Oracle® Financial Services Data Foundation Cloud Service for Banking

Data Operations Guide

Release 25A
G31033-01
April 2025

ORACLE

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

G31033-01

Contents

1

Data Integration

Data Integration is a core service within OFSAA that enables robust and seamless data exchange between OFSAA applications and external systems. It abstracts the underlying **Financial Services Data Foundation (FSDF)** and exposes it through **Application Data Interfaces (ADIs)**, simplifying the process of data ingestion, transformation, and extraction.

Data Source Templates: These templates help define external data sources that are to be integrated into OFSAA.

- Standardized **metadata** definitions
- Consistent **mappings** for seamless ingestion
- Support for various **input formats**

Example: If integrating an external banking system, a Data Source Template ensures that customer account fields are properly mapped to the corresponding FSDF attributes.

- **Connectors**: Connectors establish the **technical connection** between OFSAA and external sources such as:
  - **Databases** (e.g., Oracle, SQL Server)
  - **Flat files** (e.g., CSV, XML)
  - **APIs** (REST/SOAP)
  - They define:
    - **Connection parameters** (e.g., host, port, credentials)
    - **Supported protocols and data formats**
    - **Pre-load behavior and mappings**
      Example: A connector can be configured to pull investment data nightly from an external data warehouse via SFTP.
- **Data Operations**: This module provides a graphical or script-based interface for creating **end-to-end data workflows**.
- Core capabilities include:
  - Data **transformation** (e.g., cleansing, enrichment)
  - Process **orchestration**
  - Support for **batch** or **real-time** execution
  - Error handling and **monitoring**
    Example: A data operation may transform raw transaction feeds by aggregating balances and enriching them with customer metadata before loading into FSDF.
- **Data Extraction**: This service allows users to extract data **from** the OFSAA data foundation for:
  - External **reporting**
  - **Analytics** tools integration (e.g., Power BI, Tableau)
  - Feeding **regulatory systems**

Supported methods:

- **Scheduled** extracts
- **API-based** pulls
- Custom **scripts or jobs**

Example: Extracting the daily balance sheet summary to a flat file for upload into a central reporting system.

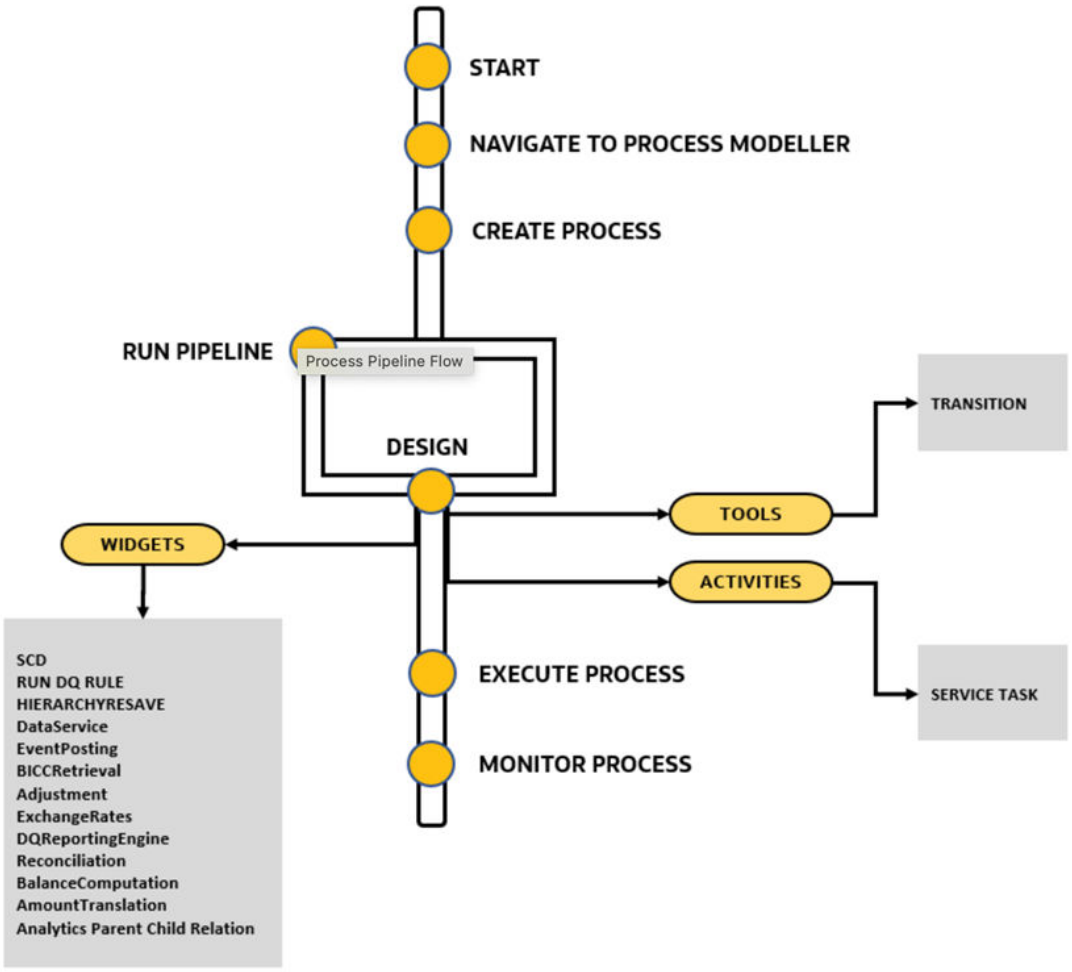
1.1 Key Features of Data Integration

- Support for visual modeling of pipelines
- Provision of seeded pipelines with ability to modify them
- Ability to create new Run pipelines
- Support for registration of process, activity, and implementing transition logic separated from the modeling itself
- Built-in Orchestration Engine for task execution
- Provision of multiple components and canvas to visually stitch the DFCS components to form the Run pipeline
- Ability to monitor process during all stages of the execution process

1.2 Process pipeline flow

The Process Pipeline Flow diagram illustrates an end-to-end flow of designing, executing, and monitoring a data process pipeline using a visual Process Modeller. It provides a structured and intuitive approach for building workflows in data integration and reconciliation environments.

Figure 1-1 Process Pipeline Flow



2
Data Operations

Data Operations is a design and execution service that empowers process pipeline developers to implement pipelines based on models created by business analysts. This framework enables developers to orchestrate **Run Pipelines** within DFCS and design the artifacts required for these pipelines.

The service includes two essential components:

- **Process Modeler**: A tool for designing data pipelines, it helps represent the necessary artifacts and provides implementation details for DFCS process artifacts.
- **Process Monitor**: A monitoring tool that tracks and manages the execution of instantiated pipelines within DFCS.

2.1 Key Features of Data Operation

- Support for visual modeling of pipelines
- Provision of seeded pipelines with ability to modify them
- Ability to create new Run pipelines
- Support for registration of process, activity, and implementing transition logic separated from the modeling itself
- Built-in Orchestration Engine for task execution
- Provision of multiple components and canvas to visually stitch the DFCS components to form the Run pipeline
- Ability to monitor process during all stages of the execution process

2.2 Access Data Operations

To view or manage Data Operations within the **Data Integration** module, follow these steps:

1. **Log in** to the DFCS application.
2. From the **Home screen**, locate the **Data Integration** tile or menu.
3. Under **Data Integration**, click on the **Data Operations** section (as shown in the image).
4. You will be directed to the **Data Operations** workspace, where you can:
    - View existing operations
    - Create new data workflows
    - Manage, edit, or monitor operational processes

3
Process Modeler

The **Process Modeler** page provides a comprehensive overview of existing pipelines along with key details such as process ID, process name, description, version, instance, application, last modified date, and the user who last modified it.

On this page, you can perform the following actions:

1. **Create a New Pipeline**: Click the **+** icon to design a new pipeline.
2. **Edit an Existing Pipeline**: Click the process name link to open and modify the pipeline.
3. **Launch a Process**: Open the pipeline in a new window.
4. **Delete a Pipeline**: Remove an unwanted pipeline.
5. **Access the Menu Options** for individual pipelines to:
6. View the process flow.
7. Copy the process flow.
8. Monitor the pipeline in the **Process Flow Monitor** window.
9. Execute the pipeline.
10. Apply filter conditions to a **Run Pipeline**.

Additional functions include:

- **Search**: Use the search grid to find pipelines by entering keywords such as process ID, process name, or description. The reset icon clears the search fields.
- **Pipeline Filters**: Refine search results using the **Pipeline Filter** options. For instance, to display only Run Pipelines, select **Run Pipeline** in the filter and remove other types if selected.
- **Sorting**: Sort pipelines by process ID, process name, or application using the **Sort By** drop-down.

Navigation to Process Monitor: Easily access the **Process Monitor** window for detailed monitoring.

4

Process Flow

The Process Flow canvas is accessible by clicking the process name or the launch icon.

The Process Flow window consists of a toolbar and a canvas. The canvas is used for designing process flows, while the toolbar on the left-hand side contains the following tabs:

- Activity
- Transition
- Widget

You can drag and drop these components onto the canvas, connect them, and configure each component to create and design your process flow.

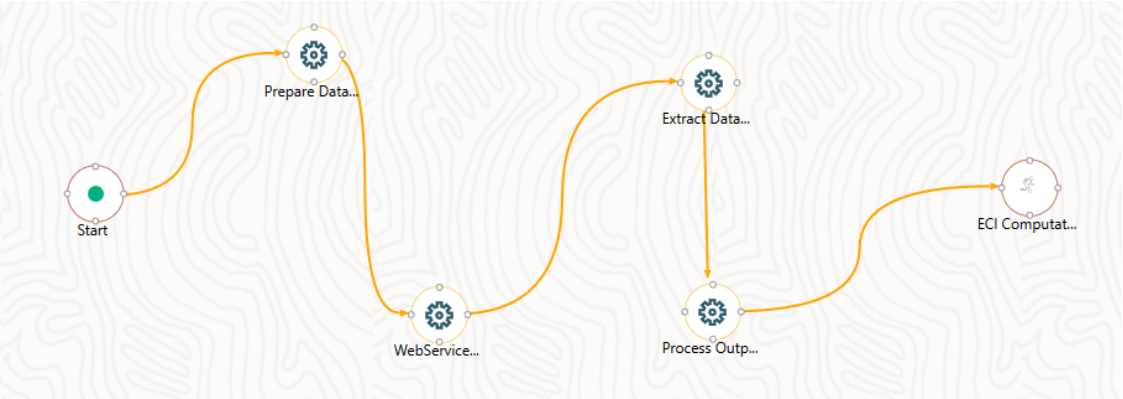
5
Activity

Service Task

The **Service Task** is available under the Activity toolbar and is used to invoke an application component, such as triggering a business rule to calculate a specific threshold. It is an automated task that gets activated within the process flow and is commonly utilized to execute business logic defined by an Execution Rule type application rule.

Service Tasks are used to invoke an External Model Service through stored procedures and functions.

Figure 5-1 An Example: Service Tasks Flow



5.1 How to Use a Service Task

1. Expand the Activity section in the toolbar on the left, then click and drag the **Service Task** icon onto the canvas.
2. Double-click the **Service Task** component on the canvas to open the **Job Configuration** window.

5.1.1 Job Configuration

Job Configuration refers to the process of defining and setting up the parameters, settings, and resources required for a specific job or task to be executed within a system. This configuration ensures that the job runs according to specified requirements, including input data, processing steps, dependencies, and output handling. Job configurations are essential for automating tasks, managing workflows, and ensuring that jobs execute in a controlled and efficient manner within a larger process or system.
For more information, see How to Use Parallel Gateways.

5.1.1.1 Activity Tab

1. **Activity ID** is auto-populated and you can mouse-over the **i** icon to view it.
2. By default, the activity ID is populated in the name field. You can enter your own **Activity Name** and **Activity Description**.
3. The **Activity Type** is selected by default depending on the Activity ID.
4. Select **Status** to determine the current state of the activity (e.g., Pending, In Progress, Completed). It helps track the activity's progress.
5. Select **Outcomes** to specify the potential results or conditions of the activity. It determines the next steps based on the activity's completion (e.g., Success, Failure, Skipped).
6. Click the check mark icon to save.
7. Click anywhere outside the configuration window to close it.

5.1.1.2 Implementation Tab

1. Click the **Implementation** tab on the configuration window.
2. Click the search icon to select the **Execution Rule** that has to be executed for this activity. The **Participant Details** window is displayed with all **Application Rules** of type **Execution Rule** available in your process.
   - Click the name link of the Application Rule to view its details.
   - Select the required rule and click the **check mark**.
   - Click outside the window to return to the configuration window.
3. Add **Parameters** you want to pass to the Execution Rule.
   - Click **+** next to **Parameters**. The **Participant Details** window is displayed.
   - Select the **Data Fields** to which you want to pass the value. This list displays all data fields for the current process or package.
   - Select the **Parameter Type**. Choose **Static** if you want to pass a static value to the selected data field. Enter the **Value**. Choose **Dynamic** to pass the value during the execution of the workflow.
   - The parameters you added are displayed under the **Parameters** in the **Implementation** tab.
4. Select the **Pre Rule**, which is the Application Rule you want to execute before executing the Execution Rule. Click the corresponding search icon to select the Application Rule that you want to set as the pre rule.
5. Select the **Post Rule**, which is the Application Rule you want to execute after executing the Execution Rule. Click the corresponding search icon to select the Application Rule that you want to set as the post rule.
6. Click **Save**.
7. Click anywhere outside the configuration window to close it.

5.1.1.3 Transition

The Transition component is used to control the flow between various components in the Process Flow. Transition connects two activities and the flow is configured based on a Conditional Expression or Decision Rule.

To configure Transition details, follow these steps:

1. Click the **Transition** tab.
2. Click the **+** icon to display the **Add New Transition from Node** window.
3. Select the required job to be connected to from the **Connected To** drop-down.
4. Enter a name for the transition in the **Transition Name** field.
5. Enter the order of execution in the **Order** field. For example, 1 marks the order as to be executed first.
6. Click the **Search** icon from the **Decision Rule** drop-down and select the required application rule.
7. Depending on the Execution Type, enter the required details.
8. Click the **check mark** icon to save the details.
9. Select the stroke format from the **Stroke** drop-down.
10. After entering the details, click the **check mark** icon to save the entered details.

5.1.1.4 Action

- **Access Action Tab**: Click on the **Action** tab on the configuration window.
- **Set Group Approvals and Routing Policy**: To configure the **Group Approvals** and **Routing Policy**, click the **Group Approval** button.
- Select the Voting Policy depending on the selected Routing Policy.
- The available routing policies are Parallel and Sequential.
- **Add Options**: Click the **+** icon to open the available options for configuring the policy. These options include:
  - **Task Details**: Allows you to define the tasks and parameters related to the group approval. To select an appropriate **Condition**, refer to the **Application Rule** section for more information.
  - **Expiry**: Lets you set an expiration date for the task or approval.
  - **Escalation**: Provides the option to escalate the task if it is not completed within a certain timeframe.
  - **Email**: Option to send emails regarding the task, such as notifications or updates.
  - **Reminder**: Allows you to set a reminder for follow-ups or pending actions.

5.1.1.5 Notification

You can set up notifications by configuring the following parameters:

1. **Access the Notification Group Configuration**: Navigate to **Activity Details > Notification Group**
2. **Review or Enter Notification ID and Name**
   - **Notification ID:** Auto-generated
   - **Notification Name:** Enter a descriptive name
3. **Define the Condition (Optional)**
   - Click in the **Condition** box.
   - Optionally, input a condition that determines when the notification should be triggered.

- Use the search icon to look up available conditions if needed.

4. **Select When to Generate the Notification**
   - In the **Generate** dropdown, choose from:
     - On Entry
     - On Stage
     - On Exit
     - On Escalation
     - On Reminder
5. **Set Notification Status**
   - **Enabled:** Choose **Yes** or **No** to enable or disable the notification.
6. **Configure Email Settings (Optional)**
   - **Email Required:** Choose **Yes** if an email should be sent.
     - If **Yes** is selected:
       * Select an **Email Template** from the dropdown:
         * Global Task Template
         * Default Task Template
7. Click the **checkmark** icon at the bottom right to save the configuration.

6
Transition

Transitions are commonly modeled using **gateways** (such as **parallel**, **sequential**, or **exclusive**) to split or merge paths based on certain conditions or events.

6.1 Gateways

A **gateway** is a decision point in a process flow where the flow can diverge or converge, directing the flow of execution based on certain conditions or rules. Gateways are used to control the path that process flows take depending on the logic of the process. They are essential for splitting or merging transitions in a workflow.

There are several types of gateways, each serving different purposes:

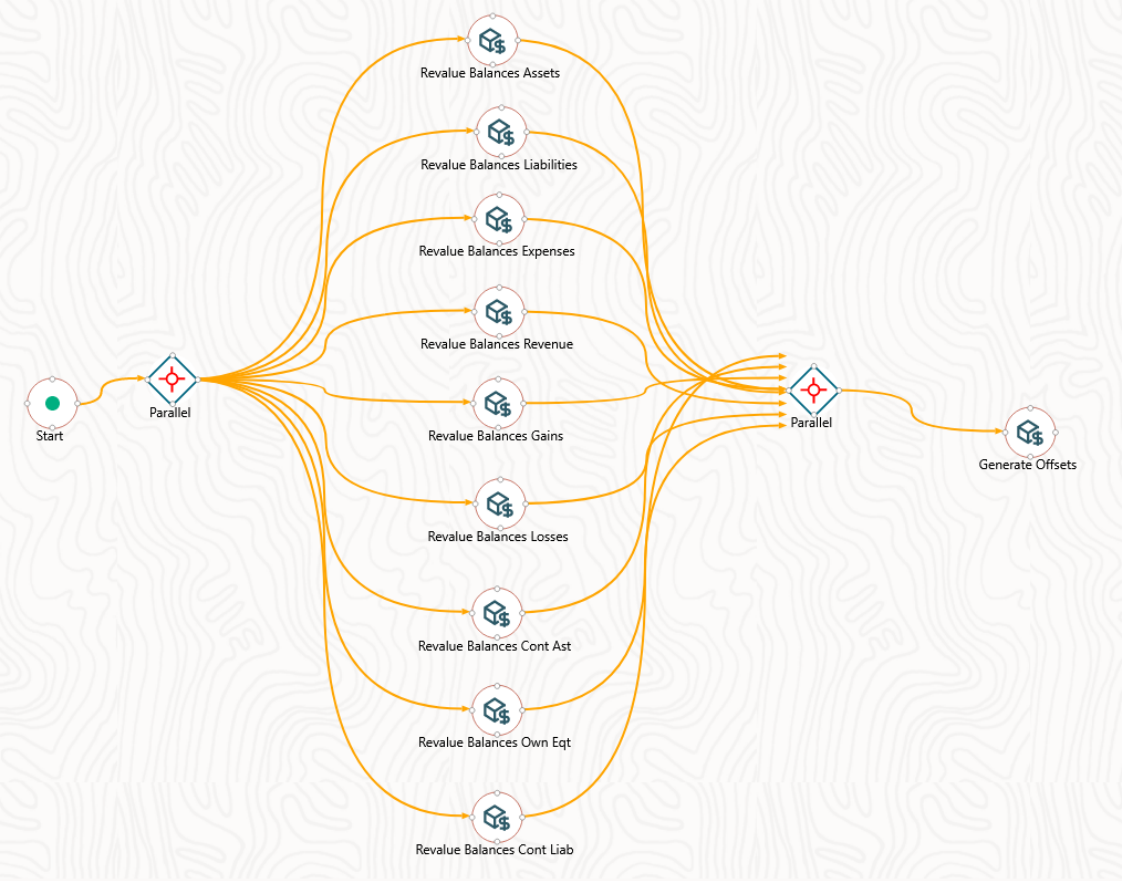
- **Parallel Gateway**: This type of gateway is used when multiple transitions or flows need to be executed simultaneously. It ensures that all outgoing flows are triggered at the same time, allowing parallel execution.
- **Sequential Gateway**: This gateway is used when multiple transitions or flows need to be executed in a specific, sequential order. The outgoing flows are triggered one after the other, following a predefined sequence.
- **Multi Choice Gateway**: A Multi-Choice Gateway is used when different transitions or flows are chosen based on a decision rule. It allows the process to branch into multiple paths depending on the conditions or rules applied at the gateway.
- **Connector**: A Connector helps connect two activities with a different path (other than the default), in case the default path is overlapping an existing flow.

> **Note:**
>
> If you use a Parallel or Multi Choice Gateway in your pipeline, ensure that after all the activities are added to these gateways, they are merged or closed again with a Parallel or Multi Choice Gateway, respectively. If there is no more activity to be performed after the Parallel or Multi Choice Gateway, it is mandatory to add an empty service task activity. Otherwise, the status is not updated correctly and the next activity execution does not happen.

6.1.1 Executing Parallel Tasks

A Parallel Gateway is used to execute multiple tasks in parallel. In the usual flow, tasks are executed sequentially.

Figure 6-1 Executing Parallel Tasks flow



In the example shown, when the Parallel Gateway is invoked, all the OFSAA components that are placed between Parallel Gateways, are executed simultaneously. After all components are executed, the execution moves to the next activity in the Process Flow.

> **Note:**
>
> In case if any one of the task fails as part of parallel gateway, the other ongoing parallel tasks executions are completed. The status of the activities get updated appropriately even though one or more activities fail.

Table 6-1 PMF Run Pipeline Design

Do's for a Gateway (Parallel and Multichoice)	Dont's for a Gateway (Parallel and Multichoice)
Ensure to begin and end with the same gateway during run pipeline design.	A parallel gateway cannot exist within another parallel gateway.
Ensure to add a service task if a parallel gateway is the last executing activity.	Multiple transitions are not supported for any activity in a run pipeline. Therefore, you must only use a parallel gateway.
Tasks which start from a parallel gateway must end in the successive closing parallel gateway.	

Table 6-1 (Cont.) PMF Run Pipeline Design

Do's for a Gateway (Parallel and Multichoice)	Dont's for a Gateway (Parallel and Multichoice)
Execution of a pipeline will begin from the **Start** activity. Therefore, the transition cannot be stopped for this activity.	

6.1.2 How to Use Parallel Gateways

This section explains how to design parallel tasks using Parallel Gateways. For example, executing the Reconciliation widget in parallel for different rules.

1. Create a process in the Process Modeler canvas.
2. On the **Process Flow** page, you will see a **START** activity is already present on the canvas. This activity indicates the beginning of the process.
3. From the LHS menu, under **Transition**, click and drag a **Parallel Gateway** widget on to the canvas.
4. Connect the **START** activity with this **Parallel Gateway** widget.
5. Drag and drop the widgets representing the tasks that run in parallel on to the canvas. Example: add multiple Reconciliation widgets on to the canvas.
6. Connect the Parallel Gateway widget with each of these widgets representing tasks that must run in parallel. This represents the beginning of the transition.
7. Drag and drop another Parallel Gateway widget on to the canvas.
8. Connect each of the Reconciliation widgets to this newly added Parallel Gateway widget. This represents the end of the transition.
9. Configure the Reconciliation widgets based on your requirement.
10. Continue following this process to create your pipeline, which may include one or more parallel tasks.
11. Click **Save** to finalize the changes.

7
Widgets

Widgets are used to execute Connectors, Data Service, and Event Posting.

7.1 How to Use a Widget

1. On the **Process Flow** page, expand the toolbar on the left, click and drag the required widget icon under **Widget** on to the canvas.
2. Double-click the widget to open the configuration window for the widget. The **Activity** tab is selected.
3. Enter the **Activity Name** and **Activity Description**.
4. Specify the **Dynamic Parameters** for the widget.

7.1.1 Dynamic Parameters for Widgets

- Transform Data
- BalanceComputation
- AmountTranslation
- OBIEEPCRELATION
- RUN DQ RULE
- PeriodDayYTDBalance
- RetrieveRevalConfig
- DFComponents
- HIERARCHYRESAVE
- Connector
- DataService
- DQReportingEngine
- Reconciliation
- SCD (Slowly Changing Dimensions)

7.1.2 Transform Data

This activity is responsible for applying transformation rules to the data extracted from the specified datastore. It uses defined business rules to cleanse, enrich, or format the data in preparation for downstream processing or reporting. The transformation ensures that the data conforms to the required standards and is aligned with the system's data quality objectives.

1. Navigate to the **TRANSFORM DATA** configuration panel.
2. **Fill in the Activity Name and Description**
   - **Activity Name**: Enter a name for the activity.

- **Activity Description**: Optionally add a description to clarify the activity's purpose.

3. **Set Dynamic Parameters**
   - **Datastore Name**: Select from the dropdown.
   - **Rule Name:**

     Choose a rule from the dropdown list. This defines the transformation logic.
   - **Parameter List**: Enter or select the relevant parameters used by the transformation rule (e.g., filter values, limits, formats). This may be manual input or dynamically populated based on the rule.

> **Note:**
>
> Click on the Transition and Notification tabs to enter further details. For more information on Transition and Notification, see Transition and Notification.

4. Click the **checkmark** icon at the bottom right of the screen to save your activity.

7.1.3 BalanceComputation

This screen allows you to associate and manage notification rules and participants with the Balance Computation activity. Notifications define the conditions under which alerts are triggered, while participants determine who receives the alerts or is involved in the process. You can configure multiple notification rules and assign relevant participants to ensure timely communication and action.

- **ID:** Links to the notification job
- **Name:** Custom name of the notification
- **Condition & Generate:** Describe when and how the notification should be triggered
- Click **+** icon to define the parameters.
- **ID:** System-generated unique identifier for the notification entry (default is 0 for new entries).
- **Decision Rule:** Logic that determines whether the notification should be sent.
- **Role Code:** Defines the functional or organizational role for filtering the recipient (optional).
- **User Type:** Specifies how the recipient(s) will be selected:
  - **None** – No recipients will be added.
  - **User** – A specific user is selected manually.
  - **Rule** – Recipients are selected based on a predefined rule.
  - **User Group** – Notification will be sent to all members of a selected group.

7.1.4 AmountTranslation

The **AmountTranslation** activity is used to convert monetary amounts from one currency to another within a financial processing workflow. This setup allows defining the activity name, status, and any additional comments needed for tracking or configuration purposes.

- **Activity Name:** The label or identifier for this specific translation task.
- **Activity Description:** Optional text to describe the purpose or scope of this activity.

- **Activity Type:** Predefined and fixed as AmountTranslation to denote this function.
- **Status:** Dropdown to set the current operational state (e.g., Active, Inactive, Pending).
- **Dynamic Parameters – Comments:** Field to capture any additional remarks or notes relevant to this translation activity.

7.1.5 OBIEEPCRELATION

This widget defines relationships within business intelligence or data hierarchy structures.

1. In the **Activity Name**, enter the Name. For example: OBIEEPCRELATION.
2. (Optional) Enter a description in **Activity Description** field.
3. Under **Dynamic Parameters**, open the **Dimension Hierarchy Entity** dropdown.
4. Choose one of the following:
   - General Ledger Hierarchy Dimension
   - Legal Entity Hierarchy Dimension
   - Line Of Business Hierarchy Dimension
   - Organization Unit Hierarchy Dimension
   - Account Hierarchy Dimension
5. Click the **checkmark** to save.

7.1.6 RUN DQ RULE

To configure this widget, complete the following steps:

1. On the **Process Flow** page, expand **Widget** section in the LHS menu and drag this widget on to the canvas.
2. Double-click the widget on the canvas to open the configuration window with the **Activity** tab selected.
3. Under **Dynamic Parameters**, configure the settings described in the below table.
4. Select the check mark icon to save your changes. To close the configuration window, click anywhere outside the window on the canvas.

Table 7-1 Dynamic Parameters for RUN DQ RULE Description

Property	Description
Datastore Name	Refers to the name of the Information Domain. By default, the Information Domain to which you are connected is selected.
DQ Group Name	Select the required DQ group. The unique identifier name for the groups. The name should start with alphabet and should not be more than 50 characters.

Table 7-1 (Cont.) Dynamic Parameters for RUN DQ RULE Description

Property	Description
Rejection Threshold	Enter the maximum allowable number of errors, in absolute value, that a Data File can contain while still marking the Data Load as successful. This setting is part of the property specification for the "Load to Table" option. If the number of erroneous records exceeds the Rejection Threshold value, the data loading task will fail, and the inserted values will be rolled back for that table. Inserts for previous tables will not be reverted. The Rejection Threshold is applied to each target table individually within a batch. By default, the Rejection Threshold is considered as 100 when no input is added. When the Rejection Threshold is set as 100 or blank, the whole data file will be loaded irrespective of the number of errors.

Note:

The Rejection Threshold value should be a number which is considered as percentage of records. For example, if you have entered the value as 10, it is considered as 10 % of the entire records.

Note:

The Threshold limit is not applicable for duplicate Data Quality Check.

Table 7-1 (Cont.) Dynamic Parameters for RUN DQ RULE Description

Property	Description
Additional Parameters	Enter any additional parameters for the Run DQ Rule filtering criteria in the following format: Key#Data type#Value; Key#Data type#Value; and so on. **Note:** If these additional parameters are not specified, the default value will be set to NULL. Except for the standard placeholders $MISDATE and $RUNSKEY, all additional parameters for DQ execution must be enclosed in single quotes. For example: `STG_EMPLOYEE.EM` P_CODE = '$EMPCODE'
Fail if Threshold Breaches	By default, the **Fail If Threshold Breaches** option is set to **TRUE** when no input is provided. If the threshold is breached and the **Fail If Threshold Breaches** field is set to TRUE, the job will abort, and the failure records will not be inserted into the DQ Result tables. If the threshold is breached and the **Fail If Threshold Breaches** field is set to **FALSE**, the job will continue, and the failure records will be inserted into the DQ Result tables.
Stop Insert on Threshold Breach	This is applicable only when **Fail if Threshold Breaches** is set to **FALSE**. When **Stop Insert on Threshold Breach** is set to **Y**, only the invalid records based on rejection threshold are inserted in the DQ Result tables. If set to **N**, all the invalid records are inserted. By default, it is set to **N**.
Rule Execution Connection	To establish a connection for DQ execution, enter the Rule Execution Connection as **STAGE**.
Result Store Connection	To store the Data Quality execution results, enter the Result Store Connection as **PREPROCESS**.
Batch on Fail	Leave this parameter blank.
Micro service Id	Enter the Micro Service ID.

7.1.7 PeriodDayYTDBalance

Retrieves and calculates the Year-To-Date (YTD) balance by specific period and day, supporting financial reporting and trend analysis.

1. On the **PeriodDayYTDBalance** activity window, enter the Required Fields **Activity Name**.
2. Enter a brief description of the activity's purpose in the **Activity Description** field.
3. **Dynamic Parameters**: This section handles any input or configuration that is dynamically required by this activity.
4. **Comments**: Add any relevant notes or information related to the activity.
5. Once all fields are complete, click the **checkmark** icon (bottom-right corner) to save the configuration.

7.1.8 RetrieveRevalConfig

Fetches revaluation configuration settings, including currency rules, frequency, and threshold parameters for financial revaluation processes.

1. Navigate to the activity screen where the **Activity Name** is set to

   ```
   RetrieveRevalConfig
   ```

   .
2. **Enter the Activity Name (if not already filled)**
   - **Field:** Activity Name
   - Already pre-filled with RetrieveRevalConfig. You can leave it as-is unless you're customizing it.
3. Activity Description: Add a meaningful description that explains the purpose of this activity.
4. Add Any Dynamic Parameters or Notes in the **Comments** field.
5. Click the **checkmark** icon at the bottom-right to save the activity.

> **Note:**
>
> Hover over or click the icon next to the *Activity Name* for definition (if available).

7.1.9 DFComponents

Retrieves or processes data foundation components such as dimensions, hierarchies, or metrics used for reporting and analytics.

1. **Open the Activity Configuration Screen**: This screen is for setting up the **DFComponents** activity.
2. **Activity Name**: It is a defaulted field. You can modify it as needed.
3. **Activity Description**: Enter a clear explanation of what this activity does.
4. **Comments (Optional)**: You can add notes for future reference, testing instructions, or integration notes.

5. Click the **checkmark** icon in the bottom-right corner to confirm and save the setup.

> **Note:**
>
> Click the icon next to *Activity Name* if you need suggestions or tooltips.

7.1.10 HIERARCHYRESAVE

Hierarchy Resave is the process of updating the hierarchy's data with either a complete replacement of existing data (Resave) or adding new data into the current hierarchy structure (Refresh). This ensures that the data remains accurate and up-to-date.
To configure this widget, complete the following steps:

1. On the **Process Flow** page, expand the **Widget** section in the left-hand menu. Then, drag and drop the widget onto the canvas.
2. Double-click the widget on the canvas to open the configuration window. The **Activity** tab will be selected by default.

Table 7-2 Widget Table

Field	Description
Activity Name	Enter a name for the activity in the **Activity Name** field (e.g., "HIERARCHYRESAVE").
Activity Description	Enter a description for the activity in the **Activity Desc** field. This helps to clarify the purpose or function of the activity.
Activity Type	Ensure the activity type is set to **HIERARCHYRESAVE** as shown in the image. This is typically auto-filled based on the widget type.
Status	Select the status for activity type.

3. In the **Dynamic Parameters** section, configure the settings as described in the table below.

Table 7-3 Dynamic Parameters for HIERARCHYRESAVE

Property	Description
Entity	Select one or more entities. Use the **Include All** icon to include all entities, or the **Exclude All** icon to exclude all entities
Hierarchy	Select one or more hierarchies. Use the **Include All** icon to include all hierarchies, or the **Exclude All** icon to exclude them.
Load type	Select the Load Type: • Resave - Existing data is replaced with the freshly populated data. • Refresh - Freshly populated data is added to the existing data.

4. Once you've configured the necessary settings, click the checkmark icon to save your changes. To close the configuration window, simply click anywhere outside the window on the canvas.

7.1.11 Connectors

To configure this widget, complete the following steps:

1. On the **Process Flow** page, expand **Widget** in the LHS menu, then drag and drop the widget onto the canvas.
2. Double-click the widget on the canvas. The configuration window will appear with the **Activity** tab selected.
3. Under **Dynamic Parameters**, configure the settings as outlined in the table below.

Table 7-4 Dynamic Parameters for Connectors

Property	Description
Connector Name	Select the required connector from the drop-down list. The selected connector loads the data as mapped in the staging tables of the ledger into DFCS.
Variables	Enter the required variables.

4. Click the check mark icon to save your changes. To close the configuration window, click anywhere outside the window on the canvas.

7.1.12 DataService

The **DataService** widget is used to configure and manage data-related activities within a process flow. It allows users to define and execute operations involving data services.

To configure this widget, complete the following steps:

1. On the **Process Flow** page, expand the **Widget** section in the left-hand menu. Then, drag and drop the widget onto the canvas.
2. Double-click the widget on the canvas to open the configuration window. The **Activity** tab will be selected by default.
3. In the **Dynamic Parameters** section, configure the settings as described in the table below.

Table 7-5 Widget Table

Field	Description
Activity Name	Enter a name for the activity in the **Activity Name** field (e.g., "HIERARCHYRESAVE").
Activity Description	Enter a description for the activity in the **Activity Desc** field. This helps to clarify the purpose or function of the activity.
Activity Type	Ensure the activity type is set to **HIERARCHYRESAVE** as shown in the image. This is typically auto-filled based on the widget type.
Status	Select the status for activity type.

Table 7-6 Dynamic Parameters for DataService

Property	Description
Variables	Enter the required variables.
DataService Name	Select the required Data Service from the drop-down list.

7.1.13 DQReportingEngine

The **DQReportingEngine** is a component or tool used for generating and managing Data Quality (DQ) reports. It processes data to assess its quality, identifies issues, and produces reports that help users monitor and improve the quality of their data.

To configure this widget, complete the following steps:

1. On the **Process Flow** page, from the LHS menu, expand **Widget** and drag and drop this widget on to the canvas.
2. Double-click the widget on the canvas. The configuration window is displayed with the **Activity** tab selected.

Table 7-7 Widget Table

Field	Description
Activity Name	Enter a name for the activity in the **Activity Name** field (e.g., "HIERARCHYRESAVE").
Activity Description	Enter a description for the activity in the **Activity Desc** field. This helps to clarify the purpose or function of the activity.
Activity Type	Ensure the activity type is set to **HIERARCHYRESAVE** as shown in the image. This is typically auto-filled based on the widget type.
Status	Select the status for activity type.

3. Under **Dynamic Parameters**, configure the settings described in the following table.

Table 7-8 Dynamic Parameters for DQReportingEngine

Property	Description
DQ Reporting Rule	Select the DQ Reporting Rule from the Run DQ rule job that you wish to use for validation of the data.

4. Click the check mark icon to save your changes. To close the configuration window, click anywhere outside the window on the canvas.

> **Note:**
>
> Post execution, you should perform a fresh execution from the PMF pipeline. It is not recommended to perform a re-run operation once the execution is complete.

7.1.14 Reconciliation

The **Reconciliation** widget is used to configure and automate the process of reconciling transactions and generating statements. The configuration ensures that financial or transactional data is accurately compared and balanced.

To configure this widget, complete the following steps:

1. On the **Process Flow** page, expand the **Widget** section in the left-hand menu. Then, drag and drop the widget onto the canvas.
2. Double-click the widget on the canvas to open the configuration window. The **Activity** tab will be selected by default.

Table 7-9 Widget Table

Field	Description
Activity Name	Enter a name for the activity in the **Activity Name** field (e.g., "HIERARCHYRESAVE").
Activity Description	Enter a description for the activity in the **Activity Desc** field. This helps to clarify the purpose or function of the activity.
Activity Type	Ensure the activity type is set to **HIERARCHYRESAVE** as shown in the image. This is typically auto-filled based on the widget type.
Status	Select the status for activity type.

3. In the **Dynamic Parameters** section, configure the settings as described in the table below.

> **Note:**
>
> The Transition component is used to control the flow between various components in the Process Flow. Transition connects two activities and the flow is configured based on a Conditional Expression or Decision Rule.

Table 7-10 Dynamic Parameters for Reconciliation

Property	Description
Rule Name	Select the Rule Name to apply during execution for reconciling transactions and generating statements.

4. Click the check mark icon to save your changes. To close the configuration window, click anywhere outside the window on the canvas.

7.1.15 SCD (Slowly Changing Dimensions)

> **Note:**
>
> If you have entered the **Full Load** parameter as **YES**, you can load snapshot data for master tables and execute SCD process with additional parameters.
> In case, when you have entered the **Full Load** parameter as **YES** and if the load is incremental and does not contain all the nodes, those nodes are retired by default.
>
> When entered as **NO**, you can load incremental data for master tables and execute SCD process with additional parameters. The incremental data load for master tables is supported.
> The retired dimensions can be brought back as part of subsequent SCD load by updating the Closed Flag column to null or 'N'.

Current behavior of Disabled Nodes:

- The Closed Flag attribute is not supported for the Product Processor (PP) accounts.
- A new node in the Stage Master table with Closed Flag 'Y' is supported and will be ignored during SCD load.

To configure this widget, complete the following steps:

1. On the **Process Flow** page, from the LHS menu, expand **Widget** and drag and drop this widget on to the canvas.
2. Double-click the widget on the canvas. The configuration window is displayed with the **Activity** tab selected.
3. Under **Dynamic Parameters**, configure the settings described in the following table.
4. Click the tick mark icon to save your changes. To close the configuration window, click anywhere outside the window on the canvas.

Table 7-11 Dynamic Parameters for SCD Description

Property	Description
Datastore Name	Refers to the name of the Information Domain. By default, the Information Domain to which you are connected is selected.
Name	Select the name of the required dimension table.
Full Load	If you have entered the **Full Load** parameter as **YES**, you can load snapshot data for master tables and execute SCD process with additional parameters. When entered as **NO**, you can load incremental data for master tables and execute SCD process with additional parameters.
Execution Venue	Enter name of the Execution Venue. This field is case-sensitive and must be unique. For example, logical name “R_DF” is not allowed if a name “R_DF” exists. Ensure that there are no special characters such as `, {,},”, ‘, ~, <,>, /, \, and multiple spaces.

Table 7-11 (Cont.) Dynamic Parameters for SCD Description

Property	Description
Effective Date	Enter the start date from when the data is valid.

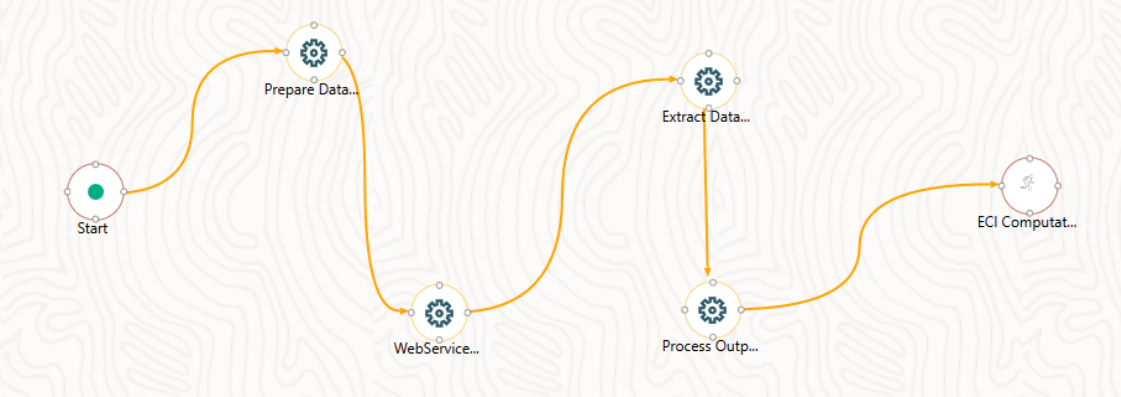
8
Activity

Service Task

The **Service Task** is available under the Activity toolbar and is used to invoke an application component, such as triggering a business rule to calculate a specific threshold. It is an automated task that gets activated within the process flow and is commonly utilized to execute business logic defined by an Execution Rule type application rule.

Service Tasks are used to invoke an External Model Service through stored procedures and functions.

Figure 8-1 An Example: Service Tasks Flow



8.1 Connector

A Connector helps connect two activities with a different path (other than the default), in case the default path is overlapping an existing flow.

8.2 Design a Pipeline

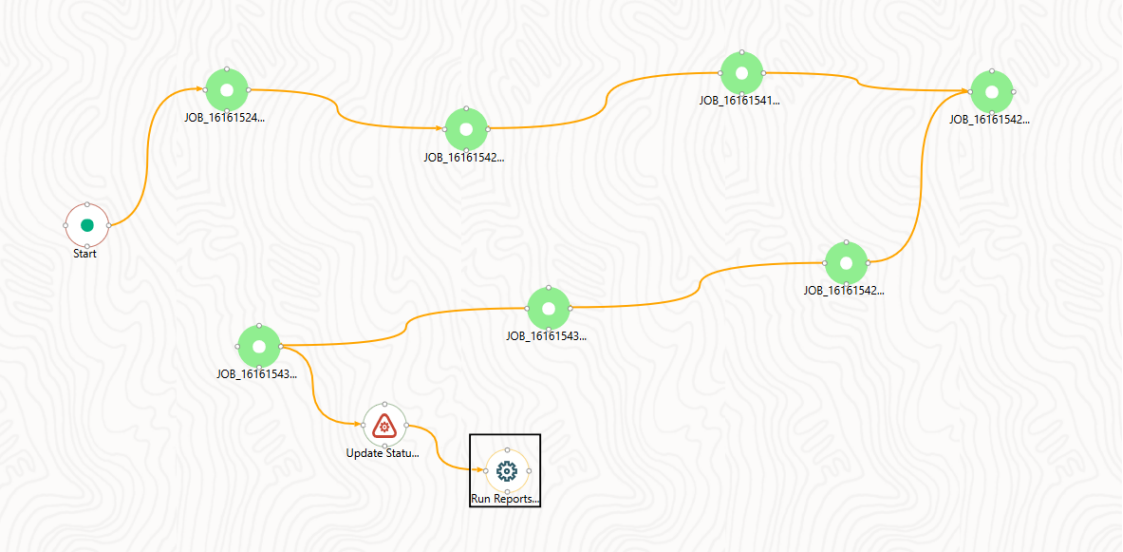
In **Financial Services Data Foundation Cloud Service for Banking**, pipelines are used to design and execute a sequence of tasks—either **Financial Services Data Foundation Cloud Service for Banking** tasks or external tasks—that lead to a well-defined outcome. This flow is created by utilizing various components available in the **Financial Services Data Foundation Cloud Service for Banking** component toolbar. Using the **Process Modeler**, you can orchestrate and run pipelines by modeling data pipelines.

8.2.1 Run Pipeline

A Run Process is used to create a Run Definition in the Rule Run Framework (RRF). A Run Pipeline is the visual representation of the Run enabled through Process Orchestration.

An Example of Run Pipeline

Figure 8-2 Run Pipeline Example



8.2.2 Creating a Run Pipeline

1. Click the **+** icon in the **Process Modeler Summary** page.

Table 8-1 Process Details

Field	Description
Process ID	The **Process ID** is automatically generated and unique for each process. This ID identifies the specific process within the system.
Process Name	Enter the name of the process. This will help users identify the process within the application.
Process Description	Provide a brief description of what this process does, to give context and help other users understand its purpose.
App Package ID	The **App Package ID** refers to the identifier for the application package that contains the process. In this case, it is set to "FSDF Components."
Registered Topics	This field allows the selection or listing of topics that are registered in the system. Topics could refer to areas of data or categories the pipeline will work with.
Spark DB	The **Spark DB** option indicates whether or not the process is using **Apache Spark** for database operations. Here, it is set to "No," meaning Spark is not being used.
Tag	This is an optional field where tags can be added for better categorization or to facilitate searching for this process in the future.

Table 8-1 (Cont.) Process Details

Field	Description
Service ID_Workspace	This shows the **Service ID_Workspace** associated with the process. The service is identified as **Oracle Financial Services Data Service for Analytics**, which suggests this process is related to data services for financial analytics.

2. Click the **check mark** icon to save the entered details.

   The Process Flow canvas is displayed.

3. By default, the START activity appears on the canvas indicating the beginning of the process.
4. Drag and drop components from the toolbar on the left. You can drag and drop service tasks, gateways, connectors, and widgets on the canvas.
5. To connect one component with another, right-click the source component and choose **Start Link**. Next, click on the destination component and click **End Link**. Alternatively, if you drag and drop a component on another component on the canvas, the two components will be linked.
6. Proceed to link all the components per your requirement.
7. Click **Save**.

8.2.3 Creating and Executing a Custom DQ Run Pipeline

1. Click the **+** icon in the **Process Modeler Summary** page.

Table 8-2 Process Details

Field	Description
Process ID	The **Process ID** is automatically generated and unique for each process. This ID identifies the specific process within the system.
Process Name	Enter the name of the process. This will help users identify the process within the application.
Process Description	Provide a brief description of what this process does, to give context and help other users understand its purpose.
App Package ID	The **App Package ID** refers to the identifier for the application package that contains the process. In this case, it is set to "FSDF Components."
Registered Topics	This field allows the selection or listing of topics that are registered in the system. Topics could refer to areas of data or categories the pipeline will work with.
Spark DB	The **Spark DB** option indicates whether or not the process is using **Apache Spark** for database operations. Here, it is set to "No," meaning Spark is not being used.

Table 8-2 (Cont.) Process Details

Field	Description
Tag	This is an optional field where tags can be added for better categorization or to facilitate searching for this process in the future.
Service ID_Workspace	This shows the **Service ID_Workspace** associated with the process. The service is identified as **Oracle Financial Services Data Service for Analytics**, which suggests this process is related to data services for financial analytics.

2. Click the **check mark** icon to save the entered details.

   The Process Flow canvas is displayed.

3. By default, the START activity appears on the canvas indicating the beginning of the process.
4. Drag and drop the **RUN DQ Rule** process widget.
5. To connect one component with an other, right-click the source component and choose **Start Link**. Next, click on the destination component and click **End Link**. Alternatively, if you drag and drop a component on another component on the canvas, the two components will be linked.
6. Proceed to link all the components per your requirement.
7. Double click the **RUN DQ Rule** process widget to enter the details. For more information on the parameters, see RUN DQ RULE section.
8. Click **Save**.
9. Execute the Custom DQ Run Pipeline. For more details on how to execute a pipeline, see Run Pipeline.

8.3 Additional Functionalities

This section describes how to manage pipelines including tasks such as viewing, modifying, copying, or deleting pipelines.

8.3.1 Modifying a Pipeline

This option enables you to modify a pipeline. To modify a pipeline, complete the following steps:

1. On the **Process Modeler** window, search for the pipeline and click the pipeline name. The **Process Flow** tab is displayed.
2. Modify the process flow, definition, application rules, or data fields as required.

> **Note:**
>
> If you try to delete a component in the Business Pipeline that is used as a Sub Pipeline, a confirmation message is displayed. After you confirm the deletion, an alert is displayed with the list of the Business Pipelines where the Sub Pipeline is used. Click **OK** again to delete the component.

8.3.2 Viewing a Pipeline

Use this option to view the workflow of an existing business process.

From the **Process Modeler** window, click the more information icon (the three horizontal dots icon) to view the sub-menu and select **View**. The Process Flow of the selected pipeline is displayed.

You can also **View** the pipeline in a new window by clicking on the **Launch in a new window**.

8.3.3 Copying a Pipeline

Use the copy option to quickly create a new business process based on an existing process by updating the process flow or other required details.

To copy a business process, complete the following steps:

1. From the **Process Modeler** window, click the more information icon (the three horizontal dots icon) to view the sub-menu of the selected pipeline and select **Copy**.

   The **Process Flow** window appears.
2. In the **Process Details** window, enter a unique **Process ID**, **Process Name**, and **Process Description**. If you select the same **App Package ID**, then **Data Fields** and **Application Rules** are also copied.
3. Enter a **Tag** for the Process ID.
4. Select the **Service ID_Workspace** from the drop-down list.
5. Click **check mark** to save the copied pipeline.

8.3.4 Deleting a Pipeline

Use this to delete a Business Pipeline or Run Pipeline.

To delete a pipeline, complete the following steps:

1. From the **Process Modeler** window, click the delete icon corresponding to the pipeline you want to delete.
2. A confirmation window is displayed. If you wish to delete the process, click **Accept**.

   If this pipeline is used as a Sub Pipeline in another Business Pipeline, an alert is displayed with the list of Business Pipelines where it is used.

> **Note:**
>
> You cannot delete the pipeline, if it is already triggered.

8.3.5 Process Monitor

Use the Process Monitor to monitor the current stage of a process. After integrating with a service, a workflow is invoked. After it is invoked, the workflow goes through all the defined stages. Using the Process Monitor, you can view all the stages of the workflow such as current stages, stages to follow, if any, and finished.

Your user group must be mapped to the function role WFMACC (Workflow Monitor Access) to access the Process Monitor window.

1. On the **Process Modeler** page, click the **Process Monitor** icon on the header to view the Process Monitor page. All workflows that are invoked from the service are displayed along with details such as **Entity Name**, **Entity ID**, **Process Name**, **Process Description**, **MIS Date**, **Execution Start Time**, **Last Execution Time**, **Last Updated By**, and **Status**.

To monitor only a selected pipeline, on the Process Modeler summary page, click the menu button corresponding to that pipeline and select **Process Flow Monitor**. The execution details are displayed.

Click the menu button corresponding to the selected pipeline to view options to:

- **Resume**: To resume a Run pipeline.
- **Re-run**: To execute a Run pipeline again irrespective of the previous execution status.
- **Abort**: To abort an ongoing Run pipeline

> **Note:**
>
> In case of Disaster recovery (DR) after switch over to the secondary site, the processes that are in running status should be aborted and resume to proceed further.

To return to the Process Modeler page, click the Process Modeler icon on the header of the Process Monitor page.

Use the **Search** field to search for a specific pipeline by providing a keyword from either the Process ID, Process Name, or Process Description of the process you are looking for and click **Search**.

8.3.6 Executing Run Pipeline

After creating your Run pipeline, your next step is to execute it. To execute a Run pipeline, complete the following steps:

1. From the **Process Modeller Summary** page, click the menu button corresponding to Run pipeline you want to execute and select **Execute Run**.
2. The **Execution** window is displayed.
3. Select the **Execution Type**, **Object ID**, and **Application Params (in JSON format)** and click **Execute**.

The execution is triggered using the selected FIC MIS DATE. The RUNSKEY is generated and inserted into the “DIM_RUN” table. For the RUNSKEY generated, the corresponding user-selected parameters are inserted into the “RUN_EXE_PARAMETERS” table.

You can use the Process Flow Monitor to view the status of the execution as the Run pipeline is executed.

8.4 Data Fields

Data Field, also known as a process variable, enables Process Pipelines access and store information from external sources. Often, the process flow is based on the value of this information. In other cases, this information is the result of running tasks in the pipeline.

8.4.1 Add a Data Field

1. On the **Process Modeler** page, click the pipeline to which you want to add a Data Field.
2. Select **DataFields** from the header to display the **Data Fields** window.
3. Click the + icon to display the **Addition of Data Field** window.
4. Provide the details described in the following table and click the check mark icon to save the details:

Table 8-3 Data Field Details Description

Field Name	Description
Data Field Code	Enter the Variable Name/Code, which will be used to read or write into this variable. This field is non-translatable.
Data Field Description	Enter a brief description.
Data Field Type	Select the Data Field Type.
Initial Value	Enter the initial value for the data.
Is Mandatory	Applies to the **AOM** Data Field Type. **Yes** is the default selection for AOM. Select **No** if you do not want this parameter to be displayed as Execution Parameter for the Run pipeline.
Scope	Select the scope of the Data Field. Options are: • **Process** - To use the Data Field only in the current process. • **Package** - To use the Data Field across all the processes in the package.

8.4.2 System Data Fields

Some data is tracked internally by Financial Services Data Foundation Cloud Service For Banking using a predefined set of Data Fields such as Status. You can access these activity instance attributes in the same way you access Regular Data Objects, but you cannot assign them new values.

8.5 PMF Dashboard

When multiple pipelines are running in parallel, it becomes important to monitor their execution status. This includes tracking whether tasks within the pipelines have completed, failed, been canceled, or are still in progress.

The **Process Modeling Framework (PMF) Dashboard** provides detailed execution information for all PMF processes associated with **Loader** components and **Data Quality** executions within an environment (tenancy). Access to the PMF Dashboard is restricted to authorized users who belong to the **Data Foundation Admin Group** or the **Data Foundation Operations User Group**.
Authorized users can drill down into each PMF process to view related activities and examine recorded events for each activity. This includes information such as timestamps, event messages, and error codes, helping users monitor the progress of each task. Additionally, users can monitor a pipeline and view it directly on the PMF canvas.

To access the PMF Dashboard, click the **PMF Dashboard** icon on the **Process Modeler** page. Once on the dashboard, you can explore detailed information about the executed processes and events (activities) that occurred during task execution.

You can filter activities and tasks by parameters such as **completion status**, **Process ID**, or **execution date**. The results can also be exported as a report in **Excel** or **CSV** format.

8.5.1 View and Filter PMF Dashboard

The **PMF Dashboard** provides detailed, drill-down information about the processes that have run, along with the events (activities) that occurred during task execution. You can use the available filters to quickly search for and locate a PMF process, whether it completed successfully or failed.

> **Note:**
>
> The PMF Dashboard supports wildcard search where you can search for objects without having to provide the complete activity name, process name, or instance ID. In addition, the search feature is case-agnostic; you can type the search strings in upper or lower case.

To view and filter the PMF Dashboard:

1. From the **Data Foundation Cloud Service** home page, select **Process Orchestration**.

   The **Process Modeler** page is displayed.

2. Click the **PMF Dashboard** icon on the header. The **PMF Dashboard** page appears and displays the PMF processes that have run.

3. You can filter the PMF dashboard to quickly search for a PMF process you are interested in. Note that the search is case-sensitive. The following filter options are available:

   - **Process Status**: Select one or multiple statuses to filter the PMF dashboard. For example: to see all processes that have failed, click **Process Status** under the search bar and select **Failed** to view a list of all the processes that have failed.
   - **Process Name**: Enter the process name.
   - **Process ID**: Enter the process ID.
   - **Process Instance ID**: Enter the process Instance ID.
   - **More Filters**: Select from one of the multiple options available.
   - **Execution Date**: Select a time period or a date range based on the time zone in which the user is logged in.

   The dashboard is refreshed and the PMF process(es) matching your filter criteria are displayed.

4. You can drill-down each PMF process to view the activities (tasks) that have been defined for that process. Use the additional filters to filter activities based on **Activity**, **Event Parameters**, **Event Message**, and **Event Error Code**. To apply a filter, click inside the search bar, select the filter, and press Enter.

5. Each PMF process has the following information:

   - **Process Instance ID**: The unique system-generated value that identifies the process instance.

- **Data Source ID**: The Data Source ID of the process.
- **Run Params**: The input parameters provided to execute this process.
- **As Of Date**: The As of date.
- **Run Skey**: The unique number created for each instance.
- **Start Time**: Start time of the process execution.
- **End Time**: End time of the process execution.
- **Duration**: Duration of the process execution.
- **Status**: Indicates if the process has completed, failed, was canceled, or is still running.
- **Monitor**: Click this icon to view the progress of this process on the PMF canvas.

6. Click the PMF process link to drill down and display the activities associated with this process. The associated activities (for example: SCD, Connectors, Run DQ Rule, and so on) are displayed. For each activity, the following information is available:
   - **Component**: The component type.
   - **Start Time**: Start time of the activity execution.
   - **End Time**: End time of the activity execution.
   - **Duration**: Duration of the activity execution.
   - **Status**: Indicates if the activity has completed, failed, was canceled, or is still running.
7. Click the activity name link to drill down further and view details of the events within this activity. For example: if the activity is an **Insert Connector**, the following event information is displayed:
   - **Timestamp**: The latest record is displayed on top.
   - **Number of error records**: The number of error records.
   - **Event**: The event that is being recorded.
   - **Number of records loaded**: The number of records that were loaded.
   - **Event message**: Status of the event completion.

8.5.2 Exporting PMF Dashboard Report

You can also export the PMF dashboard contents as a report in Excel or CSV formats. Here again you have options to filter the data that is exported.

To export contents of the PMF Dashboard into a report:

1. On the **PMF Dashboard** page, use the filters to select the PMF process(es) and the activities within the PMF process(es) you are interested in.
2. Click **Export**.
3. Choose the format for your report (Excel or CSV).
4. You need to apply at least one filter criteria to export the report as there is a limit to the number of records you can export. To filter the contents to export to your report, click the **With Filter** option and select the filter criteria from the search box.
5. Click **Export** and save the report file to a local directory. The report will contain details of the selected process(es), activities within the process, and the tasks within the events.

> **Note:**
>
> The export file can contain a maximum of 6500 records (rows) only. Hence it is recommended to use at least one filter when exporting the contents of the Dashboard to a report.

8.6 PMF Email Notification

Use this feature to configure the email notifications for package/process or user level.

The PMF Email Notification page enables you to send Email notifications for indicating the status (completion or failure) of a PMF process. This will permit pushed notification of users via email with failure/completion status for taking timely and necessary actions.

8.6.1 PMF Email Notification at Package or Process Level

Use this feature to set Email notification at package/process level.

To set the PMF Email notification at Package/Process level:

1. From the **Financial Services Data Foundation Cloud Service for Banking** Home page, select **Data Pipelines**.

   The **Process Modeler** page is displayed.

2. Click the **PMF Email** icon on the header. The **PMF Email Notification** page appears and displays the Package and Process Level email notifications.

   You must be mapped to FINANCIAL SERVICES DATA FOUNDATION CLOUD SERVICE FOR BANKING ADMNGRP user group to view the PMF Email icon on the header.

3. Click the plus icon and select the Package radio button and the package IDs for which the email notifications has to be triggered.

4. Click **Apply**.

   The selected packages are displayed.

5. Click **Add User** and select the **User** or **User Group** to whom the email notifications to be shared.

   You should add users at package level and not at the process level.

6. Click **Apply** and **Save**.

8.6.2 PMF Email Notification at User Level

Use this feature to set Email notification at user level.

To set the PMF Email notification at User level:

1. From the **DFCS Home** page, select **Data Pipelines**.

   The **Process Modeler** page is displayed.

2. Click the **PMF Email** icon on the header. The **PMF Email Notification** page appears and displays the Package and Process Level email notifications.

3. Click the plus icon and select the User/User Group radio button and the users or groups to receive the email notifications.

You should add users at package level and not at the process level.

4. Click **Apply**.

   The selected User/User Groups are displayed.

5. Click **Add Package** and select the Package or Process for which the email notifications has to be triggered

6. Click **Apply** and **Save**.

8.6.3 PMF Email Notification Template

Use this page to configure the email notification messages for package/process or user level.

To configure the PMF Email notification messages:

1. From the **DFCS Home** page, select **Data Pipelines**.

   The **Process Modeler** page is displayed.

2. Click the **PMF Email** icon on the header. The **PMF Email Notification** page appears and displays the Package and Process Level email notifications.

   You must be mapped to DFCSADMNGRP user group to view the PMF Email icon on the header.

3. Click **Template** option next to Email Notification at the bottom of the page.

   The Email Notification template is displayed. By default, the following variables are added to the message body:

   - Tenant-Id
   - Run-Sk
   - Process-Id
   - Object-Id
   - Task-Status
   - As-Of-Date
   - Entity-Name
   - Process-End-Time
   - Entity-Id
   - Title
   - locale
   - Process-Start-Time
   - Entity-Type
   - Process-Name
   - Process-Instance-Id
   - Received-On

4. You can remove the variables from the Template body to be displayed in the email based on your requirements.

   The Template you configure is at global level and not at the user group or package/process level.

> **Note:**
>
> Forwarding the email will send all the parameters available and do not retain the template.

8.7 Process Monitor

Use the Process Monitor to monitor the current stage of a process. After integrating with a service, a workflow is invoked. After it is invoked, the workflow goes through all the defined stages. Using the Process Monitor, you can view all the stages of the workflow such as current stages, stages to follow, if any, and finished.

Your user group must be mapped to the function role WFMACC (Workflow Monitor Access) to access the Process Monitor window.

1. On the **Process Modeler** page, click the **Process Monitor** icon on the header to view the Process Monitor page. All workflows that are invoked from the service are displayed along with details such as **Entity Name**, **Entity ID**, **Process Name**, **Process Description**, **MIS Date**, **Execution Start Time**, **Last Execution Time**, **Last Updated By**, and **Status**.

To monitor only a selected pipeline, on the Process Modeler summary page, click the menu button corresponding to that pipeline and select **Process Flow Monitor**. The execution details are displayed.

Click the menu button corresponding to the selected pipeline to view options to:

- **Resume**: To resume a Run pipeline.
- **Re-run**: To execute a Run pipeline again irrespective of the previous execution status.
- **Abort**: To abort an ongoing Run pipeline

> **Note:**
>
> In case of Disaster recovery (DR) after switch over to the secondary site, the processes that are in running status should be aborted and resume to proceed further.

To return to the Process Modeler page, click the Process Modeler icon on the header of the Process Monitor page.

Use the **Search** field to search for a specific pipeline by providing a keyword from either the Process ID, Process Name, or Process Description of the process you are looking for and click **Search**.

8.7.1 Process Monitor

From the **Process Monitor** window, click the Entity ID link corresponding to the process you want to monitor.

The status of the activity is represented as: **Completed**, **Failed**, **Cancelled** or **Running**.

You can also set auto-refresh in the header. Select **Enabled** in **Auto Refresh**, enter a value from 1 to 10 in **Refresh Interval (In Min)** and then click **Apply**.

8.7.1.1 Viewing Activity Execution Logs

Use this feature to view logs of the execution of each activity from the Process Monitor window.

To view Activity Execution Logs:

1. From the Process Monitor page, click the process which you want to monitor. The canvas displaying the workflow along with the status is displayed. On the canvas, double-click an activity to view the **Activity Execution** window. The activity details are displayed.
2. Click the **Execution Logs** button to view the Activity Execution Log.

   The Log Viewer window shows all the execution stages (successful and failed) of the selected activity if it is already executed.

   The **Batch Run ID** and **Process ID** fields show the information for the previously selected parameters in the activity configuration.

   > **Note:**
   >
   > Use the **Search** field to search within the execution log.

   The information available is segregated in to sequence, timestamp, severity, and message.

   You can sort by Ascending or Descending Order by clicking the Up and Down Arrows in the Header Row.

   - Sequence depicts the order of the log messages generated during execution.
   - Timestamp shows the date and time of the generation of the log message.
   - Severity shows labels such as Info to show the type of message, which helps determine if any intervention is required.
   - Message displays the log message generated.

   > **Note:**
   >
   > The information in the Log Viewer window is specific to the selected Process Orchestration activity such as a widget or transition.

8.8 Abort Run Pipeline

You can abort a Run Pipeline that is in the process of execution.

To abort a Run pipeline, complete the following steps:

- From the **Process Monitor** window, click the menu icon corresponding to the Run pipeline that is being executed that you want to abort and click **Abort**.

8.9 Resume Run Pipeline

You can resume a Run Pipeline which has not been executed successfully or which has been explicitly interrupted, or canceled, or put on hold during the Execution process. By resuming a

Run pipeline, you can continue its execution directly from the point of interruption or failure and complete executing the remaining tasks.

To resume a Run Pipeline

- From the **Process Monitor** page, click the menu icon corresponding to the Run pipeline you want to resume and click **Resume**.

8.10 Re-run Run Pipeline

You can re-run a Run pipeline that was previously executed, irrespective of the previous execution state.

To re-run a Run pipeline, complete the following step:

- From the **Process Monitor** page, click the menu icon corresponding to the Run pipeline you want to re-run and click **Re-Run**.

8.11 Application Rule

This Application Rule is used to execute any queries in the Process Flow.

Function Application Rule

Function as the Application Rule Type from the drop-down list. Function represents a predefined function for application logic.

Table 8-4 Function Application Rule Details Description

Field Name	Description
Name	Enter a unique name for the Function Application Rule.
Rule Type	Select the Rule Type from the drop-down list. The Function Application Rule can be used as a Decision Rule, Execution Rule, or Selection Rule based on your requirement.
Execution Type	Displays the Application Execution Type as Function.
Implementation Detail	• Decision Rule- For Decision Rule the Function Statement should return 'PASS' for success condition. For example, select 'PASS' from dual where {EXPENSES} <= {THRESHOLD} Note: {EXPENSES}, {THRESHOLD} are user defined Data Fields. • Execution Rule- For Execution Rule the Statement can be any DML statement. For example, update fct_expenses set expenses={EXPENSES} where id = {WF_ENTITYID} • Selection Rule- For Selection Rule the Statement should be a Select statement that returns a list of values. For example, select v_created_by from fct_expenses id = {WF_ENTITYID}

Table 8-4 (Cont.) Function Application Rule Details Description

Field Name	Description
Return Parameter	Select the Data Field that receives the return parameter of the Rule, from the drop-down list. • For Execution Rule type, the business logic is implemented in the method and the parameter value returned from the Function Rule is saved in the mapped Data Field. • In the case of Selection Rule type, the Java method should be a String value. • In the case of the Decision Rule type, the Java method should return Boolean values “True/ False”.
Scope	Select the Scope as Process to use the Application Rule only in the current process or Package to use the Application Rule across all the processes in the package.

Expression Application Rule

Used to define custom expressions for business rules or calculations. This is same as the SQL execution type. You need to specify only where clause in the Expression field. It can be any SQL expressions including ‘AND’/ ‘OR’.

Table 8-5 Expressions

Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the Rule Type from the drop-down list. The Expression Application Rule can be used as Decision Rule, Execution Rule or Selection Rule based on your requirement.
Execution Type	Displays the Application Execution Type as Expression.
Expression Type	Select Expression Type as SQL to use SQL expressions or JSON to use JSON expressions.
Expression	Enter the expression in SQL format or JSON format.
Return Parameter	Select the Data Field which will receive the return parameter of the Expression, from the drop-down list. • For ExecutionRule type, the business logic is implemented in the method and the parameter value returned from the Application Rule is saved in the mapped Data Field. • In case of SelectionRule type, the Application Rule should be a String value. • In case of Decision Rule type, the Application Rule should return Boolean values “True/ False”.

Table 8-5 (Cont.) Expressions

Name	Description
Scope	Select the Scope as Process to use the Application Rule only in the current process or Package to use the Application Rule across all the processes in the package.

Attribute Expressions

The Attribute Expressions is used to configure expressions based on specific attributes or data fields.

Table 8-6 Attribute Expressions

Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Displays the rule type as DecisionRule. This Rule Execution type supports only DecisionRule type.
Execution Type	Displays the Application Execution Type as Attribute Expression.
Attribute	Select the attribute for which you want to define the application rule, from the drop-down list. The list displays the attributes configured for the selected application and component. Click **Add** to add values to the selected attributes. A row is added in the Attribute Values pane. Click the Value column to select the values for the attribute from the drop-down. You can select one or more values. You can delete a row by clicking **Delete** icon. You can select multiple attributes and click Add to assign values to those attributes.

Advanced Attribute Expressions

This is an advanced version of Attribute Expression Application Rule with additional logical conditions and assignment operators. The expression can be dynamically built and will return True or False value after evaluation. This is used as a DecisionRule in transitions. You can define this application rule with multiple conditions and nested groups.

Table 8-7 For Advanced Attribute Expression Application Rule

Field Name	Description
Rule Name	Enter a unique name for the Application Rule.
AND/ OR	Select the logical operator to be used for the conditions in a group.

Table 8-7 (Cont.) For Advanced Attribute Expression Application Rule

Field Name	Description
Add Condition	When you click Add Condition, a new row gets added. To define a condition, select the attribute, the operator and the value from the drop-down lists. Multiple values can be selected for each attribute. **Attribute**- The drop-down list displays the attributes configured for the selected application and component. **Operator**- Available options are in,=,<>,<,<=,>,>=. · Value- Displays the values configured for the selected attributes. Select the required value. Click Remove Condition to delete an already added condition.
Add Group	Click Add Group if you want to have nested conditions. For each group, select the required logical operator as AND or OR. Click Remove Group to delete a group of conditions.

JSON Write to DB

This Application rule is used to extract data from the JSON Path Expression, which gets returned from a Rest API or Web Service call, and you can store it into a Data Field for further processing.

Table 8-8 For JSON Path Expression Application Rule

Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the rule type from the drop-down list. The available rule types are Decision Rule and Execution Rule. For Decision Rule, the output of JSON Path Expression will be compared with RHS expression and rule will return either true or false accordingly. For execution rule, JSON Path Expression is evaluated and the output is returned to the DataField selected as Output DataField.
Execution Type	Displays the Application Execution Type as JSON Path Expression.
JSON Input	Select the Data Field in which the output of Web Service is stored from the drop-down list. You should select a DataField which has JSON as its value.
JSON Path Expression	Enter the JSON path expression.
Operator	This field is displayed only if Rule Type is selected as Decision Rule. Select the required operator for comparison from the drop-down list. The options are =,<,>,>= and <=.

Table 8-8 (Cont.) For JSON Path Expression Application Rule

Field Name	Description
RHS Expression	This field is displayed only if Rule Type is selected as Decision Rule. Enter the expression to which you want to compare the JSON path expression.
Output DataField	This field is displayed only if Rule Type is selected as Execution Rule. Select the DataField to which you want to return the value of JSON Path Expression, from the drop-down list.
Scope	Select the scope of the Application Rule from the drop-down list. The options are: • Process- Select Process if you want to use the Application Rule only in the current process. • Package- Select Package if you want to use the Application Rule across all the processes in the Application package.
Convert To Type	Select JSON ARRAY to store the output in Array format or select String to store as a string, from the drop-down list.

JSON Read from DB

This Application Rule is used to read data from the database in JSON format.

Table 8-9 JSON Read from DB

Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Only the Execution Rule type is supported.
Execution Type	Displays the Execution Type as JSON Read From DB.
Table Name	Enter the table name from which you want to read the data.
Column List	Enter the column names of the selected table.
Where Condition	Enter the filter condition (where clause) of the SQL query.
Return JSON Type	Select the JSON type of the returned value as JSON Object or JSON Array based on your requirement. **JSON Object**- Select this option if the returned value is a single row. **JSON Array**- Select this option if the returned data has multiple rows.
Output DataField	Select the DataField to which you want to return the value of the SQL query, from the drop-down list.

Table 8-9 (Cont.) JSON Read from DB

Field Name	Description
Scope	Select the scope of the Application Rule from the drop-down list. The options are: • Process- Select Process if you want to use the Application Rule only in the current process. • Package- Select Package if you want to use the Application Rule across all the processes in the Application package.