and description for the connector.
14. Under **Pre-Load Options**, select the truncate option to be defined in the target.
    - Select one of the available truncate options if you want to remove data from the table. Select No, if you do not wish to truncate the table before loading.

    The Partial Truncate is enabled and should not be selected.

    > **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. Example: To truncate the entire table. Here no expression is required.
    - Select **Selected Rows** to truncate based on a specified filter expression. Click the edit button next to **Specify filter expression** to define the expression. Example: If you want to remove some rows, use the **Selected Rows** option. Specify the filter condition for the rows to be deleted. Those rows are removed from the table before load.
      – Click **Edit** to filter the selected rows.
      – Select the required entity and click **Validate**. This validates the expression.
      – Click **OK** after the expressions are selected.
15. Under **Properties**, enter the value in **the Target Rejection Threshold** field. You can define the value in number or percentage of error records. For more details, see the Using Target Rejection Threshold section.
16. Click **Save** or **Save As Draft**.

> **Note**
>
> If one or more connectors are referring to the same File or EDD, then create a process pipeline to execute the connectors sequentially and they must not be executed in parallel.

3.3 Filter

This section provides information about the **Filter** component.

1. Drag and drop the **Filter** component onto the canvas to define a filter for an entity such as **EDD** (Insert Connector) or **ADI** (Extract Connector).
2. To position the component on the canvas, either drag and move it, or click the component and use the arrow keys on the keyboard to adjust its position.
3. If the filter component accepts input only from an entity and can have only one output.
4. To apply filters to multiple entities, drag and drop the corresponding number of filters. Connect each filter to its respective entity and define their expressions. For example, to add a filter to three entities, place three filters on to the canvas and connect them accordingly.
5. At any time, right-click the filter component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. To define the filter expression, double-click filter component. The **Filter Expression** window is displayed, showing selected entities and parameters. Specify the required filter expression using columns and parameters.
7. Click **Validate** to verify the correctness of the SQL Expression.
8. Click **OK** to confirm and apply the filter expression.

> **Note**
>
> - You do not need to add a **WHERE** clause for the filter.
> - For **file data loading**, use the filter expression of the type **Number** with single quotes. For example: `N_DRAWN_AMOUNT = '40000'`.
> - For the **Date** field, see `To_CHAR` function for comparison.
> - 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')`

3.4 Join

This section provides information about the Join component.

1. Drag and drop the Join component on the canvas to link multiple entities such as EDDs (Insert Connector)/ADIs (Extract Connector).
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. Join accepts input from two entities.

4. 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**.
5. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. Double-click the Join component to define a join condition. The selected entities are displayed on the left and right tabs.
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 a column from the left, select a column from the right tab and click = (Add Join). This displays the joined entities. You can join multiple entities.
9. To remove two joined conditions, select the two columns from the left and right tabs, 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.

3.5 Lookup

This section provides information about the Lookup component.

1. Drag and drop the Lookup component on the canvas to lookup values from an entity.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. The lookup component accepts input from two entities. One from the Value Entity and the other from the Lookup Entity.
4. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
5. Double-click the lookup component to define a lookup condition. In the Lookup window, you will see the connected entities on the left and right tabs.
6. You can drag the lookup entities on the right and left to reorder them.
7. To specify the lookup condition, then select a data element from the left, select a data element from the right and click = (Add Join). The lookup condition is displayed at the bottom of the window. Do this for as many entities on the left and right tabs you want to define.
8. To remove a lookup condition, select data elements from left and right entities and click **Remove Join**.
9. Click **Reset** to reset the lookup condition.
10. Click **Ok**.

> **Note**
>
> This creates a left outer join between the connected entities.

3.6 Aggregation

This section provides information about the Aggregation component.

1. Drag and drop the Aggregation component on the canvas to define an aggregation on an EDD.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. Aggregation component accepts input only from an EDD and it can have only one output.
4. If you have multiple EDDs to be aggregated then you must select as many number of aggregation components, connect each to the respective EDD, and then define their group by having clauses. Example: To add aggregation to three EDDs, drag three aggregation components on to the canvas and link them to their respective EDDs.
5. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. Double-click the Aggregation component to define an aggregation condition. In the Aggregation Window, you will see the selected EDD under the Entities tab. Double click the entities, functions, and operators to build your expression in the Group By field. Repeat this step to build your clause in the Having field.
7. Click **Validate** to verify the correctness of the SQL Expression.
8. Click **Ok**.
9. Click **Reset** to reset all the aggregation conditions and begin afresh.

3.7 Reference Identifier Generator

This section provides information about the Reference Identifier Generator component. Use this component to generate a unique sequence identifier for the selected attribute.

1. Drag and drop the **Reference Identifier Generator** component on the canvas.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
4. Double-click the component to define a reference identifier generator condition.
5. Provide a **Name**.
6. Specify the source attributes with unique values by selecting the relevant value from the drop-down list.
7. Specify **Identifier Type** as Numeric or Varchar.
8. Select the **Target Attribute**.
9. Click **OK**.

3.8 Transpose (Rows to Columns)

This section provides information about the Transpose (Rows to Columns) component.

1. Drag and drop the Transpose (Rows to Columns) component on to the canvas.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. The Transpose (Rows to Columns) component accepts input only from an EDD and can have only one output.
4. 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 number of Transpose (Rows to Columns) components, connect each to its respective EDD, and then define their expressions.
5. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. Double-click the component to transpose the entity rows into columns. Specify the pivot data element to transpose rows into columns.
7. Specify the Column name - matching row value – Expression combination. You must have a minimum of two combinations.
8. Click **Review** to review the transformation. The sample of the transformed data is displayed.
9. Click **Ok**.

3.9 Transpose (Columns to Rows)

This section provides information about the Transpose (Columns to Rows) component.

1. Drag and drop the Transpose (Columns to Rows) component on the canvas to define a Transpose (Columns to Rows) Component on an EDD.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. The Transpose (Columns to Rows) component accepts input only from an EDD and can have only one output.
4. If you have multiple EDDs selected, and you want to have a Transpose (Columns to Rows) component for more than one EDD, then you must select as many number of Transpose (Columns to Rows) components, connect each to its respective EDD, and then define their expressions.
5. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. Double-click the component to transpose the entity columns into rows. Specify the **Unpivot Data Elements** to transpose columns into rows.
7. Specify the **Header Column Name** and **Value Column Name**.
8. Specify the column value (**Header column**) and expression pair (**Value column**) for each transposed row. You must have a minimum of two pairs.
9. After specifying the **Unpivot Data Elements**, click **Auto Transpose**. This will transpose columns into rows based on the unpivot data elements selected.

10. Click **Review** to view the transformed data.
11. Click **Ok**.

3.10 Derived Column

This section provides information about the Derived Column component.

1. Drag and drop **Derived Column** component on the canvas.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. Connect the output of the **Derived Column** component to the **Mapping** component.
4. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
5. Double-click the **Derived Column** component and click **Add** to define a new derived column.
6. On the right pane, under **Entities**, double-click to select the entities which you want to use to build the expression. The selected entities appear in the Expression field at the bottom. Modify the expression based on your needs. Provide the Name for the derived column.
7. Click **Validate** to verify the correctness of the SQL Expression.
8. Click **Apply**. The saved details appear as a list on the left pane.
9. Click the **Edit** button to modify the name and expression of the required Derived Column.
10. Repeat the steps to create as many Derived Columns as you require.
11. Click **Ok**.

3.11 Mapping

This section provides information about the Mapping component.

1. Drag and drop the **Mapping** component on the canvas to define a mapping. Connect the inputs and outputs for the **Mapping** component before specifying the mapping details.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. Click **Mapping** component on the canvas. The EDDs, ADIs, and their respective data/ derived data elements are displayed in the Mapping window.
4. Filter the source attributes by selecting an option from the drop-down list. Click a data element under **Source**.
5. Filter the target attributes by selecting an option from the drop-down list. Click a data element under **Target** and then click the **Map** button. The column mappings are displayed.
6. Click the **Map** button. The mapped items are displayed on the right column mappings are displayed.
7. Repeat the steps to map as many source-target pairs as required. Note that you can use the filter icons for the **Source** and **Target** lists to filter unmapped, mandatory, or those items that are valid for the application.
8. The following validations are performed for the mapping:
   a. Data type validation
   b. Data length validation

c. Data precision validation

9. The result of the validation for each mapping is indicated as **Remarks** next to each mapping.
10. At any given time, you can select the **Unmap** button to unmap the source and target.
11. 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 and target.

12. Hover your mouse over a data element under **Target** column to see detailed information which includes the description, length, and scale.
13. Use the search icons to search for source or target data elements.
14. Click the **Delete All** icon to delete all the mappings. You can also delete individual mappings by selecting the cross symbol next to the column mapping.
15. Click the **Import Mapping** icon to import a mapping Excel sheet.
16. Click the **Export Mapping** icon to export the mapping information in an Excel format.
17. Use the search field to search for mappings. You can search for an item based on the **Source Column Name**, **Target Column Name**, **Source Entity**, **Target Entity**, or **Remarks**.

3.12 Flattened Table to Hierarchy

This section provides information about the Flattened Table to Hierarchy component.

1. Drag and drop the **PC Hierarchy** component on the canvas.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. Connect an EDD / Source Filter to the input of the **Flattened Table to Hierarchy** component.
4. Connect the output of the **Flattened table to Hierarchy** component to the **Join / Lookup / Mapping** component.
5. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. To specify the details, double-click the **Flattened table to Hierarchy** component.

   The **Flattened Table to Hierarchy** window is displayed.
7. Choose the **Hierarchy Type** as **Balanced**, **Ragged**, or **Skipped**. Click **Help** icon to view the details and understand how the hierarchies are defined.
8. Specify the **Number of levels** in the hierarchy.
9. Specify the **Parent Node Column Name** and **Child Node Column Name**.
10. Select the **Key Elements** from the drop-down list. You can select multiple Key Elements for a hierarchy.
11. Select all the node level details from the respective drop-down lists.
12. Click **Review** to view the transformation changes.

13. Click **Ok**.

3.13 Hierarchy Data Flattening

The hierarchy flattening component in the Data ingestion definition allows you to use input files in a parent-child hierarchy structure and convert them into a flattened dataset during the ingestion process without any additional configuration outside the system. The application supports balanced, unbalanced, and skipped hierarchies for flattening.

This section provides information about the Hierarchy to Flattened Table component.

1. Drag and drop the **Hierarchy to Flattened Table** component on the canvas.
2. To position the component on the canvas, drag and move the component, or click the component and use arrow keys on the keyboard to move it around.
3. Connect an EDD / Source Filter to the input of the **Hierarchy to Flattened Table** component.
4. Connect the output of the **Hierarchy to Flattened Table** component to the **Join / Lookup / Mapping** component.
5. At any given time, right-click the component and select the relevant options to either delink, remove inlinks or outlinks, or delete the component.
6. To specify the details, double-click the **Hierarchy to Flattened Table** component.

   The **Hierarchy to Flattened Table** window is displayed.
7. Select **Key Elements** from the drop-down list. You can select multiple Key Elements for a hierarchy. Parent or Child attributes cannot be selected as Key Elements.
8. Specify the maximum **Number of levels** in the hierarchy. By default, the number of levels is set to 3.
9. Select the **Parent Identifier**, **Parent Code**, and **Parent Name**.
10. Select the **Child Identifier**, **Child Code**, and **Child Name**.
11. Click **Review** to view the transformation changes.
12. Click **Ok**.

3.14 Specifying Alias in Connector

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

Aliases are automatically initiated by Data Integration Hub (DIH) as and when required while defining connectors.

The following scenario explains the mechanism:

1. Drag and drop an 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.
2. Enter a name of your choice under **Alias Name** and click **Ok**.

   Note that the Alias Names must be unique within a connector.

3.15 Modifying and Viewing a Connector

To edit or view a connector, complete the following steps:

1. To edit or view a connector, you can select the required connector from the Connector Summary.
2. The details of the selected connector are displayed. You can modify or view the details.
3. Modify the connector's details as required. Connector Name cannot be edited.
4. Click **Save**.
5. To make changes to a published connector, click **Unpublish**. This 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.

3.16 Copying a Connector

To copy an existing connector, complete the following steps:

1. Click the Copy button of the required connector. Depending on the view in which the original connector was created, the copied connector too will have the same view.
2. Enter the **Name** and **Description**.
3. Click **Save**. The details are saved with a new connector name. The existing connector remains unmodified.

3.17 Deleting a Connector

To delete an existing connector, complete the following steps:

1. Click the **Delete** button of the required connector.
2. Acknowledge the confirmation message.

3.18 Search and Filter

Use the search and filter options in the Connector Summary page to find the required connector. Enter the nearest matching keyword to search, and filter the results. You can search for a connector with either the name, description, or status of the connector.

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

3.19 Parameters in Connector

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

For example, if you want to use the Runtime MIS Date as the parameter, you must first convert it to date. 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. You must convert it to a valid SQL date format.

3.20 Using Target Rejection Threshold

Target Rejection Threshold is a value defined for the number or percentage of records with errors that are allowed before the execution is configured to fail when you move records from the source database to the target database. It can be entered as an absolute value or as a percentage in the **Connector Properties** window.

Financial Services Data Foundation Cloud Service for Banking creates an error table by duplicating the target table name and appending it with **_ERR** during the execution process. The erroneous records are logged into the error table and valid records are moved to the relevant table in the target. The errors can be of the types notional, public key, or data. For example, if the target table name is STG_CASA, then the error table name is created as STG_CASA_ERR. The objective is to log only a required number of errors and then correct them in the source table. After correcting the errors, rerun the execution. In the process, you can iteratively collect the errors and correct them, making the task of correcting the errors in the source table much simpler.

You can also move records from file table to the target database and the structure of the table remains same as file table added with five additional columns. For example, if the file table name is EDD12_THRESHOLD_VOLUMETEST, then the error table name is created as EDD12_THRESHOLD_VOLUMETEST_ERR.

Let us understand this process with the help of the following example.

1. Enter 10 in the Target Rejection Threshold field.

   > **Note**
   >
   > a. To specify the value as 10 percent of the records in the execution, enter 10%.
   > b. If you leave the field blank or enter 0, DFCS reads the threshold as 0 and the execution fails when an error is encountered.
   > On execution of the process, DFCS permits up to 10 records with errors to process to the target table and any number of valid records. If the number exceeds 10, the execution process fails and stops. The erroneous records are available in the target table name appended with _ERR.

2. Click **Data Ingestion** and select **View Data**. This will display **Data Visualization** window where you can search for the target table name appended with _ERR.
3. Check the details in the error table. For more information, see Reading the Error Table.
4. After checking the error table details, navigate to the source table, and troubleshoot the errors.
5. After correcting the errors, rerun the execution.
6. Repeat the process iteratively and correct the errors.

3.20.1 Reading the Error Table

Financial Services Data Foundation Cloud Service for Banking creates the error table in the data service execution layer/schema to store records with errors. The structure of the error table is similar to the target table but with five additional columns. You can refer to the following for executions:

- ORA_ERR_NUMBER$ - The error number.
- ORA_ERR_MESG$ - The ORA error message with the description of the error.
- ORA_ERR_ROWID$ - The row ID of the error.
- ORA_ERR_OPTYP$ - The operation type.
- ORA_ERR_TAG$ - This column stores the Execution ID. You can use this column to filter and view error records for a specific execution.

3.21 Executing Connectors

Use the Process Modelling Framework to execute a DIH connector.

3.21.1 Executing Connector Using Process Modelling Framework

Refer to the Data Operations User Guide for more details.

3.21.2 Specifying Runtime Parameters

Note the following points regarding runtime parameters:

- If the connector contains runtime parameters, they can be set in the Variables input field of the connector's definition window. Example: FILE_DATE=31-Jan-2022.
- 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 a part of connector mappings or filter expressions, specify them as given in this example. Example: FILE_DATE=$MISDATE:dd-MMM-yyyy.

3.22 Quality

For information about the data quality checks (DQ Checks) and the out-of-the-box pipelines, see the Data Quality Checks section in the Oracle® Financial Services Data Foundation Cloud Service for Banking User Guide.

3.23 Protection

This section explains the General Data Protection Regulation (GDPR) related data protection methods implemented in Data Foundation Cloud Service (DFCS).

3.23.1 Data Redaction

Data Redaction is one of the data security features that protects sensitive data against unauthorized access and data theft.

Data Redaction is an built-in process in DFCS. It is applied automatically on all the business terms containing Personally Identifiable Information (PII).

To implement data redaction on a business term, on the **New Business Term** page, enable the Personally Identifiable Information button for that business term. For information about business term creation, see the Manage Business Terms section in the Oracle® Data Foundation Cloud Service Data Catalog.

4

Application Data Services (DFCS Only)

Application Data Service (ADS) feature of DFCS enables secure, automated data exchange between DFCS and other downstream applications such as Profitability and Balance Sheet Management Cloud Service (PBSMCS). ADS provides pre-built components for data exchange jobs like pre-built Glossary mapping and connectors. Users can view out-of-the box Glossary mapping and Connectors. The solution also supports extension of existing connectors.

This user guide demonstrates:

- Setting up Application Data Services
- Managing Application Data Service Definitions
- Subscribing Application Data Services
- Published Application Data Services

For information on Glossary mapping and connectors for PBSMCS see, DFCS Integration with PBSMCS for ADS User Guide.

4.1 Setting up Application Data Services

Setting up Application Data Services involves below steps:

- Create A New User in DFCS IDCS Console for ADS Subscription
- Setup User Access for ADS Subscription
- Get the OAuth Client ID and Client Secret for the User
- ADS Version Compatibility between Subscriber and Publisher

4.1.1 Create A New User in DFCS IDCS Console for ADS Subscription

Steps to create a new user in DFCS IDCS console

1. Enter the IDCS URL in the browser's URL Address Bar. The Oracle Cloud Account Sign In Window appears.
2. Log in to Oracle Identity Cloud Service (IDCS).
3. Select Oracle Cloud Services. For more information, see Accessing the IDCS Console in the Identity Management Guide.

> **Note**
>
> - Ensure the DFCS Cloud Administrator who creates your account has granted you administrative privileges to access the IDCS Console.
> - For more information on the privileges available, see User Roles and Privileges.
> - If the DFCS Cloud Administrator has granted you only Identity Management privileges and no other DFCS privilege, you are automatically redirected to the IDCS Console specific to DFCS after logging in successfully.
> - After a User signs in to the DFCS Cloud Service, the User to User-Group Mapping created in the **IDCS Console** will onboard into the Master and Mapping Tables.
>   For more information about how to Unmap a User from a Group in the **IDCS Console**, see the **Create Application Users** section in the Getting Started Guide.

4.1.2 Setup User Access for ADS Subscription

To access the FSGBU OFSAA SaaS (Example: PBSMCS) application subscription user interface, an **application user** must be assigned to the **DFCS_<*App ID*>_INTEGRATION** group. For example: PBSMCS user must be mapped to **DFCS_PBSM_INTEGRATION** group.

Although the user may be assigned to additional groups, membership in the **DFCS_<*App ID*>_INTEGRATION** group is the minimum requirement. Failure to meet this requirement will result in an error.

Similarly, a **DFCS user** must be mapped to the integration groups listed below.

- DFCS_PBSM_INTEGRATION
- IDNTY_AUTH
- IDNTY_ADMN
- DFBDATANALST
- DFBINTANLST
- DFBPRCADMIN

> **Note**
>
> If no user is assigned to the group, the system cannot be used for subscription. For more information on user group see, User Group and User Role Mapping.

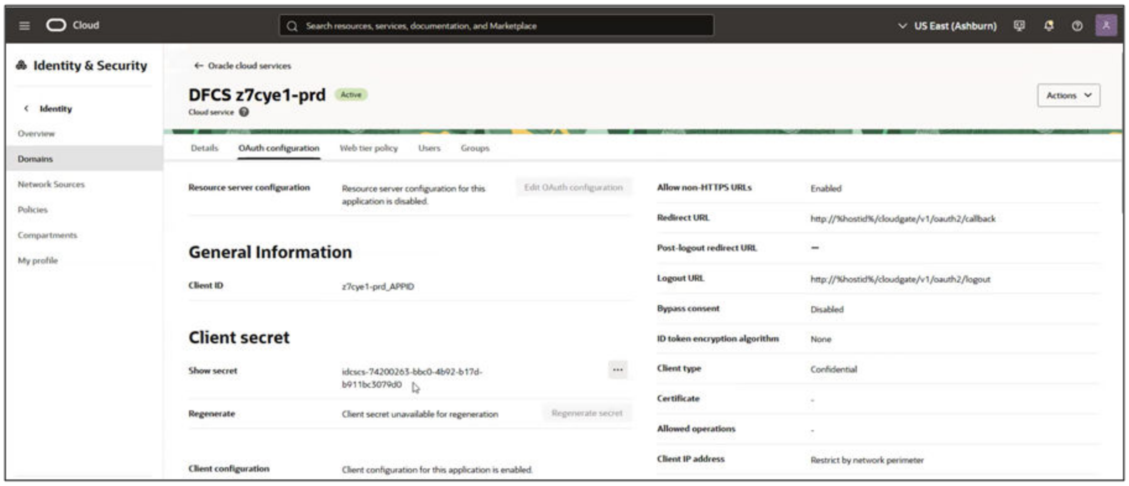
4.1.3 Get the OAuth Client ID and Client Secret for the User

To get the OAuth Client ID and Client Secret, follow these steps:

1. Enter the IDCS URL in the Browser's URL Address Bar. The Oracle Cloud Account Sign In Window appears.
2. Log in to Oracle Identity Cloud Service (IDCS).
3. From the LHS, click **Domain** and select the DFCS provisioned domain.

4. Select **Oracle Cloud Services**. For more information, see [Accessing the IDCS Console] from Administering Oracle Identity Cloud Service.
5. From the **Oracle Cloud Services** window, search and select the required DFCS Service from the list.
6. Click the **OAuth Configuration** tab. The Client ID and Client Secret Details are displayed in the General Information section.
7. Copy the Client ID and Client Secret.

Figure 4-1 Oracle Cloud Service - Client Secret



4.1.4 ADS Version Compatibility between Subscriber and Publisher

This section lists the ADS versions on the Subscriber (ADS-PBSM) that are compatible with those on the Publisher (ADS-DFCS). The system permits subscription and refresh operations only when the ADS versions are compatible. This applies to the following components:

Table 4-1 Version Compatibility for DFCS and PBSMCS

ADS-PBSM Version	ADS-DFCS Version	Result- Version Compatibility
25.9.1	25.9.1	Allowed – Operation Successful.
25.6.1	25.6.1	Allowed – Operation Successful.

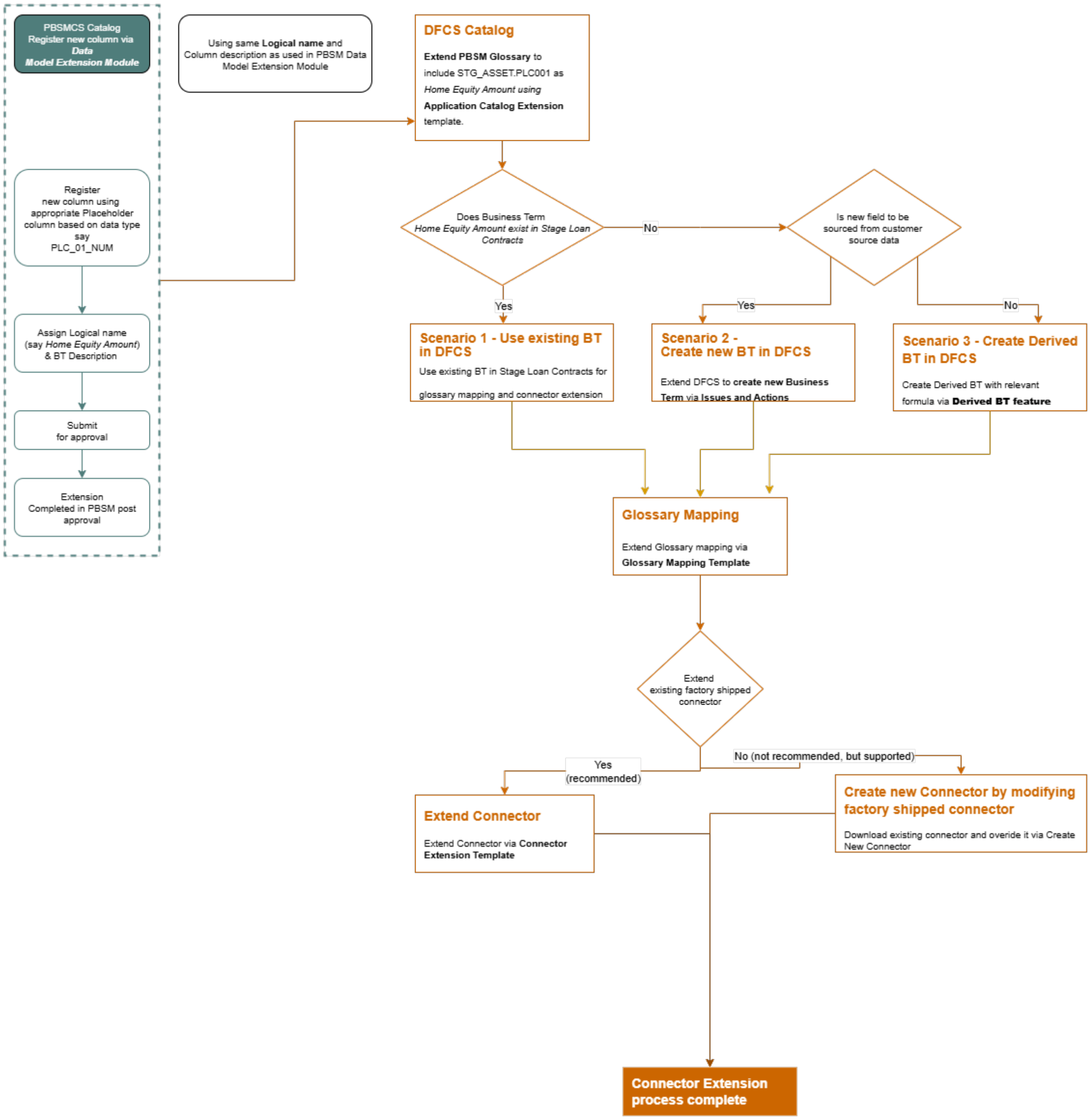
4.2 Managing Application Data Service Definitions

The Catalog section on the DFCS home page provides access to Application Data Services (ADS) related glossary mappings and connectors. Users can browse available solutions to view out-of-the-box glossary mappings and connectors. Users can also extend or override existing connectors as needed.

For example, a bank may need to add an additional attribute to a PBSM entity and include it in the ADS connectors.

The diagram outlines the steps for extending connectors and business terms (BT) in DFCS:

Figure 4-2 DFCS ADS Connectors



This below list outlines the key steps followed to browse and manage ADS definitions.

1. **Register New Column**: Register a new column in the PBSCMS Catalog with a logical name and description, then submit for approval.
2. **Extend DFCS Catalog**: Add the new column to the PBSCM glossary using the Application Catalog Extension template.
3. **Business Term Check**: This step results in one of the following three scenarios:

Scenario	Description	Required Action
Scenario 1	The newly added Business Term in the ADS application already exists in the DFCS entity.	Create a Glossary Mapping and extend the existing connector.

Scenario	Description	Required Action
Scenario 2	The newly added Business Term in the ADS application doesn’t exist and has no data points available for derivation (data must be sourced from the customer).	Create a new Business Term in DFCS, then perform Glossary Mapping and extend the existing connector.
Scenario 3	The newly added Business Term in the ADS application doesn’t exist but can be derived from existing DFCS data points.	Create a Derived Business Term and override the existing connector by creating a new connector using the current one as the base.

4. **Extend Connector**: Extend or modify the existing connector as needed.
5. **Completion**: Finalize the connector extension process.

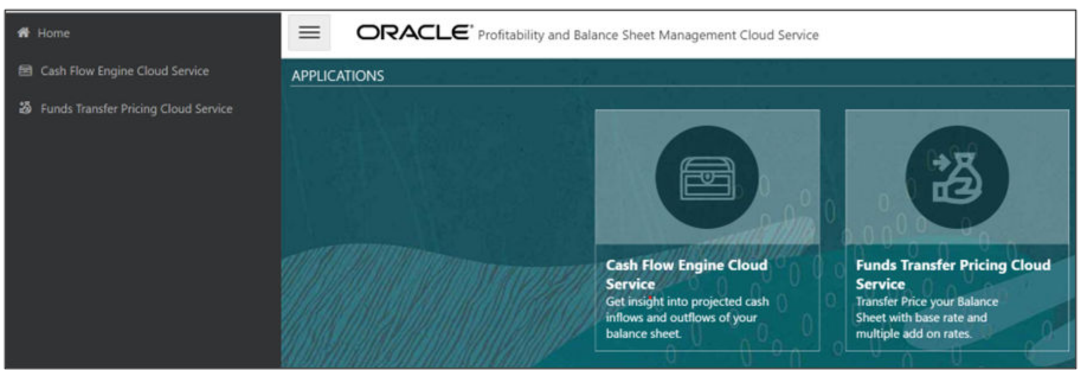
Refer to Oracle® Data Foundation Cloud Service Data Catalog for **Managing Application Data Service Definition**.

4.3 Subscribing Application Data Services

This section describes configuring Application (say, PBSMCS) to receive data from DFCS by establishing a subscription, enabling the data import process from the publisher.

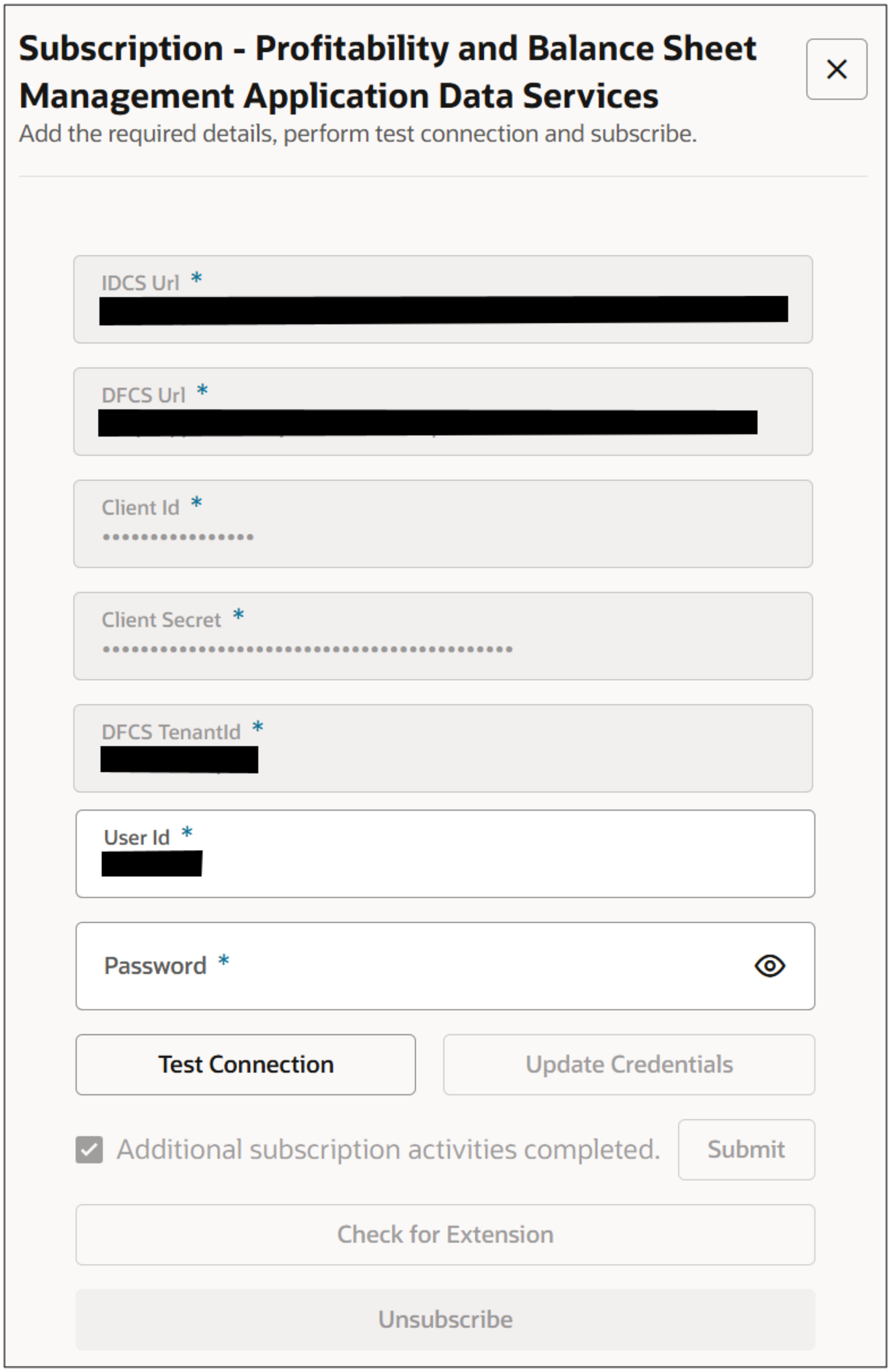
1. Start with subscribing OCI Application, say, **Profitability and Balance Sheet Management Cloud Service**.

Figure 4-3 Profitability and Balance Sheet Management Cloud Service Console



2. Select a specific service, say, **Funds Transfer Pricing Cloud Service** option and then from the LHS, click **Admin Tools** → **Data Foundation Integration** → **Data Services Subscriptions**.
3. Click **Subscribe** if you are subscribing for the first time. The Subscription pane is displayed.

Figure 4-4 Subscription Pane



- **For First-Time Subscription**:

a. Enter the following.

Table 4-2 Application Data Service

Field	Description
IDCS URL	To obtain the IDCS URL, the user must log in to the DFCS Identity Cloud Service (IDCS) instance that is used for authentication. This enables PBSMCS to authenticate securely with DFCS. **Note** For more information on how to obtain the IDCS URL, see Get the OAuth Client ID and Client Secret section in the Application Data Services (DFCS Only) chapter.
DFCS URL	The base URL of the **Data Foundation Cloud Service (DFCS)** instance. It specifies the source environment from which PBSMCS will receive data. **Note** For more information on how to obtain the IDCS URL, see Get the OAuth Client ID and Client Secret section in the Application Data Services (DFCS Only) chapter.
Client ID	A unique identifier registered in IDCS for the application (PBSMCS). It is used as part of the OAuth2 authentication process when establishing a connection with DFCS. **Note** For more information on how to obtain the IDCS URL, see Get the OAuth Client ID and Client Secret section in the Application Data Services (DFCS Only) chapter.

Table 4-2 (Cont.) Application Data Service

Field	Description
Client Secret	A confidential key associated with the Client ID. It is used to authorize the PBSMCS application to access DFCS data securely. **Note** For more information on how to obtain the IDCS URL, see Get the OAuth Client ID and Client Secret section in the Application Data Services (DFCS Only) chapter.
DFCS Tenant ID	The unique identifier for your DFCS tenant environment. It ensures that the connection and data transfer are scoped correctly to your organization's DFCS instance.
User ID	The DFCS application user's login ID authorized to access and retrieve data from the DFCS environment. This user must have the appropriate roles and permissions for data publishing. **Note** The User ID is case-sensitive and must be created in lowercase letters.
Password	The corresponding password for the DFCS User ID. It is used during authentication to validate the user's access when connecting PBSMCS to DFCS.

b. Click **Test Connection** to verify the connection.

> **Note**
>
> This checks if the current credentials are working or if there's a connection issue.

c. Once the credentials are entered correctly, you can click **Update Credentials** to save your changes.

d. If successful, a message *“Operation successfully connected”* appears at the top of the pane, and the **Subscribe** button is enabled.

e. If the test fails, recheck and verify the entered details.

f. Click **Subscribe**.

g. When the subscription starts, a message *“Please Contact Oracle Support to Complete Additional Activities”* will appear.

> **Note**
>
> Subscription completion may take a few minutes.

h. If you've already subscribed, you can unsubscribe from the service or make changes to your subscription, click **Unsubscribe**. The pop-up message informs you that unsubscribing from the **DFCS Tenant** is **permanent** and **cannot be undone**.
You have two options:
   - **Cancel**: If you change your mind and do not want to unsubscribe, click the **Cancel** button. This will close the prompt and allow you to continue using the service.
   - **Yes, Continue**: If you're sure you want to unsubscribe, click the **Yes, Continue** button. This will confirm the action and unsubscribe you from the service permanently.

i. After the subscription is completed, raise a **Service Request (SR)** with the DFCS Tenant details and PBSMCS Tenant details to finalize the additional subscription process.

j. Once the SR is processed and confirmation is received, select the **Additional Subscription Activities Completed** checkbox.

k. Click **Submit** to complete the subscription.

- **For Editing an Existing Subscription**:

  a. Update the **User ID** and/or **Password** if required.

  b. Click **Update Credentials** to save the changes.

> **Note**
>
> If you are already subscribed, click **Edit Subscription** to update the User ID and Password. Note that fields such as **IDCS URL, DFCS URL, Client ID, Client Secret,** and **DFCS Tenant ID** cannot be edited, as they are mapped/ created by the admin during the initial subscription. For more information, see Accessing the IDCS Console.

You can also click **Check for Extension** if extensions have been already published.

4.3.1 How to Refresh the Extension from DFCS

After subscribing to Application Data Services (ADS), you may need to apply **extensions** to include additional data elements or custom mappings. The following steps outline how to verify, extend, and execute the required processes.

1. **Check for Extension (PBSMCS - Subscriber)**:
   - After completing the ADS Extension at the DFCS setup, click **Check for Extension**. A confirmation message appears: **Do you want to proceed with the extension?**
   - Click **Confirm** to proceed.
   - Then click **Submit** to finalize the confirmation.
2. **Refresh ADS**:

- Click **Extend ADS** to trigger the extension of Application Data Services.
- This process updates the existing **PMF** specific to the extensions.

4.4 Published Application Data Services

The **Data Services Publications** screen in DFCS displays all active subscribers receiving data via Application Data Services (ADS). This helps validate that the subscription from **Profitability and Balance Sheet Management (PBSMCS)** to DFCS has been successfully established.

1. Access the Data Foundation for Banking portal.
2. Click **Administrator** → **Data Services Publication**.
3. Verify that Profitability and Balance Sheet Management Application Data Services appears as a subscriber.
4. Confirm that your PBSMCS instance (e.g., PBSMADS E26SJU-PRD) is listed.
5. Review the number of:
   - Application Data Services
   - Entities Data Shared
   - Attributes Data Shared
6. Click the arrow next to the subscriber name to view more details about the shared ADS, entities, and attributes.

5 Integration with Oracle Applications

This section explains the pre-built integration with other Oracle Applications.

5.1 Integration with Oracle Enterprise Data Management Cloud Service

Enterprise Data Management Cloud Service (EDMCS) is an Oracle Cloud application designed to centrally manage and govern enterprise data across business applications. It provides a unified platform for creating, maintaining, and sharing master data such as dimensions, hierarchies, properties, and mappings.

Key features of EDMCS include:

- Centralized data governance with audit tracking
- Support for multiple hierarchies and viewpoints
- Role-based access and workflow approvals
- Integration with Oracle and non-Oracle systems
- Simplified modelling of enterprise structures with drag-and-drop interface

This section outlines the steps required to source dimension-related data from EDMCS into DFCS, a key capability for managing dimensions within DFCS.

You can significantly save the efforts required to manage dimensions in DFCS if those dimensions are already available and maintained in EDMCS within another application, such as *Oracle Cloud ERP*. A dimension from an existing EDMCS application—like *Oracle Cloud ERP*—can be subscribed into the EDMCS application for DFCS. This eliminates the need to manage these existing dimensions independently for DFCS.

> **Note**
>
> When setting up a DFCS application in EDMCS for the first time, you must manually import the existing data for the subscribed dimensions from the source EDMCS application. After the initial import, the subscription will automatically capture new changes and updates (including edits).

For detailed instructions on setting up such subscriptions, refer to the EDMCS User Guide

5.1.1 Pre-Configured Standard Dimensions

The following dimensions are delivered by default in EDMCS and are ready for use:

Table 5-1 Pre-Configured Standard Dimensions in Use

Dimension	Purpose
Product	Defines products or services offered by the organization.
Project	Captures details of specific projects or initiatives.
Data Source	Identifies the origin of data within the system.
Business Segment	Represents major business divisions or market segments.
Line of Business	Groups related products/services under a business line.
Transaction Type	Classifies financial or operational transactions by type.
Organization Unit	Represents organizational entities such as departments or regions.
Product Type	Categorizes products into high-level types or families.
General Ledger	Stores chart of accounts and financial account structures.
Legal Entity	Maintains information about registered legal entities.
Branch	Lists and manages details of organizational branches.

> **Note**
>
> These dimensions can be directly used or customized by adding, renaming, or restructuring nodes to suit specific business requirements.

5.1.2 Validations in EDMCS

In our EDMCS setup, the following validation rules are implemented to maintain hierarchy integrity and ensure consistency across dimensions:

1. All leaf nodes must be shared for all hierarchies as leaf nodes

- **What it means:** The lowest-level items (leaf nodes) must be the same across all hierarchies of a dimension.
- **Applicability:** This rule is enforced when multiple alternative hierarchies exist for a given dimension
- **Example:** If "Product" dimension has *Product A* as leaf in one hierarchy, then *Product A* must appear as leaf in every other hierarchy.

Intermediate nodes must be unique across a dimension

- **What it means:** A non-leaf node cannot be shared whatsoever. By extension, a specific parent–child link can only belong to one hierarchy.
- **Example:** If “Product Y” is a leaf node and "Category X → Product Y" is a parent-child relationship in one hierarchy, the exact same relationship cannot appear in another hierarchy.

- **But allowed:** Leaf nodes can have a different parent across hierarchies as long as respective parent names in those hierarchies are unique.

> **Note**
>
> - All hierarchies must have the same leaf nodes.
> - But the way these leaf nodes are related to their parent (parent–child links) must be unique in each hierarchy.

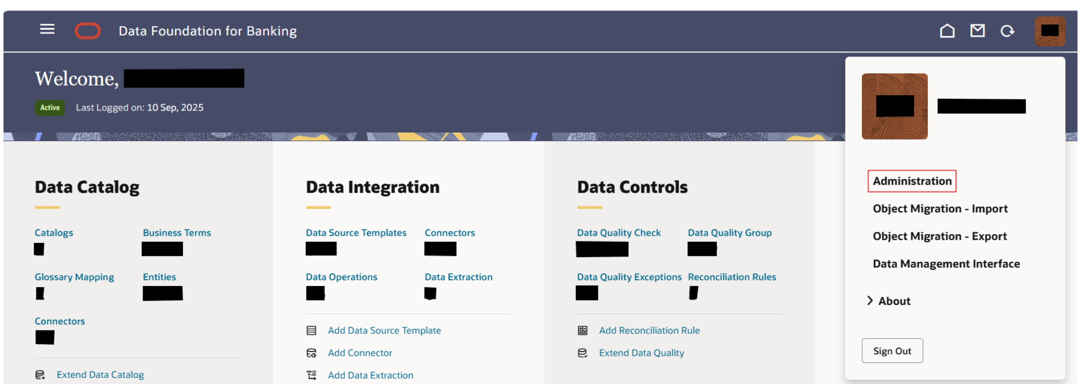
5.1.3 Steps for Integrating EDMCS with DFCS

This interface allows users to provide necessary credentials and connection details to integrate EDMCS (Enterprise Data Management Cloud Service) with the Data Foundation Cloud Service. It allows DFCS to fetch and utilize dimension metadata for applications.

> **Note**
>
> Before getting started user must submit a Service Request (SR) with My Oracle Support to have the EDMCS URL whitelisted.

1. Navigate to the **DFCS Home Page** > **User Menu** > **Administration** > **Integration with EDMCS**.
   The **EDMCS for Data Foundation Cloud Service** page appears.

   Figure 5-1 EDMCS Navigation

2. On the **Administration** page, click on the **Integration with EDMCS** tile.
3. Enter the credentials to access the EDMCS.

Figure 5-2 EDMCS Configuration Details

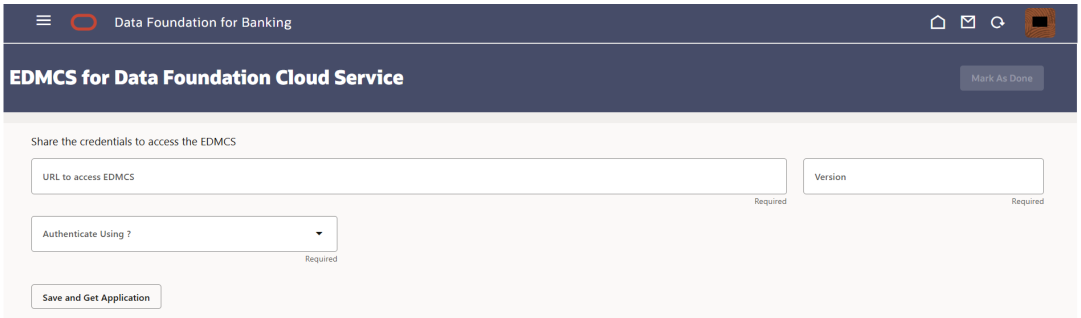


Table 5-2 Share the credentials to access the EDMCS

Field	Description
URL to access EDMCS	Enter the URL for your EDMCS instance in this field. This field is required. (e.g., https://<prod>.oraclecloud.com). Enter 500 or fewer characters.
Version	Enter the version of your EDMCS instance in this field. This field is required. (e.g., 24.04.30). **Note** Enter 20 or fewer characters.

4. Click **Save**.
5. Once saved, the **Mark As Done** option will be enabled.
6. Select **Authenticate Using?** dropdown. Select the preferred authentication method.
   - Two options are available:
     - **Password**
     - **Access Token**
7. **Option 1: Password Authentication**
   Once **Password** is selected:
   - Provide the **Username** and **Password** for EDMCS.

   Table 5-3 Password Authentication

Field	Description
Username	EDMCS service account username. Enter 100 or fewer characters.
Password	Corresponding user password. Enter 255 or fewer characters.

> **Note**
>
> Ensure the user has the necessary EDMCS API access privileges.

8. Option 2: Access Token Authentication
   If **Access Token** is selected:
   - Provide the full access token in the field:

     Specify the token

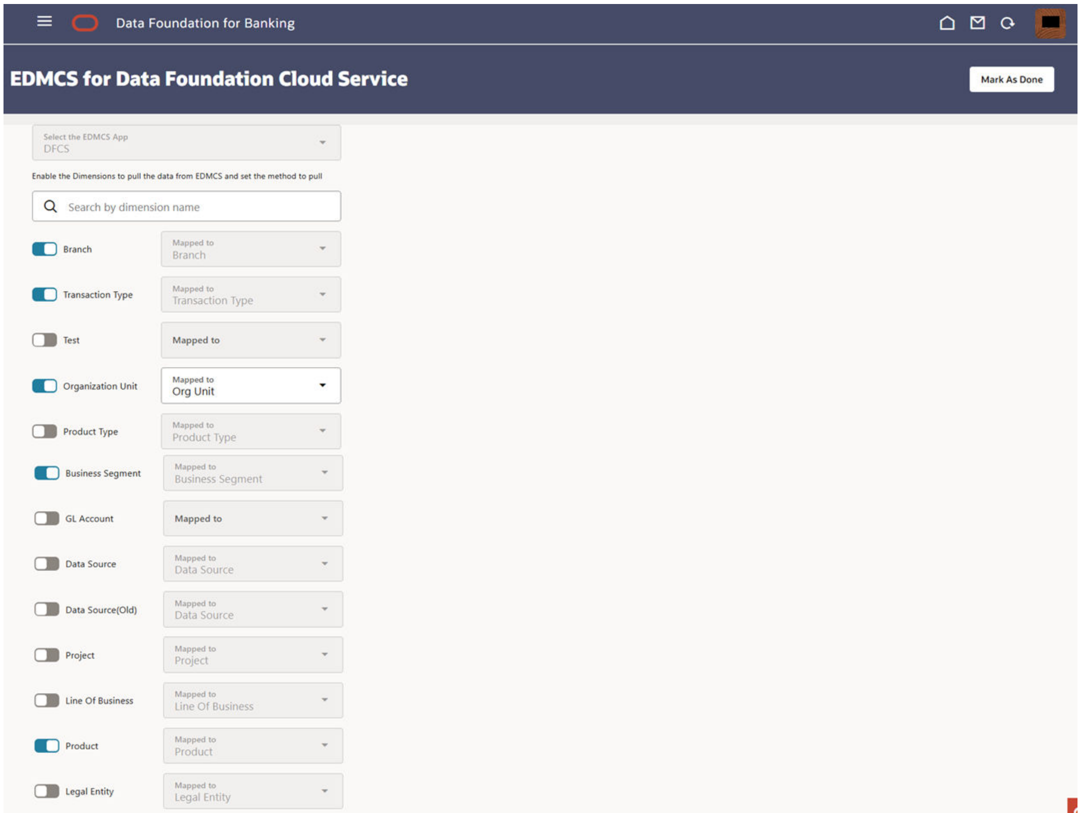
> **Note**
>
> Token must be valid and scoped for EDMCS APIs. Ensure the token hasn't expired.

9. Click **Save and Get Application** to:
   - Validate the connection
   - Fetch available EDMCS applications and their dimension structures

> **Note**
>
> If connection or credentials are invalid, appropriate error prompts will guide the user.

10. Select the required **Dimension**.
11. Click **Mark As Done** located at the top right to complete the integration setup.

Figure 5-3 Dimension Mapping



12. After saving successfully, the **PMF** will be created.

Checking PMF Executions

1. Navigate to the **Home** page → **Data Integration** → **Data Operations**.
2. On the Search bar, locate and click on **EDM_DIM_EXPORT_LOAD**.
   All selected dimensions will appear.
3. Click **Execute**.

Figure 5-4 EDMCS Pipeline

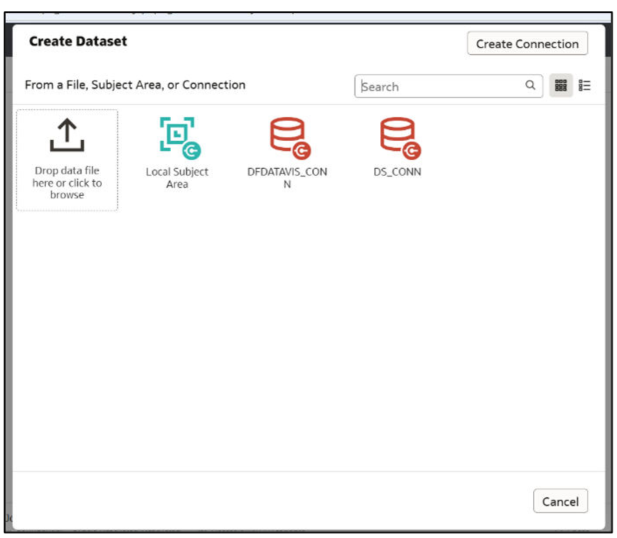
4. Select the **Execution Type** and enter the **Object ID**.

5. Click **Execute** again to run the process.
6. Go to the **Process Monitor** page.
7. Check the status for EDM:
   - **Completed** → Process ran successfully.
   - **Failed** → Double-click the failed record to view **Execution Logs** for troubleshooting.
8. Click **Execution Logs** to view the logs.

Validate Data Sourced from EDMCS

1. From the left menu click **Data Visualization** → **Source Data Visualisation**.
2. Click the **Search** box near the top.
3. From the Source Data Visualisation window, select **Others** or type the dataset name or keyword (e.g. branch) in the search bar and press **Enter**. The dataset page opens showing the data grid.
4. On the dataset page, you can filter the results datasets (e.g., **As Of Date**, **Data Source Code**).
5. Click the three-dots to perform the following actions:
   - **Sort By** — sort the dataset by a chosen column.
   - **Drill to Attribute/Hierarchy** — navigate to related attributes or hierarchies for the selected dimension value.
   - **Edit** — edit dataset metadata or definitions.
   - **Export** — export the data.
6. Click **Go back** to return to Oracle Analytics window and choose another dimension.
7. **Create New Dataset**: In the next screen, click on **Create**, then select **Dataset**.
8. **Choose Connection**: In the dataset creation screen, choose the connection **DFDATAVIS_CONN**.

Figure 5-5 Connections



9. **Expand Schema and Select Table**: Expand the **Schemas** section, then expand **DFREP**, and select the **STAGE_PRODUCT_MASTER** table.

10. **View Sourced Data**: The sourced data will now be visible for your selected dimension.

> **Note**
>
> Always ensure that configuration details are accurate before executing any process to avoid failures.

> **Note**
>
> For detailed instructions on Accelerator Applications, refer to Working with Accelerator Applications in the Oracle Cloud Enterprise Data Management guide.

6 Data Extraction

Steps for Accessing Data Extraction in DFCS:

1. **Navigate to the Home Page**: Open the DFCS User Interface.
   Access Data Integration: From the **Left-Hand Side Menu**, click on **Data Integration**.
2. **Data Extraction**: Under **Data Integration**, click on **Data Extraction**. This will display the basic details of the extraction definitions.
3. **Understand Extraction Behavior**: DFCS allows user-configured extraction of data from **Staging** and **Results entities** using **Connectors**.
4. **As Of Date Behavior**: If the **Extraction Entity** contains the **As Of Date** column, the data will be filtered based on the **PMF execution date**.
   - If no date is selected during PMF Execution, the data will be filtered based on the **current date**.
5. **Default Filter for Active Nodes**: The default filter for dimension extraction is "Execution Date between Record Start Date and Record End Date", instead of the "As of Date."
6. **No Need to Provide As Of Date Filter**: Since the PMF execution date is used as the default for extraction, you are **not expected** to provide a filter on the **As of Date** during connector creation.
7. **Finalize**: Follow the necessary steps in Data Extraction based on your specific requirements and configurations.

7

Dimension Loader

The Dimension Loader in the Data Foundation for Banking is a data ingestion tool that loads dimension data (members, attributes, and hierarchies) from staging tables into PBSM interface tables using a PMF pipeline. It supports structured loading via preparation tables and integrates with cloud services where the loaded data can be viewed. The Data File Specification is not applicable to Dimension Data Loaders. The file format and the file names are static in nature.

Purpose of the Dimension Loader

The Dimension Loader performs the following key functions:

- Loads **Dimension Members** and their **Attributes** from the staging area into the Dimension Tables registered with the OFS Cloud Service framework.
- Creates **Hierarchies** for Key Dimensions in the Cloud Service.
- Loads **Hierarchical Relationships** between Key Dimension Members into the Cloud Service.

The following are the features of Dimension Loader:

- Loading Simple Dimensions from Staging Tables.
- Multiple Hierarchies can be loaded from Staging Tables.
- Validations of Members and Hierarchies are similar to that of being performed within the Cloud Service Screens.

Pre-requisites

Before initiating the Dimension Loader:

- Upload the **Data Files** containing dimension details.
- Ensure that each file includes a valid **Dimension Identifier** for both key and simple dimensions.

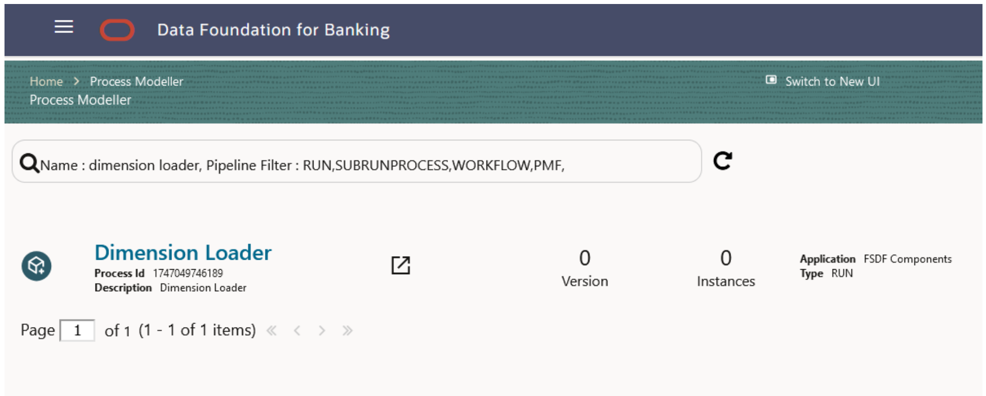
System Components and Workflow

The dimension loading process involves the following stages:

- **Source Systems**: These are the systems where raw and reference data is sourced:
  - **Stage Dimensions Master**: Stores raw dimension data received from upstream transactional or master systems.
- **Dimensions Reference**: Contains metadata for validation and enrichment of staging data.

1. **Step 2: Launch Execution**
   a. Navigate to **Process Modeler** → **Dimension Loader** in the **Data Foundation for Banking** UI.
      i. Open the **Dimension Loader** component.
      ii. Select **Execution Type.**

Figure 7-1   Dimension Loader - Process Orchestration



iii. Enter the following:

- **Object ID** (e.g., DIM_AMH_Load_20250131_1)
- **Application Params**:

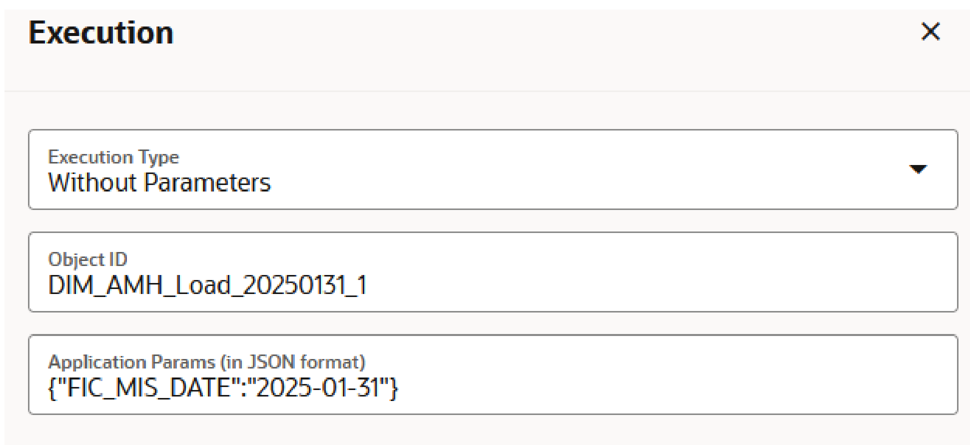
  ```
  json
  ```

  ```
  {"FIC_MIS_DATE":"2025-01-31"}
  ```

Run the component to begin data load for the selected.

```
FIC_MIS_DATE
```

Figure 7-2   Dimension Loader - Execution



2. **ADS Connectors**
   - Extract data from Preparation Tables
   - Apply additional formatting if needed
   - Transfer data securely to PBSM Data Schema
- Reprocess failed loads
- Investigate errors
- Monitor success rates and traceability

- File Format and Specification:
  - Data File Specification is **not required** for Dimension Loader.
  - File **format and names are static** and predefined within the framework.
- Once the ingestion process is completed, start the PMF Execution process. For more information, see PMF Execution Process.

7.1 Execution through PMF

Pre-requisite: Before executing the PMF, you must configure the necessary settings to ensure that the Dimension Loader Attributes are populated with accurate conversion values.

1. Navigate to **Data Foundation for Banking** and click **Data Operations**. The **Process Modeller Summary** page is displayed.
2. In the **Search filter placeholder** bar, select **Pipelines** > **Run Pipeline** > **Dimension Loader**. The available list of Process Id's is displayed.
3. A dropdown is displayed with the following menu items:
   - To access the Process Modeller, click **Process Modeller**.
   - To monitor currently running processes, click **Process Monitor**.
   - To view an overview, click **Dashboard**.

Process Modeller

The Process Modeller is used to create and modify types of Pipelines, test the Process Flow, and execute the Run.

The Process Modeller page displays the existing Business Process Pipelines and Run Pipelines with the details such as Process ID, Process Name, Process Description, Version, Instance, Application, and Last Modified.

You can perform the following tasks from the Process Modeller page:

- Click **Create** to create a new Pipeline.
- Click the Process Name link to launch and edit the Process Flow.
- Click the **Delete** icon to delete a Pipeline.
- Click the **More** icon to view the following menu:
  - Click **View** to see the process flow.
  - Click **Copy** to create a new Pipeline with the same process flow.
  - Click **Process Flow Monitor** to monitor the Pipeline.
  - Click **Execute Run** to execute a Run Pipeline.
- Use the Search grid to search for a specific Pipeline by providing a keyword from the Process ID, Process Name, or Process Description and then click **Search**. Click **Reset** to reset the Search fields.
- You can sort the Pipelines based on Process ID, Process Name, or Application. Click **Columns** and select the required attribute to sort.
- Click the **Process Monitor** icon to launch the Process Monitor page.

Process Monitor

The Process Monitor is used to supervise the current stage of the process for different instances. After integration with an application, you can invoke the workflow. After invoking, the workflow goes through all the stages defined. The Process Monitor displays all the completed stages, current stage, and future stages. Your user group must be mapped to the function role WFMACC (Workflow Monitor Access) to access the Process Monitor page.

This page displays all the workflows, which are invoked from the application with details such as Entity Name, Process Code, Process Description, Execution Start Time, Last Execution Time, Last Updated By, and Status.

You can perform the following tasks on the Process Monitor page:

- Click the **three dots** to view the following submenu :
  - View: To view the process for dimension loader components.
  - Abort: To abort an ongoing Run Pipeline process.
  - Resume: To resume a Run Pipeline process.
  - Re Run: To execute a Run Pipeline process again irrespective of the previous execution status.
- Use the Search grid to search for a specific Pipeline by providing a keyword from the Job ID, Process Name, or Process Description, and then click **Search**. Click **Reset** to reset the Search fields.
- You can sort the Pipelines based on Job ID, Process Name, or Application. Click the Sort by list and select the required attribute to sort.
- You can use the **Search** bar to filter pipelines based on the pipeline type. For example, to view only the Run Pipelines, remove Process from the Filter Pipeline field.
- Click the **Process Modeller** icon to open the Process Modeller page.

To view the details of the selected component, follow the below steps.

1. Select the required process ID to view **DF Components**.
2. Click on the three dots from the DF Components option and click **Properties**. The Activity details pane is displayed.
3. Review the details and view the execution logs.

Execution Logs

To view the Execution Logs for Dimension Loader, follow the below steps.

- Navigate to the Activity Details pane, scroll down and click on **Execution Logs**. The **Logviewer Service** screen is displayed.

> **Note**
>
> The Logviewer Service is used for viewing, monitoring, and analyzing logs generated during a batch job or process execution.

- **Download Button:** Export logs in CSV or other supported formats.
- **Last Refreshed:** Displays the timestamp of the last log refresh.
- **Auto Refresh Toggle:** Enables or disables automatic refresh of the log feed.

- **Batch Run ID:** Unique identifier for the job execution.
- **Process ID:** Identifies the specific subprocess within the batch job.
- **Log Filter Section**:
  - Tabs to filter logs by **Severity Level**: ALL, DEBUG, INFO, WARN, ERROR, FATAL
  - Search bar to quickly locate logs using keywords.
  - **Time Elapsed:** Duration since the job started.
  - **Total Logs:** Count of log entries recorded.
  - **Timezone:** Local time zone of the log timestamps.
  - **Log Frequency Chart:** Visual graph of log volume over time.

7.2 For Key Dimensions

The following is a list of sample files that you can use to build the Dimension Data.

The name of the Data Files must be same as mentioned below and the file extension must be .DAT. The value of column HIERARCHY_CODE must not contain space in files STG_DIMENSIONS_HIER_INTF, STG_HIERARCHIES_INTF and STG_HIERARCHY_LEVELS_INTF. HIERARHY_CODE must be in capital letters.

> **Note**
>
> Any column description that contains a ",", then it should be enclosed within "" (double quotes). For example, Account, Type should be "Account, Type".

To load the Dimensions:

1. Define a new Batch and save it.
2. Add the following Tasks to the above Batch:

Table 7-1 Dimension Loading Process Tasks

Task Code	Task Name	Component	Parameters	Parent Task
1	Stage Dimension Loader	Stage Dimension Loader	All parameters are automatically generated.	
2	Stage Hierarchy Loader	Stage Hierarchy Loader	All parameters are automatically generated.	1

Table 7-1 (Cont.) Dimension Loading Process Tasks

Task Code	Task Name	Component	Parameters	Parent Task
3	Stage DRM Loader	Stage DRM Loader	• **Dimension Name**: Select the relevant one or more dimensions. • **Sync Stage and Dimension**: – **Yes**: The record(s) that is/are already present will be overwritten by the incoming dimension loader. – **No**: The new records will be merged to the existing records. • **Force Member Delete**: This is used only when the above flag is Yes. – **Yes**: This allows you to delete a members even if is referred in hierarchies. – **No**: No records will be deleted.	2

> **Note**
>
> The above Tasks must be executed in the same order.
> The **File to Stage** task must precede the **Stage to Processing** task in a batch.
>
> The Stage DRM Loader allows you to select a Dimension.

3. Execute the Batch.

> **Note**
>
> This method of loading dimensions will be deprecated in future releases. Recommended approach is in the following section.

Dimension Loader with ZIP File Support

To to upload the Dimension data using a zip file:

1. Create a Folder.
2. Follow the naming convention for the DAT files as follows and save them:
   - DIMENSIONS_ATTRIBUTES
   - DIMENSIONS_BASE
   - DIMENSIONS_HIERARCHIES
   - DIMENSIONS_TRANSLATION
   - HIERARCHIES
   - HIERARCHY_LEVELS
3. Place all the DAT files inside the folder.
4. Zip the folder.
5. Create a Batch.
6. Create a Task with the Component name as **Dimension and Hierarchy Loader**.
7. Enter the Zipped Folder Name along with the extension. For example, DIMENSIONS.ZIP.
8. Execute the Batch.

For detailed instructions on Creating a Batch, Defining a Task, Execute the Task, and Schedule a Batch, see Scheduler Services.

7.3 For Simple Dimensions

The following is a list of sample files that you can use to build the Dimension Data.

The name of the Data Files must be same as mentioned below and the File Extension must be .DAT.

> **Note**
>
> Any column description that contains a ",", then it should be enclosed within "" (double quotes). For example, Account, Type should be "Account, Type".

To load the Dimensions, perform the following steps:

1. Define a new Batch and save it.
2. Add the following Tasks to the above Batch:

Table 7-2 Dimension Loading Process Tasks

Task Code	Task Name	Component	Parameters	Parent Task
1	Stage Dimension Loader	Stage Dimension Loader	All parameters are automatically generated.	
2	Stage DRM Loader	Stage DRM Loader	• **Dimension Name**: Select the relevant one or more dimensions. • **Sync Stage and Dimension**: – **Yes**: The record(s) that is/are already present will be overwritten by the incoming dimension loader. – **No**: The new records will be merged to the existing records. • **Force Member Delete**: This is used only when the above flag is Yes. – **Yes**: This allows you to delete a members even if is referred in hierarchies. – **No**: No records will be deleted.	1

> **Note**
>
> The above Tasks must be executed in the same order.
> The **File to Stage** task must precede the **Stage to Processing** task in a batch.
>
> The Stage DRM Loader allows you to select a Dimension.

3. Execute the Batch.

> **Note**
>
> This method of loading dimensions will be deprecated in future releases.
> Recommended approach is in the following section.

Dimension Loader with ZIP File Support

To to upload the Dimension data using a zip file:

1. Create a Folder.
2. Follow the naming convention for the DAT files as follows and save them:
   - DIMENSIONS_BASE
   - DIMENSIONS_TRANSLATION
3. Place all the DAT files inside the folder.
4. Zip the folder.
5. Create a Batch.
6. Create a Task with the Component name as **Dimension and Hierarchy Loader**.
7. Enter the Zipped Folder Name along with the extension. For example, DIMENSIONS.ZIP.
8. Execute the Batch.

For detailed instructions on Creating a Batch, Defining a Task, Execute the Task, and Schedule a Batch, see Scheduler Services.

7.4 Clear and Back up Dimension Data

This process helps you to clear or delete the existing Dimension Data from the relevant Dimension tables using the Scheduler Services.

Before clearing the data from the Dimension tables, the service creates a back up of the table.

To clear the Dimension Data:

1. Navigate to **Operations and Processes**, select **Scheduler**, and then select **Define Batch**.
2. Create the Batch and save it.
3. Navigate to **Operations and Processes**, select **Scheduler**, and then select **Define Task**.
4. Select the created Batch and create a Task with **Clear Dimension Members** as Component.
5. Select the Dimension Name that you want to delete. You can select one or more Dimension Names.

6. From the LHS menu, navigate to **Operations and Processes**, select **Scheduler**, and then select **Execute Batch**.
   The service first creates a backup of the existing Dimension table and then deletes the Dimension Member entries for the selected Dimensions.

While deleting the data from the tables, there are no validations. The back up files are suffixed with As Of Date and Current Time Stamp.

The following table gives the sample actual and backed up table names:

Table 7-3 Sample Table Names

Dimension Type	Actual Table Name	Backup Table Name
Simple Dimension	FSI_ACCRUAL_BASIS_CD	FSI_ACCRUAL_BASIS_CD_<AS_OF_DATE>_<CURRENTTIMESTAMP>
	FSI_ACCRUAL_BASIS_MLS	FSI_ACCRUAL_BASIS_MLS_<AS_OF_DATE>_<CURRENTTIMESTAMP>
Key Dimension	DIM_COMMON_COA_ATTR	DIM_COMMON_COA_ATTR_<AS_OF_DATE>_<CURRENTTIMESTAMP>
	DIM_COMMON_COA_B	DIM_COMMON_COA_B_<AS_OF_DATE>_<CURRENTTIMESTAMP>
	DIM_COMMON_COA_HIER	DIM_COMMON_COA_HIER_<AS_OF_DATE>_<CURRENTTIMESTAMP>
	DIM_COMMON_COA_TL	DIM_COMMON_COA_TL_<AS_OF_DATE>_<CURRENTTIMESTAMP>

7.5 Data Preparation Guidelines

While creating the data files, ensure the following:

- Ensure the data files are in TXT, DAT, or CSV formats.
- Ensure that there are no duplicate records in a data file.
- Data file names are in the prescribed format.
- Use only comma (,) and/or pipeline (|) as delimiters.
- Only double quotes (“”) are used as Field Enclosures.
- The language code must be as per BCP 47 format. For example, **en-US**.
- In the file for hierarchies, there must be no empty space or special characters for HIERARCHY_CODE.
- Dimension member name must not contain **&** character.

8
Data Pipelines

Refer to Data Operations Guide for more details.

9

Key Terms and FAQs

Table 9-1 Key Terminologies

Term	Description
DFCS	**Data Foundation Cloud Service** - acts as the Publisher
PBSMCS	**Profitability and Balance Sheet Management Cloud Service**- acts as the Subscriber
ADS	**Application Data Services** that handle metadata exchange between tenants
PMF	**Process Modeller Flow** that enables data movement post-subscription
Wallet	Secure credentials used for establishing tenant subscriptions
BT	In OCI applications, the Business Term is applied to support and standardize DFCS integration workflows.