

**Oracle Financial Services Analytical
Applications Infrastructure**
Process Modelling Framework
Orchestration Guide

Release 8.1.0.0.0

October 2021

ORACLE
Financial Services

ORACLE

OFS AAI Process Modelling Framework Orchestration Guide

Copyright © 2022 Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

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

Document Control

Version Number	Revision Date	Change Log
1.6	August 2022	Inline process is not supported from versions 8.0.8.5.0 onwards. All references to Inline process are removed.
1.5	October 2021	Updated the User Role Mapping and Access Rights section (Doc 33506217).
1.4	January 2021	Added the Patch Application Information for v8.1.0.1.0 section to provide information for the mandatory patch 32198124 (Doc 32406409).
1.3	October 2020	Added the User Role Mapping and Access Rights section to provide information specific to the function roles required to access and use PMF (Doc 31589561).
1.2	September 2020	Added the Using ICC Batch section for Doc 31854849.
1.1	June 2020	Added note in the Operations Module section for Object ID field (Doc 31460787).
1.0	Created May 2020	Updated the following sections for the enhancements done in 8.1.0.0.0 version: <ul style="list-style-type: none">• Applying Filter Condition on Run Pipeline• Applying Filter Condition on Sub Pipeline

Table of Contents

1	Introduction	10
1.1	Categorization of Pipelines	10
1.2	Key Features of Process Modelling Framework	11
1.3	Process Pipeline Flow.....	12
1.4	Prerequisites.....	12
1.4.1	<i>User Role Mapping and Access Rights</i>	13
1.5	Access Process Modelling Framework.....	14
2	Process Modeler	15
3	Components for Designing Your Process Flow	17
3.1	Transitions	17
3.2	Gateways	18
3.3	Connector.....	18
3.4	Human Task	18
3.5	Service Task.....	18
3.6	Sub Pipeline	19
3.7	OFSAA Widgets	19
4	Artifacts of Process Modelling	20
4.1	Application Package	20
4.2	Application Registration	20
4.2.1	<i>Object Type Entry</i>	20
4.2.2	<i>Object Type Process Mapping</i>	21
4.3	Data Fields	21
4.3.1	<i>Add a Data Field</i>	21
4.3.2	<i>System Data Fields</i>	23
4.3.3	<i>AOM Data Field</i>	23
4.4	Application Rules.....	24
4.4.1	<i>SQL Application Rule</i>	25
4.4.2	<i>Stored Procedure Application Rule</i>	26
4.4.3	<i>Function Application Rule</i>	28
4.4.4	<i>Java Application Rule</i>	29

4.4.5	<i>Java External API Application Rule</i>	31
4.4.6	<i>Outcome Rules Application Rule</i>	33
4.4.7	<i>Expression Application Rule</i>	34
4.4.8	<i>Rest Service Application Rule</i>	35
4.4.9	<i>Attribute Expression Application Rule</i>	39
4.4.10	<i>Advanced Attribute Expression Application Rule</i>	40
4.4.11	<i>JSON Path Expression Application Rule</i>	41
4.4.12	<i>JSON Read From DB Application Rule</i>	43
4.4.13	<i>JSON Write To DB Application Rule</i>	45
4.5	Configuring Application Object Model (AOM)	46
4.5.1	<i>AAI_AOM_APP_COMP_ATTR_MAPPING Table</i>	46
4.5.2	<i>Example for Run Pipeline</i>	47
4.5.3	<i>Attribute Types</i>	47
4.5.4	<i>AAI_AOM_APP_COMP_ATTR_TL Table</i>	48
4.5.5	<i>Usage of AOM Attributes in Run Pipeline</i>	48
5	Design a Pipeline.....	50
5.1	Orchestrate a Business Pipeline	50
5.1.1	<i>An Example of a Business Pipeline</i>	50
5.1.2	<i>Creating a Business Pipeline</i>	50
5.2	Run Pipeline.....	52
5.2.1	<i>An Example of Run Pipeline</i>	52
5.2.2	<i>Creating a Run Pipeline</i>	53
5.2.3	<i>Design your Run Pipeline using OFSAA Widgets</i>	54
5.2.4	<i>Design your Run Pipeline using Sub Pipeline</i>	54
5.2.5	<i>Design your Run Pipeline using combinations of OFSAA Widgets and Sub Pipeline</i>	54
5.2.6	<i>Applying Filter Condition on Run Pipeline</i>	54
5.2.7	<i>Executing Run Pipeline</i>	58
5.2.8	<i>Abort Run Pipeline</i>	62
5.2.9	<i>Resume Run Pipeline</i>	63
5.2.10	<i>Re-run Run Pipeline</i>	63
5.3	Additional Functionalities	64
5.3.1	<i>Modifying a Pipeline</i>	64

5.3.2	<i>Testing a Pipeline</i>	64
5.3.3	<i>Viewing a Pipeline</i>	65
5.3.4	<i>Copying a Pipeline</i>	65
5.3.5	<i>Deleting a Pipeline</i>	65
6	Human Tasks.....	67
6.1	How to Use Human Task.....	67
6.1.1	<i>Activity Tab</i>	67
6.1.2	<i>Action Tab for Creating Tasks/Notification</i>	68
6.2	Additional Functionalities for Human Tasks	70
6.2.1	<i>Setting Task Expiry</i>	70
6.2.2	<i>Setting Task Escalation</i>	71
6.2.3	<i>Setting Reminder for Your Task</i>	72
7	Service Tasks.....	74
7.1	How to Use a Service Task	74
7.1.1	<i>Activity Tab</i>	74
7.1.2	<i>Implementation Tab</i>	75
8	Configuring OFSAA Tasks in Your Process Flow.....	77
8.1	How to Use an OFSAA Widget	77
8.2	Dynamic Parameters for OFSAA Widgets.....	78
8.2.1	<i>LoadT2T</i> 	78
8.2.2	<i>TransformDT</i> 	79
8.2.3	<i>RuleType2</i>  or <i>RuleType3</i> 	79
8.2.4	<i>MFModel</i> 	80
8.2.5	<i>Run</i> 	81
8.2.6	<i>DataQualityGroups</i> 	82
8.2.7	<i>RunExecutable</i> 	84
9	Orchestrating External Models/Components in Your Process Flow.....	87

9.1	How to invoke External Model through Web Service	87
9.1.1	<i>Data Preparation</i>	87
9.1.2	<i>Webservice Invocation</i>	88
9.1.3	<i>Data Extraction</i>	91
10	Configuring Custom Components	92
10.1	AAI_WF_COMPONENT_REGISTRATION Table	92
10.2	AAI_WF_COMPONENT_PARAMETERS Table	93
11	Executing Parallel Tasks	94
11.1	How to Use Parallel Gateways	94
12	Calling another Pipeline from Your Parent Pipeline	98
12.1	How to Configure Sub Pipeline	98
13	Configuring Email for Human Tasks	101
13.1	AAI_EMAIL_CONFIG Table	101
13.2	AAI_WF_APP_PACKAGE_B Table	101
13.3	AAI_WF_APP_REGISTRATION Table	102
13.4	AAI_WF_ACTIVITY_TASK_B Table	102
13.5	AAI_USER_PREFERENCE Table	103
13.6	AAI_WF_EMAIL_TEMPLATE Table	103
13.7	AAI_WF_BULK_MAIL_TRIGGER Table	103
13.8	CSSMS_USR_PROFILE Table	103
13.9	AAI_MAIL_AUDIT_TRAIL Table	104
14	Process Monitor	105
14.1	Monitoring a Business Process	106
14.2	Viewing Activity Logs	107
14.3	Viewing Execution Log for OFSAA Tasks	108
15	Invoking PMF Pipeline	110
15.1	Application UI	110
15.1.1	<i>Java API</i>	110
15.1.2	<i>Stored Procedure</i>	110

15.1.3	<i>Rest Service</i>	110
15.2	Within PMF Summary Screen UI	111
15.2.1	<i>Using Test Workflow</i>	111
15.2.2	<i>Using Execute Run</i>	111
15.3	Operations Module.....	111
15.4	Command Line Execution.....	113

16 Appendix A..... 114

16.1	Configuring Group Approval for Human Tasks.....	114
16.1.1	<i>Configuring Parallel Group Approval</i>	114
16.1.2	<i>Configuring Sequential Group Approval</i>	115
16.2	JsonPath Expressions	117
16.2.1	<i>Operators</i>	117
16.2.2	<i>Functions</i>	118
16.2.3	<i>Filter Operators</i>	118
16.2.4	<i>Path Examples</i>	119
16.3	Delegation.....	120
16.3.1	<i>Adding a Delegate</i>	120
16.3.2	<i>Viewing Delegation</i>	123
16.3.3	<i>Modifying Delegate Details</i>	123
16.3.4	<i>Revoking Delegation</i>	123
16.3.5	<i>Deleting Delegation</i>	123

17 Appendix B: Support APIs for Java External APIs..... 124

17.1	Connection API	124
17.1.1	<i>Jar Files Required</i>	124
17.1.2	<i>ConnectionAdapter Methods</i>	124
17.1.3	<i>Connection to Config Schema</i>	124
17.1.4	<i>Connection to Atomic Schema</i>	125
17.2	Logging API	126
17.2.1	<i>Jar Files Required</i>	126
17.2.2	<i>Debug Message</i>	126
17.2.3	<i>Error Message</i>	127

1

Introduction

Process Modelling Framework (PMF) is a design and execution framework that enables Process Pipeline developers to implement various Pipelines modeled by business analysts. Process Pipeline developers use the framework to orchestrate the Business Pipelines and Run Pipelines within OFSAA, and also to design the artifacts that participate in the Pipelines, in order to complete their implementation.

The Process Modeling Framework consists of Process Modeling components for modeling Pipelines and Process Monitor components for monitoring instantiated Pipelines of OFSAA applications.

See [Process Flow](#) for more information on how these tools fit into the Pipeline design and implementation.

Process Modeller aids in representing the various artifacts required for modeling and provides implementation details of the OFSAA process artifacts.

- OFSAA Process Pipeline or Run Pipeline
- Reusable process components like Sub Pipeline
- Process data (Data Fields)
- Implementation of various types of Human Tasks / Service Tasks
- Business Rules (Application Rules)
- Various External services implementations and other artifacts needed for complex implementations
- Configuring Notifications

1.1

Categorization of Pipelines

The application of the patch **32198124** in v8.1.0.1.0 creates clear segregation of the nature of tasks being orchestrated.

For details on how to apply the patch, see the [Oracle Financial Services Analytical Applications Infrastructure Installation Guide v8.1.0.1.0](#).

NOTE This section is applicable to v8.1.0.1.0 and later releases.

Applying the patch categorizes and updates the pipeline types appropriately depending on the nature of tasks that are being orchestrated. Further explanation on the nature of tasks and categorization is as follows:

1. Workflow Pipeline
2. Run Pipeline
3. Sub-Run Pipeline

4. Business Process Pipeline

You can define and orchestrate Business Process Pipelines only if the Oracle Financial Services Analytical Applications Infrastructure Extension Pack (OFSAAIE) is enabled in the OFS AAI setup. The Business Process Pipeline allows orchestration of various tasks and pipelines across Workflow, Run, and Sub-run Process Pipelines of any pipeline type whether it is a Business, Run, or Workflow Pipeline(s).

The various widgets in the PMF Canvas display depending on the type of pipeline selected, which are described as follows:

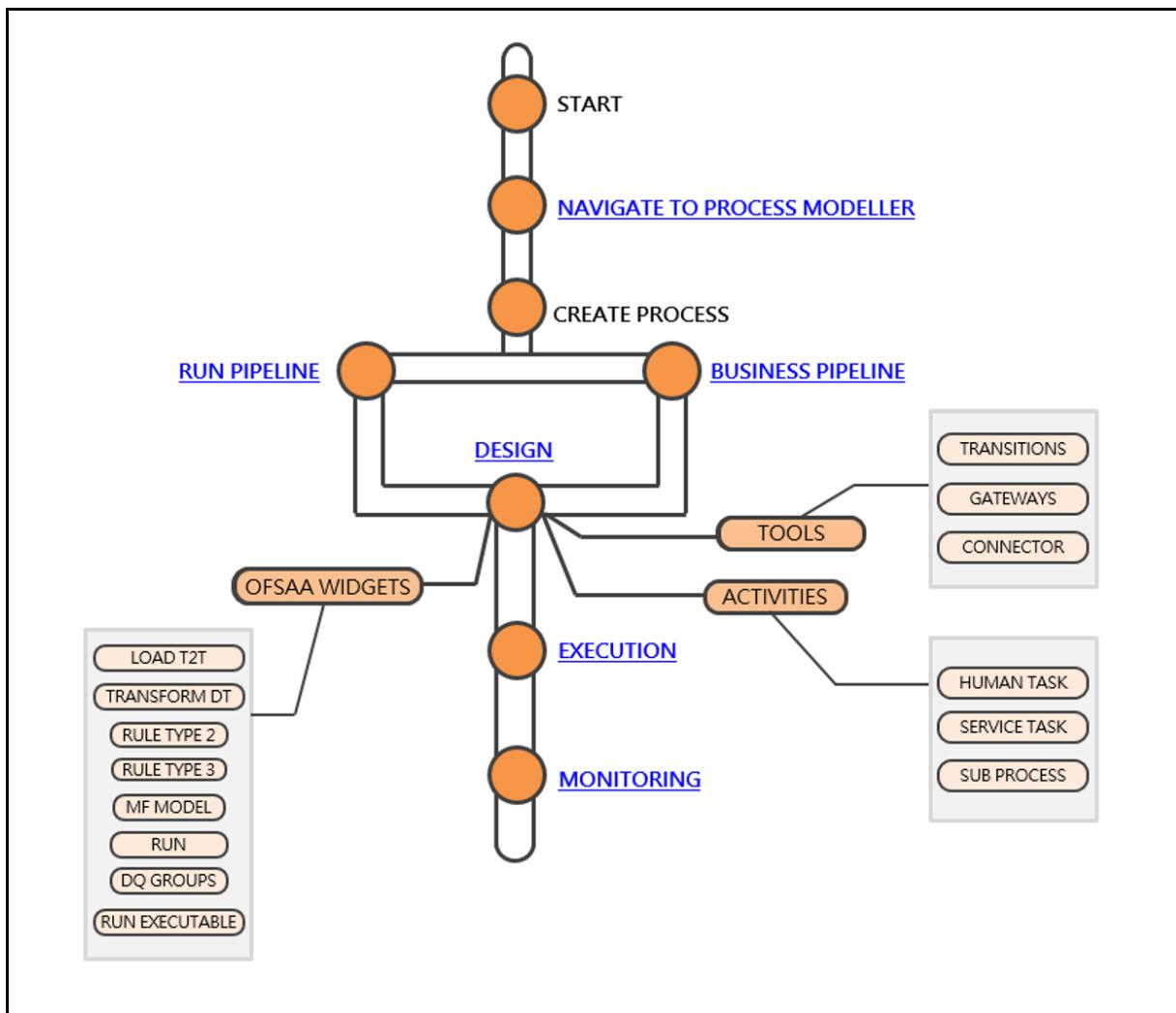
1. **Run Pipeline:** If any orchestrated pipeline consists of OFSAA tasks and service calls that run within the OFSAA context, it is categorized as Run Pipeline and it allows to stitch or orchestrate other processes of the type "subrunprocess". The Sub-run Process is further explained down or the Run can be mapped within a Run, which allows you to configure an array of tasks and orchestrate them based on the decisions attached.
2. **Workflow:** This represents a state-machine pipeline that allows you to orchestrate manual and automated system tasks, which helps you to stitch a state-change machine by performing manual or system tasks. If there is a Sub-pipeline, then you can stitch a Workflow pipeline.
3. **Subrunprocess:** This is a process similar to the Run process where a Sub-run Process Pipeline allows you to orchestrate the tasks in OFSAA as well as the services and API that run within the OFSAA context. You can also map other processes of the type "subrunprocess" within a Sub-run Process.
4. **Business Process:** This process allows you to access a full range of orchestration where the Business Process Pipeline can interleave a Run Pipeline to a Workflow Pipeline or can orchestrate a Sub-Run process or Workflow process. This enables the widget to have a full range of tasks in OFSAA as well as workflow tasks. This process also allows you to have service tasks that make calls to systems outside of OFSAA.

1.2 Key Features of Process Modelling Framework

- Support for visual modeling of the pipelines.
- Support for registration of Process /Activity/ Transition Logic implementation, separated from the modeling itself.
- Built-in orchestration engine (included within OFSAAI's runtime) for task execution (interactive model as opposed to the batch model supported through Rule Run Framework).
- Published interface for the abstraction of task implementation.
- Representation of the pipeline-routing rule logic in Java/ PL-SQL / Web-service.
- Stitching of OFSAA Components within the Process Pipeline
- Orchestration and execution of RRF Run.
- Reminder, Escalation, and Expiry of tasks.
- Registration of Custom Widgets.
- Process Monitoring Admin Tool to view the execution Process Instances.

1.3 Process Pipeline Flow

Figure1: Process Pipeline Flow



TIP

After you click the links in the Process Pipeline Flow, press ALT+ Left Arrow to come back to this page.

1.4 Prerequisites

The following is the prerequisite to access and perform functions in the PMF user-interface:

- [User Role Mapping and Access Rights](#)

1.4.1 User Role Mapping and Access Rights

User access to the PMF UI and ability to perform functions in it is dependent on the mapping of the user profile to the roles in the OFS AAI application and the access rights assigned.

The following user role mapping is mandatory for PMF:

Table 1: User Role Mapping for PMF

Role Code	Role Name	Functionality
WFACC	Workflow Access	<p>Assign this role to the user to access the Process Modeller menu from the Navigation Tree.</p> <p>NOTE: The mapping of this role does not allow view, edit, and add actions.</p>
WFMACC	Workflow Monitor Access	<p>Assign this role to the user to access the Process Monitor window.</p> <p>NOTE: The mapping of this role does not allow view, edit, and add actions.</p>
WFREAD	Workflow Read	Assign this role to the user to view the PMF workflow.
WFWRITE	Workflow Write	Assign this role to the user to perform view, edit, and add actions in PMF.
WFACCNEXE	Workflow Execute On Read	<p>Assign this Role to Users who have to Execute, Re-run, or Abort a Process.</p> <p>PREREQUISITE: Ensure that the Users are assigned the Workflow Access (WFACC) Role before you assign this Role.</p> <p>NOTE: This Role is available in the OFS AAI v8.1.0.2.0 and later versions.</p>

NOTE

For administrators, ensure that they are mapped to all the roles described in the preceding table to allow them to perform all types of operations in PMF.

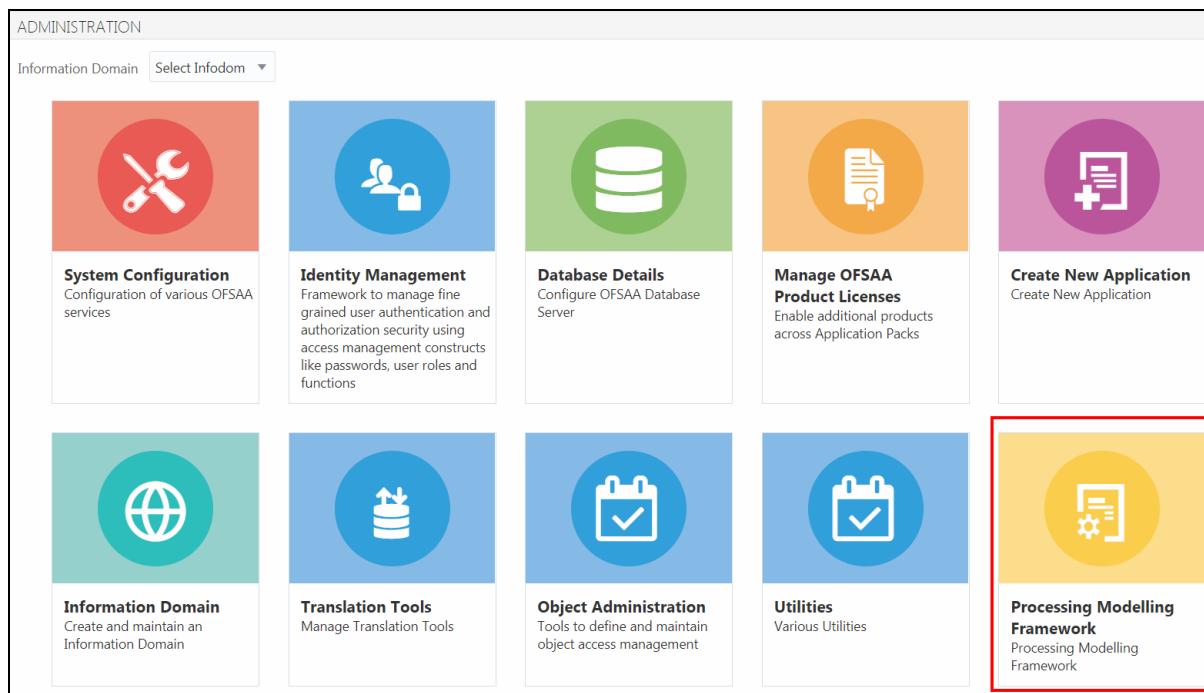
1.5

Access Process Modelling Framework

The following are the steps to access Process Modelling Framework:

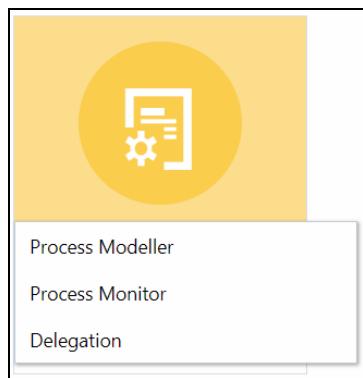
1. From the OFSAA Landing screen, click  **Administration**.

Figure 2: Administration landing page



2. Select the **Information Domain** from the drop-down list.
3. Click **Process Modelling Framework** tile to display the sub-menu.

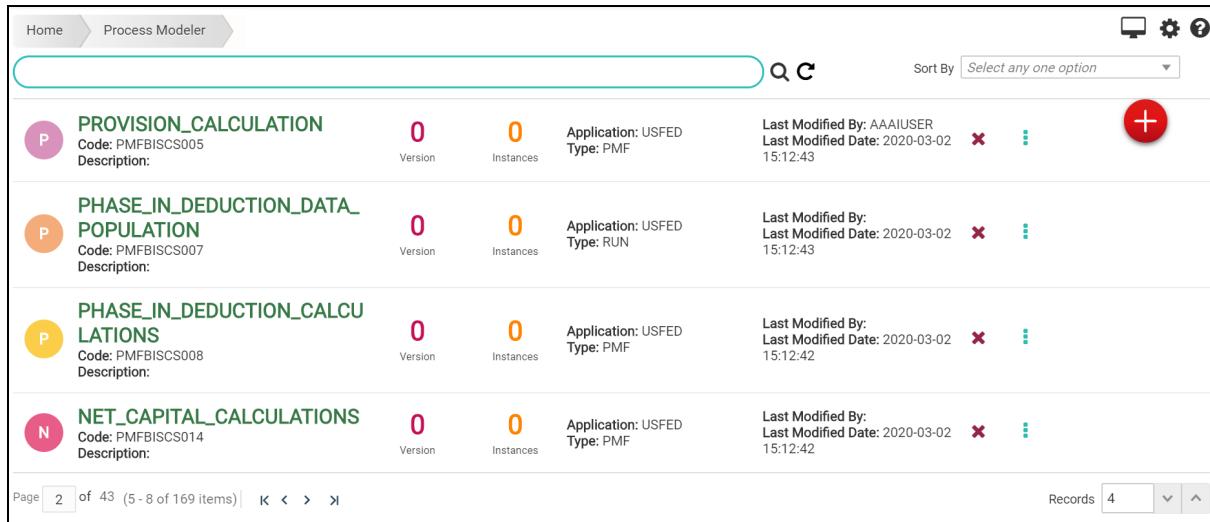
Figure 3: Process Modelling Framework tile



4. Click **Process Modeller** to launch the Process Modeller.
5. Click **Process Monitor** to monitor currently running processes.
6. Click **Delegation** to launch the Delegation framework.

2 Process Modeler

Figure 4: Process Modeler window



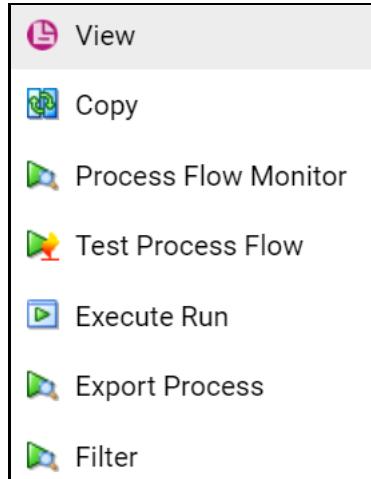
PROVISION_CALCULATION		0 Version	0 Instances	Application: USFED Type: PMF	Last Modified By: AAAIUSER Last Modified Date: 2020-03-02 15:12:43	X	...	+ (Red)
PHASE_INDEDUCTIONDATAPOPULATION		0 Version	0 Instances	Application: USFED Type: RUN	Last Modified By: Last Modified Date: 2020-03-02 15:12:43	X	...	
PHASE_INDEDUCTIONCALCULATIONS		0 Version	0 Instances	Application: USFED Type: PMF	Last Modified By: Last Modified Date: 2020-03-02 15:12:42	X	...	
NETCAPITALCALCULATIONS		0 Version	0 Instances	Application: USFED Type: PMF	Last Modified By: Last Modified Date: 2020-03-02 15:12:42	X	...	
		Page 2 of 43 (5 - 8 of 169 items)	< > >>	Records 4				

The *Process Modeler* window displays the existing Business Process Pipelines (Type: PMF) and Run Pipelines (Type: RUN) with the details such as Process ID, Process Name, Process Description, Version, Instance, Application, and Last Modified details.

You can do the following tasks from this window:

- Click  to create a new Pipeline.
- Click the Process Name link to launch and edit the Pipeline.
- Click  to delete a Pipeline.
- Click  to view the following sub-menu:

Figure 5: Process Modeler sub-menu



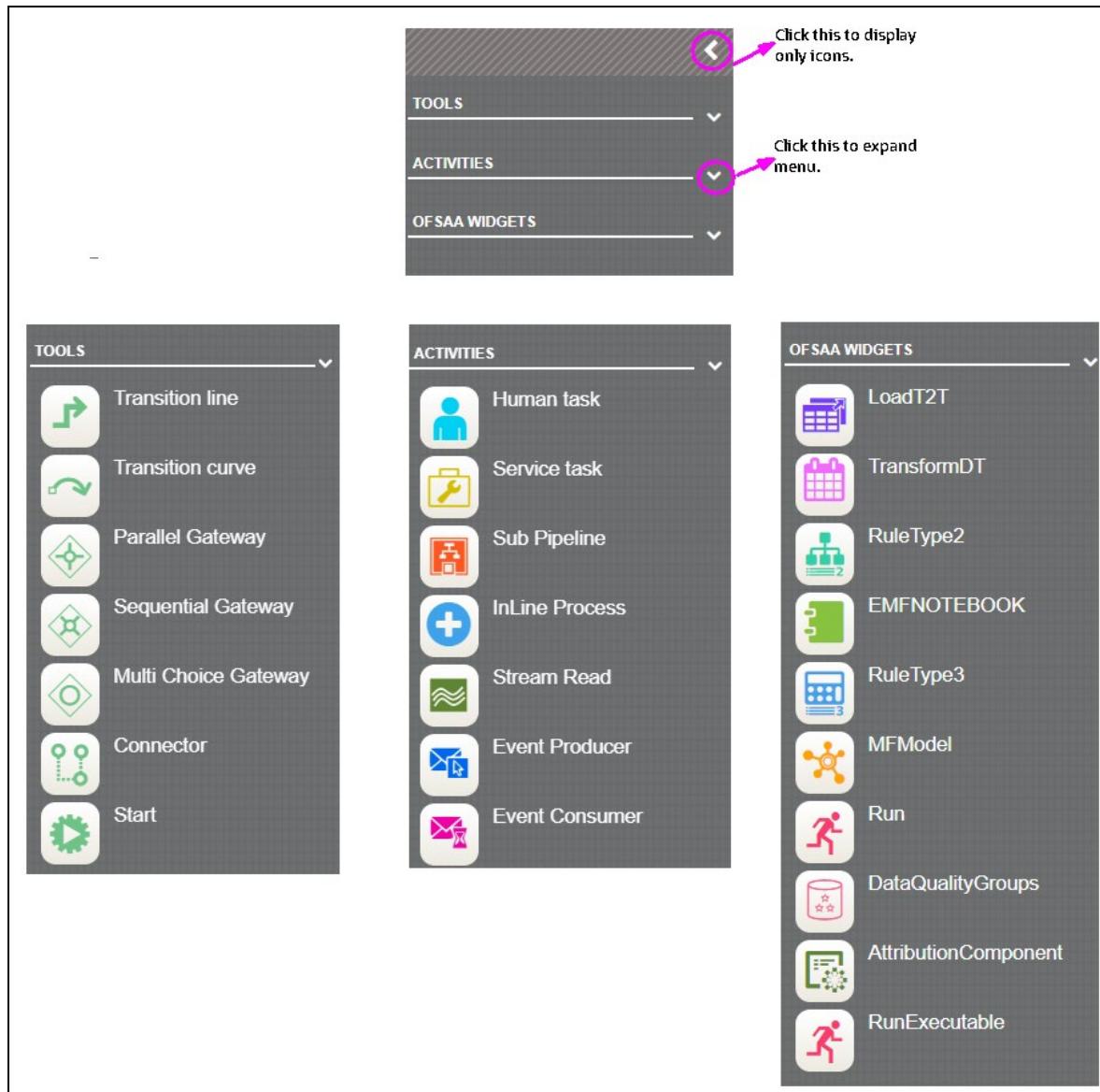
- Click **View** to see the process flow.

- Click **Copy** to [copy and create](#) a new Pipeline with the same process flow.
- Click **Process Flow Monitor** to [monitor](#) the Pipeline.
- Click **Test Process Flow** to [test or check](#) whether the Business Pipeline you designed works as expected.
- Click **Execute Run** to [execute](#) a Run Pipeline.
- Click **Export Process** to export a Process.
- Click **Filter** to [apply a filter condition](#) to a Run Pipeline.
- Using the *Search* grid, you can search for a specific Pipeline by providing a keyword from Process ID, Process Name or Process Description and clicking . Click Reset search icon  to reset the Search fields.
- You can sort the Pipelines based on Process ID, Process Name, or Application. Click the **Sort by** drop-down and select the attribute by which you want to sort.
- You can use the **Filter Pipeline** field to filter pipelines based on pipeline type. For example, if you want to view only Run Pipelines, remove **Process** from the **Filter Pipeline** field.
- Click  to go to the [Process Monitor](#) window.

3 Components for Designing Your Process Flow

The Process Flow tab has a floating toolbar and a drawing canvas. Drawing canvas is used to design the Process flow with the Tools, Activities, and OFSAA Widgets available in the floating toolbar.

Figure 6: Components for Designing your Process Flow



3.1 Transitions

Transition is used to control the flow between various components in the Process flow. Transition connects two activities and the flow is configured based on conditional expression or decision rule.

You can use Transition Line  if you want to use a straight line to connect the activities or Transition Curve  if you want to use curved lines, based on your requirement.

3.2 Gateways

Split refers to a condition where an incoming transition is split into multiple transitions. In Merge, multiple incoming transitions are merged into a single transition. The Splitting and Merging of Activities are modeled through gateways. A gateway can be Sequential, Parallel, or Multi Choice.

- **Parallel Gateway:** A Parallel gateway is used when you want to have multiple transitions/flows that should be executed in parallel.
- **Sequential Gateway:** A Sequential gateway is used when you want to have multiple transitions/flows that should be run in sequence.
- **Multi Choice Gateway:** A Multi Choice gateway is used when you want to execute multiple transitions/flows based on decision rule.

ATTENTION If you use Parallel or Multi Choice gateway in your pipeline, ensure that after all the activities added to these gateways, it is 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 empty service task activity. Otherwise, the status is not updated correctly and the next activity execution does not happen.

3.3 Connector

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

3.4 Human Task

A Human Task requires human intervention to move to the next activity. For more information on stitching human tasks in your Process flow, see the [Human Tasks](#) section.

3.5 Service Task

Service task typically invokes an application component (for example, activity to invoke a business rule to calculate a certain threshold). For more information on stitching service tasks in your Process flow, see the [Service Tasks](#) section.

3.6 Sub Pipeline

Sub Pipeline provides the reusability of Pipelines. Using the Sub Pipeline component, you can call another Pipeline from your parent Pipeline. For more information on how to use Sub Pipeline, see [Calling another Pipeline from Your Parent Pipeline](#) section.

3.7 OFSAA Widgets

OFSAA widgets are used to execute OFSAA components like T2T definitions, PLC definitions (DT), Rules (Classification Rule and Computation Rule), Models in EMF, RRF Runs, Run Executable, Data Quality Groups, and RRF Processes through Process Modeller. For more information, see [Configuring OFSAA Tasks in Your Process Flow](#) section.

You can register a new component by entering details in the AAI_WF_COMPONENT_REGISTRATION table. For more information, see the [Configuring Custom Components](#) section.

4 Artifacts of Process Modelling

Before you start designing your Pipeline, it needs to be introduced to some artifacts of the Process Modelling Framework.

- [Application Package](#)
- [Application Registration](#)
- [Data Fields](#)
- [Application Rules](#)
- [Application Object Model \(AOM\)](#)

4.1 Application Package

Application Package is a concept used to group Pipelines, Application Rules, and Data Fields that are required for an Application. When you create a Pipeline, you should select the Application Package in which the Pipeline needs to be available. Similarly, when you define a Data Field or an Application Rule, you can set it to be available across Pipelines created in that Application Package.

You can add a new package by adding a new entry in the AAI_WF_APP_PACKAGE_B table.

Figure 7: AAI_WF_APP_PACKAGE_B table

V_APP_PACKAGE_ID	V_APP_PACKAGE_DESC	V_LANGIN	V_DEFINITION_PAGE_URL	V_IS_EMAIL_REQUIRED
0	... Global N ...
10	... Business Restructure Restructure/manage_grid.jsp?userId={ASSIGNEEUSERS}&loc...	... Y ...
100	... Platform Y ...
11	... Questionnaire solution/abc_qtnr/QtnrRedirectFrmPMFInbox.jsp?appCode={app...	... Y ...

V_APP_PACKAGE_ID – Enter a unique application package ID.

V_APP_PACKAGE_DESC – Enter a description for the application package.

V_DEFINITION_PAGE_URL – Enter the URL of the definition page of the Application.

V_IS_EMAIL_REQUIRED – Set this as Y for configuring email at Application Level.

4.2 Application Registration

This section is applicable only for Business Pipeline.

Application registration is required to define the Entity and the Attributes that need to be updated for a Business Pipeline. These entries need to be seeded in the AAI_WF_APP_REGISTRATION and AAI_WF_APP_DEFINITION_MAP tables.

4.2.1 Object Type Entry

The AAI_WF_APP_REGISTRATION table stores information like the Object Type, Fact Table, Primary Key Column Name, Object Name, and Object Type Name and so on for an Application Package.

Figure 8: AAI_WF_APP_REGISTRATION table

Select aai_wf_app_package_b [Select aai_wf_app_registration] Select aai_wf_app_definition_map Select aai_wf_process_b					
V_APP_PACKAGE_ID	V_OBJECT_TYPE	V_FCT_TABLENAME	V_PRIMARY_KEY_COL_NAME	V_OBJECT_NAME_COL	V_OBJECT_TYPE_NAME
1	DFS_NGECM	... FR_EE	... kdd_cases	... CASE_INTRL_ID	... Case Management
2	DFS_NGECM	... AML_PAT	... kdd_cases	... CASE_INTRL_ID	... Case Management
3	DFS_NGECM	... AML_SURV	... kdd_cases	... CASE_INTRL_ID	... Case Management
4	DFS_NGECM	... FR_AC	... kdd_cases	... CASE_INTRL_ID	... Case Management
5	DFS_NGECM	... AML_TER	... kdd_cases	... CASE_INTRL_ID	... Case Management
6	DFS_NGECM	... AML_DD	... kdd_cases	... CASE_INTRL_ID	... Case Management
7	DFS_NGECM	... FR_ON	... kdd_cases	... CASE_INTRL_ID	... Case Management
8	DFS_NGECM	... KYC_FIRM	... kdd_cases	... CASE_INTRL_ID	... Case Management
9	DFS_NGECM	... KYC_IND	... kdd_cases	... CASE_INTRL_ID	... Case Management
10	DFS_NGECM	... KYC_CORP	... kdd_cases	... CASE_INTRL_ID	... Case Management
11	DFS_NGECM	... CS_SAN	... kdd_cases	... CASE_INTRL_ID	... Case Management
12	DFS_NGECM	... CS_EE_SAN	... kdd_cases	... CASE_INTRL_ID	... Case Management
13	DFS_NGECM	... CS_RT_SAN	... kdd_cases	... CASE_INTRL_ID	... Case Management
14	DFS_NGECM	... CS_RT_PRB	... kdd_cases	... CASE_INTRL_ID	... Case Management
15	DFS_NGECM	... CS_PEP	... kdd_cases	... CASE_INTRL_ID	... Case Management
16	DFS_NGECM	... CS_EDD	... kdd_cases	... CASE_INTRL_ID	... Case Management
17	DFS_NGECM	... CS_EE_PEP	... kdd_cases	... CASE_INTRL_ID	... Case Management
18	DFS_NGECM	... CS_EE_EDD	... kdd_cases	... CASE_INTRL_ID	... Case Management
19	DFS_NGECM	... CS_RT_PEP	... kdd_cases	... CASE_INTRL_ID	... Case Management
20	DFS_NGECM	... CS_RT_EDD	... kdd_cases	... CASE_INTRL_ID	... Case Management
21	DFS_NGECM	... CS_PRB	... kdd_cases	... CASE_INTRL_ID	... Case Management
22	DFS_NGECM	... CS_EE_PRB	... kdd_cases	... CASE_INTRL_ID	... Case Management

4.2.2 Object Type Process Mapping

The AAI_WF_APP_DEFINITION_MAP table stores Process IDs against the required Object Type. Note that multiple object types can be mapped to a single Process.

Figure 9: AAI_WF_APP_DEFINITION_MAP table

Select aai_wf_app_package_b [Select aai_wf_app_registration] Select aai_wf_app_definition_map Select aai_wf_process_b								
V_PROCESS_ID	V_VERSION	V_OBJECT_TYPE	V_DEFAULT_FLAG	V_ENABLE_FLAG	V_KBD_1	V_KBD_2	V_KBD_3	V_KBD_4
1	ECM	0 AML_DD	... Y	Y
4	ECM	0 AML_PAT	... Y	Y
2	ECM	0 AML_SURV	... Y	Y
3	ECM	0 AML_TER	... Y	Y
26	BR1	0 BR	... Y	Y
16	ECM_PEP_EDD	0 CS_EDD	... Y	Y
18	ECM_PEP_EDD	0 CS_EE_EDD	... Y	Y
17	ECM_PEP_EDD	0 CS_EE_PEP	... Y	Y
22	ECM_SAN	0 CS_EE_PRB	... Y	Y
12	ECM_SAN	0 CS_EE_SAN	... Y	Y
15	ECM_PEP_EDD	0 CS_PEP	... Y	Y
21	ECM_SAN	0 CS_PRB	... Y	Y
20	ECM_PEP_EDD	0 CS_RT_EDD	... Y	Y
19	ECM_PEP_EDD	0 CS_RT_PEP	... Y	Y
14	ECM_SAN	0 CS_RT_PRB	... Y	Y
13	ECM_SAN	0 CS_RT_SAN	... Y	Y
11	ECM_SAN	0 CS_SAN	... Y	Y
29	CUSTOMER_VERIFICATION_GATEWAY_FCCM	0 CUSTOMER_VERIFICATION_GATEWAY_FCCM	... Y	Y

4.3 Data Fields

Data Field, which is also known as Process Variable, helps Process Pipelines to access and store information from outside application. Often the process flow is based on the value of this information. In other cases, this information is the result of running the tasks in the Pipeline.

This section has the following sub sections:

- [Adding a Data Field](#)
- [System Data Fields](#)
- [AOM Data Field](#)

4.3.1 Add a Data Field

1. From the *Process Modeller* window, click  corresponding to the Pipeline for which you want to add a Data Field. The *Process Flow* tab is displayed.
2. Select the *DataFields* tab.

Figure 10: DataFields tab

Process Flow	Definition	Application Rule	DataFields		
Select	Name	Description	Type	Default Value	Is Mandatory
...	BUCKETCONVENTION	Bucket Conversion	AOM		Y
...	CONSOTYPE	Consolidation Type	AOM		Y
...	test	Datafield	STRING		
...	WF_ENTITYID	ENTITYID	STRING		
...	FIC_MIS_DATE	FIC MIS Date	AOM		Y
...	WF_INFODOM_CODE	INFODOM_CODE	STRING		
...	WF_INSTANCE	INSTANCE	STRING		
...	INTERPOLMETHOD	Interpolation Method	AOM		Y
...	WF_LOCALE	LOCALE	STRING		
...	LE	Legal Entity	AOM		Y
...	N_RUNTYPE_SK	N_RUNTYPE_SK	INT	3	
...	WF_OBJECT_ID	OBJECT_ID	STRING		
...	WF_OBJECT_NAME	OBJECT_NAME	STRING		
...	WF_OBJECT_TYPE	OBJECT_TYPE	STRING		
...	WF_OUTCOME_ID	OUTCOME_ID	STRING		
...	PD_INTERPOLATION_MTHD	PD Interpolation Method	AOM		Y
...	WF_PROCESS_ID	PROCESS_ID	STRING		
...	RCY	Reporting Currency	AOM		Y
...	V_RUN_MAIN_DESC	Run Execution Description	AOM		Y
...	WF_SEGMENT_CODE	SEGMENT_CODE	STRING		

3. Click **Add**. The *Data Field Details* window is displayed.

Figure 11: Data Field Details window

Data Field Details

Datafield Code ?	SUM_OF_BALANCES
Datafield Description ?	Sum of Balances
Datafield Type ?	String
Initial Value ?	100
Is Mandatory ?	Yes
Scope ?	Process
Ok	

4. Enter the details as given in the table:

Table 2: Data Field Details Description

Field Name	Description
Data field Code	Enter the Variable Name/Code, which needs to be used by application to read or write into this variable. This field is non-translatable.
Data field Description	Enter a brief description of the Data field.
Data field Type	Enter the Data Field type. The supported types are String, Integer, Int, Float, AOM (Application Object Model), and Boolean. For more information on where we use AOM Data Field type, see AOM Data Field section.
Initial Value	Enter the default value for the Data.
Is Mandatory	This field is applicable to the Data Field Type of AOM. By default, for AOM Data Field, this is selected as Yes . 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 from the drop-down list. The options are: <ul style="list-style-type: none"> Process- Select Process if you want to use the Data Field only in the current process. Package- Select Package if you want to use the Data Field across all the processes in the package.

- Click **OK**.

4.3.2 System Data Fields

Some data are tracked internally by the System using a predefined set of Data Fields such as Status of Process. You can access these activity instance attributes in the same way you access regular data objects, but you cannot assign them new values.

4.3.3 AOM Data Field

The AOM Data Fields are automatically created from the entries in the AAI_AOM_APP_COMP_ATTR_MAPPING table. These Data Fields, which are marked as mandatory, is displayed in the *Select Run Parameters* window while [executing Run Pipeline](#). For configuring AOM Data Fields, see [Configuring Application Object Model \(AOM\)](#) section.

4.4

Application Rules

The Application or API Rule is the interface between the process engine and the application, including any parameters to be passed.

Based on their usage the Application Rules are categorized as follows:

- **Execution Rule:** These are Business Logic executed as Task by an Activity.
- **Decision Rule:** This rule returns Boolean value “True/False”, used in decision making during split/branching of transition.
- **Selection Rule:** This rule fetches some value, useful to get value dynamically from a table or other source.

For example, `select v_created_by from fct_expenses where id=101`

Following are the available Application Rules:

- [SQL Application Rule](#)
- [Stored Procedure Application Rule](#)
- [Function Application Rule](#)
- [Java Application Rule](#)
- [Java External API Application Rule](#)
- [Outcome Rules Application Rule](#)
- [Expression Application Rule](#)
- [Rest Service Application Rule](#)
- [Attribute Expression Application Rule](#)
- [Advanced Attribute Expression Application Rule](#)
- [JSON Path Expression Application Rule](#)
- [JSON Read From DB Application Rule](#)
- [JSON Write To DB Application Rule](#)

4.4.1 SQL Application Rule

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

Figure 12: SQL Application Rule Details window

The screenshot shows the 'Rule Details' window for a SQL Application Rule. The window has a header 'Rule Details' with a close button. It contains the following fields:

- Name**: Process_Name_Data
- Rule Type**: Execution Rule
- Execution Type**: SQL
- Implementation Detail**: SELECT PROCESSNAME FROM ofsaaatm.Report_Links WHERE REPORT_TYPE='MODEL_RUN'
- Return Parameter**: PROCESS_NAME
- Scope**: Package

At the bottom is a 'Save' button.

Table 3: SQL Application Rule Details Description

Field Name	Description
Name	Enter a unique name for the SQL Application Rule.
Rule Type	Select the Rule Type from the drop-down list. The SQL 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 SQL.

Field Name	Description
Implementation Detail	<ul style="list-style-type: none"> Decision Rule- For Decision Rule the SQL 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 SQL Statement can be any DML statement. For example, update fct_expenses set expenses={EXPENSES} where id = {WF_ENTITYID} Selection Rule- For Selection Rule the SQL Statement should be a Select statement which returns a list of values. For example, select v_created_by from fct_expenses id = {WF_ENTITYID}
Return Parameter	<p>Select the Data Field that receives the return parameter of the SQL Rule, from the drop-down list.</p> <ul style="list-style-type: none"> For ExecutionRule type, the business logic is implemented in the method and the parameter value returned from the SQL Rule is saved in the mapped Data Field. In case of SelectionRule type, the Java method should be a String value. In case of DecisionRule 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.

4.4.2 Stored Procedure Application Rule

This Application Rule is used to call a Stored Procedure in your Process Flow.

Figure 13: Stored Procedure Application Rule Details window

Rule Details

Name	DQ EXE RULE
Rule Type	Execution Rule
Execution Type	Stored Procedure
Procedure Name	dqcheck
InputParameters	DQ STATUS
Parameter Mode	["OUT"]
Scope	Package
Save	

Table 4: Stored Procedure Application Rule Details Description

Field Name	Description
Name	Enter a unique name for the Stored Procedure Application Rule.
Rule Type	Select the Rule Type from the drop-down list. The Stored Procedure 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 Stored Procedure.
Procedure Name	Enter the Stored Procedure Name.
Input Parameters	Select the list of Data Fields that are passed as input parameters, from the drop-down list.

Field Name	Description
Parameter Mode	<p>Enter the Parameter Mode in JSON format. For example, suppose you have given 3 parameters as input parameters, enter parameter mode as <code>["IN","IN","OUT"]</code>. During Execution of Stored Procedure,</p> <ul style="list-style-type: none"> • In case of Decision Rule type, the first return parameter should return value 'PASS' for success evaluation. • In case of Selection Rule type, the first return parameter value is taken as Selection data. • In case of Execution Rule, the procedure return OUT parameter value overwrites the current value of respective mapped Data Field.
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.

4.4.3 Function Application Rule

This Application Rule is used to call Database functions in your Process Flow.

Figure 14: Function Application Rule Details window

The screenshot shows the 'Rule Details' dialog box. The fields are as follows:

- Name:** Getting Additional Param
- Rule Type:** Execution Rule
- Execution Type:** Function
- Function Name:** FN_DQ_ADDITIONAL_PARAMS
- InputParameters:** OBJECT_ID, USERID, OBJECT_NAME
- Parameter Mode:** (empty)
- Return Parameter:** TASK_STATUS
- Scope:** Process

Table 5: Function Application Rule Details Description

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, Execution Rule and Selection Rule.
Execution Type	Displays the Application Execution Type as Function.
Function Name	Enter the Function Name.
Input Parameters	Select the list of Data Fields which will be passed as input parameters, from the drop-down list.
Parameter Mode	<p>Enter the Parameter Mode in JSON format. For example, suppose you have given 3 parameters as input parameters, enter parameter mode as [“IN”,“IN”,“OUT”].</p> <p>During Execution of Function,</p> <ul style="list-style-type: none"> • In case of Decision Rule type, the first return parameter should return value ‘PASS’ for success evaluation. • In case of Selection Rule type, the first return parameter value is taken as Selection data. • In case of Execution Rule, the procedure return OUT parameter value overwrites the current value of respective mapped Data Field.
Return Parameter	<p>Select the Data Field that receives the return parameter of the Java function, from the drop-down list.</p> <ul style="list-style-type: none"> • For ExecutionRule type, the business logic is implemented in the method and the parameter value returned from the Java method is saved in the mapped Data Field. • In case of SelectionRule type, the Java method should be a String value. • In case of DecisionRule 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.

4.4.4 Java Application Rule

This Application Rule is used to call Java functions in your Process flow.

Figure 15: Java Application Rule Details window

Rule Details

Name	CallECMEventCreationService
Rule Type	Execution Rule
Execution Type	JAVA
Implementation Detail	com.ofss.fccm.fraud.main.ResponseMain
Return Parameter	ENTITYID
Scope	Process
Save	

Table 6: Java Application Rule Details Description

Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	This Rule Execution type supports only ExecutionRule type.
Execution Type	Displays the Application Execution Type as JAVA.
Implementation Detail	<p>Enter the complete java class name which implements the Interface : com.ofs.aai.service.wf.external.base.Activity.</p> <p>The implementation class has to override the method with the Business Logic. executeTask(List<Data Field> Data Fields)</p> <p>The Data Fields are passed by reference, so changes can be made in Data Fields value directly by the implementation class, which will be recognized by the WorkflowEngine.</p>

Field Name	Description
Return Parameter	Select the Data Field that receives the return parameter of the Java function, from the drop-down list. <ul style="list-style-type: none"> For ExecutionRule type, the business logic is implemented in the method and the parameter value returned from the Java method is saved in the mapped Data Field. In case of SelectionRule type, the Java method should be a String value. In case of DecisionRule 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.

NOTE

The class and its dependent file (or jar) need to be available in the webcontainer classpath.

For example, <TOMCAT_HOME>/webapps/<context>/WEB-INF/lib/<**forecast.jar**>

4.4.5 Java External API Application Rule

This Application Rule is used to call Java External API in your process flow. You need to specify the Class Name and the method of the API.

Figure 16: Java External API Application Rule Details window

Rule Details

Name	ForecastPD
Rule Type	Execution Rule
Execution Type	Java External API
Class Name	com.Forecast
Method	forecastPD
InputParameters	INFODOM_CODE X
Return Parameter	STATUS
Scope	Process
Save	

Table 7: Java External API Application Rule Details Description

Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the Rule Type from the drop-down list. The Java External API 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 JAVA External API.
Class Name	Enter the complete java class name that implements the Business Logic.
Method	Enter the method that you want to execute.
Input Parameters	You can pass Input Parameters for the method using Data Fields. Select the required Data Fields from the drop-down list.
Return Parameters	Select the Data Field that stores the Return Parameter of the method, from the drop-down list.

Field 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.

NOTE

1. The class and its dependent file (or jar) need to be available in webcontainer classpath.
For example,
<TOMCAT_HOME>/webapps/<context>/WEB-INF/lib/<**forecast.jar**>
2. For details on the supported APIs for using as Java External APIs, see [Appendix B](#).

4.4.6 Outcome Rules Application Rule

Figure 16: Outcome Rule Details window

The screenshot shows the 'Rule Details' dialog box. It has a title bar 'Rule Details' with a close button. Inside, there are five input fields with labels and dropdowns:

- Name**: An empty text input field.
- RuleType**: A dropdown set to 'DecisionRule'.
- ExecutionType**: A dropdown set to 'Outcome'.
- Outcomes**: A dropdown with 'Approve' selected.
- Scope**: A dropdown set to 'Process'.

At the bottom is an 'Ok' button.

Table 8: Outcome Rule Details Description

Field Name	Description
Name	Enter a unique name for the Application Rule.

Field Name	Description
Rule Type	Displays the rule type as DecisionRule. This Rule Execution type supports only DecisionRule type.
Execution Type	Displays the Application Execution Type as Outcome.
Outcomes	Select the outcome for which you want to add the rule.
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.

4.4.7 Expression Application Rule

This is same as the SQL execution type. You need to specify only the where clause in the Expression field. It can be any SQL expressions including 'AND'/'OR'.

Figure 17: Expression Rule Details window

The screenshot shows the 'Rule Details' window for an Expression Application Rule. The window has the following fields and values:

- Name:** Restr_Batch_Initiate_Exp
- Rule Type:** Decision Rule
- Execution Type:** Expression
- Expression Type:** SQL
- Expression:** '{RESTR_NEXT_STAGE}'='EI'
- Return Parameter:** (empty)
- Scope:** Process

At the bottom of the window is a 'Save' button.

Table 9: Expression Rule Details Description

Field 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 that receives the return parameter of the Expression, from the drop-down list. <ul style="list-style-type: none"> 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 DecisionRule type, the Application Rule 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.

4.4.8 Rest Service Application Rule

This Application Rule is used to call any Rest services (internal or external) in your Process flow.

Figure 18: Rest services Rule Details window

Edit API Details

RuleType	ExecutionRule
ExecutionType	Rest Service
Method Type	POST
Url	http://[REDACTED] oracle.com:4444/ofsa808/rest-api/v1/PMFService/startWorkflowProcess
Authorization Type	Basic Auth
Authorization	dXNlcjE6cGFzc3dvcmQy
Query Param	
Headers	{Content-Type:application-json}
Data	{ "SummaryPayload": { "objectid": "Rest4321", "objecttype": "MOCK" } }
Return Parameter	
Scope	Process
Is Proxy Required	<input checked="" type="checkbox"/> Yes
Save	

Table 10: Rest services Rule Details Description

Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the Rule Type from the drop-down list. This 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 Rest service.
Method Type	Select the method type from the drop-down list. The options are GET and POST .
URL	Enter the REST URL that needs to be called. For example, <IP Address/hostname of the Web Server>:<servlet port>/<context name>/restPMF/PMFService/startWorkflowProcess (A rest URL to start the workflow).

Field Name	Description
Authorization Type	<p>Select the authorization type from the drop-down list. The options are:</p> <ul style="list-style-type: none"> • No Auth- Select this option for rest services that does not need authorization header. • Basic Auth- Select this option if you want to authenticate the invocation of Rest service.
Authorization	<p>This field is displayed only if you have selected Basic Auth as Authorization Type.</p> <ul style="list-style-type: none"> • In case of OFSAA local user, enter the User ID only. • In case of an external user, enter the base 64 encoded string. For more information, see Authentication of Rest Service. • In case this field is left blank, logged in user credentials will be taken as authorization header. <p>Note: If Authorization is given in the Header explicitly, then it will take preference than the value given in the Authorization field.</p>
Query Param	<p>Enter the Query Parameters that needs to be passed to the rest API.</p> <p>For example, http://example.com/foo?bar</p>
Headers	<p>Enter any headers that need to be passed to the rest API.</p> <p>For example, "content-type": "application/json"</p> <p>To pass the header values dynamically, use the following format:</p> <p>{Content-Type:~~TYPE~~,Authorization:~~CREDENTIAL~~}</p> <p>Where TYPE and CREDENTIAL are data fields.</p>
Data	<p>Enter if any actual data that needs to be passed to the rest API. Data can be of type RAW, JSON, Form Data and so on.</p> <p>For example, " {\n \"objectid\": \"912\", \n \"objecttype\": \"1000\", \n \"infodom\": \"OFSCAPADQINFO\", \n \"segment\": \"OFSCAPADQINFO\", \n \"userid\": \"ORMUSER\", \n \"locale\": \"en_US\", \n \"securityMap\" : {}, \n \"applicationParams\" :{\n \"testparam\": \"value1\", \n \"testparam2\": \"value2\" \n } \n \n }</p>
Return Parameter	<p>Select the Data Field which will receive the return parameter of the Expression, from the drop-down list.</p> <ul style="list-style-type: none"> • 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 DecisionRule type, the Application Rule should return Boolean values "True/False".
Scope	<p>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.</p>

Field Name	Description
Is Proxy required	Select Yes if proxy is required for the Rest Service. That is, if the Rest API is outside OFSAA. For information on how to enable proxy, see Enabling Proxy for the REST Service Application Rule section.

4.4.8.1 Enabling Proxy for the REST Service Application Rule

This section explains how to configure the Proxy details if it is required for the Rest Service Application Rule.

Add the following entries in the `AAI_WF_GLOBAL_SETTINGS` table:

Table 11: AAI_WF_GLOBAL_SETTINGS Entries Table

V_PARAM_NAME	V_PARAM_VALUE	Description
PROXY_SERVER_IP	For example, <code>www.proxy.myserver.com</code>	Provide the IP address of the Proxy server.
PROXY_SERVER_PORT	For example, 80	Provide the port number of the Proxy server.

4.4.8.2 Authentication of Rest Service

Basic auth is supported for authentication of the rest service. You have to encode your username and password using the Online encoder (<https://www.base64encode.org/>) and add the encoded value in the **Authorization** field in the *Rule Details* window. The format of the user name and password to be entered in the online encoder should be `username:password`. For example, if we give `DAVID_MLRO:oracle1`, it is converted to `"REFWSURfTUxSTzpvcnFjbGUx"`.

You can also add this as a parameter `V_PARAM_1` in the `AAI_WF_APPLICATION_API_B` table. This needs to be entered as given in the following figure.

Figure 19: AAI_WF_APPLICATION_API_B table

<pre>SELECT * FROM aai_wf_application_api_t1 where v_app_api_name like 'CaseCreationServiceFwupcase'; SELECT * FROM aai_wf_application_api_b where v_app_api_id= '1549543937135'; --REFWSURfTUxSTzpvcnFjbGUx --("username":"DAVID_MLRO","password":"oracle1")</pre>

4.4.9 Attribute Expression Application Rule

Figure 20: Attribute Builder window

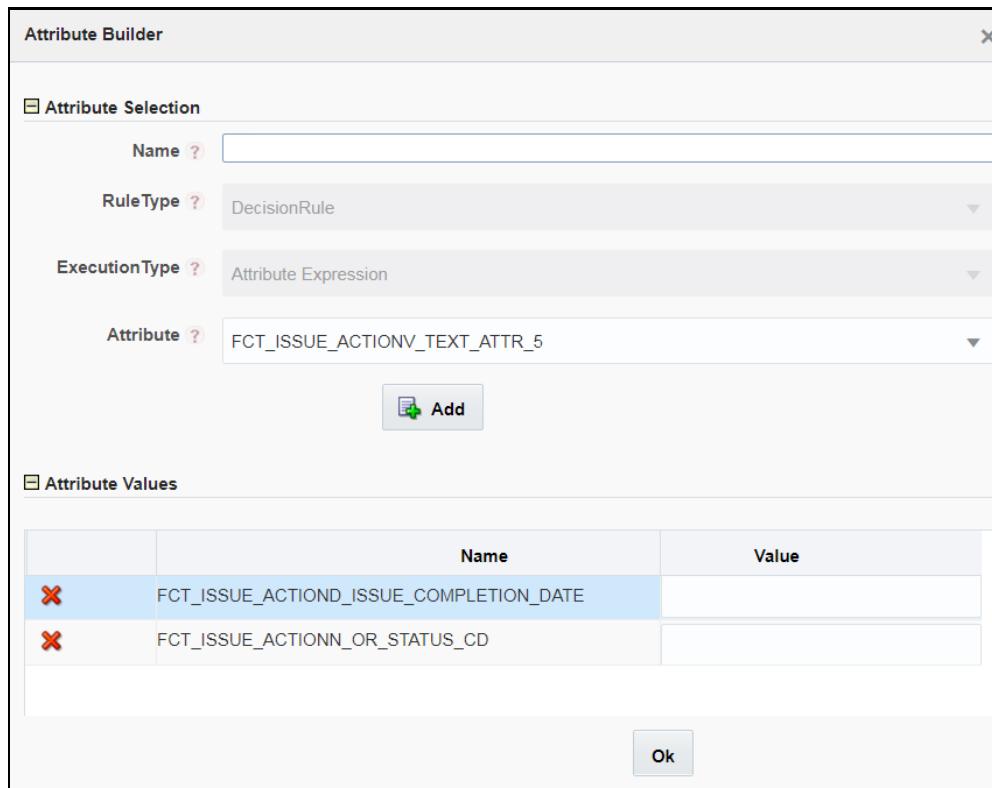


Table 12: Attribute Builder Description

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.

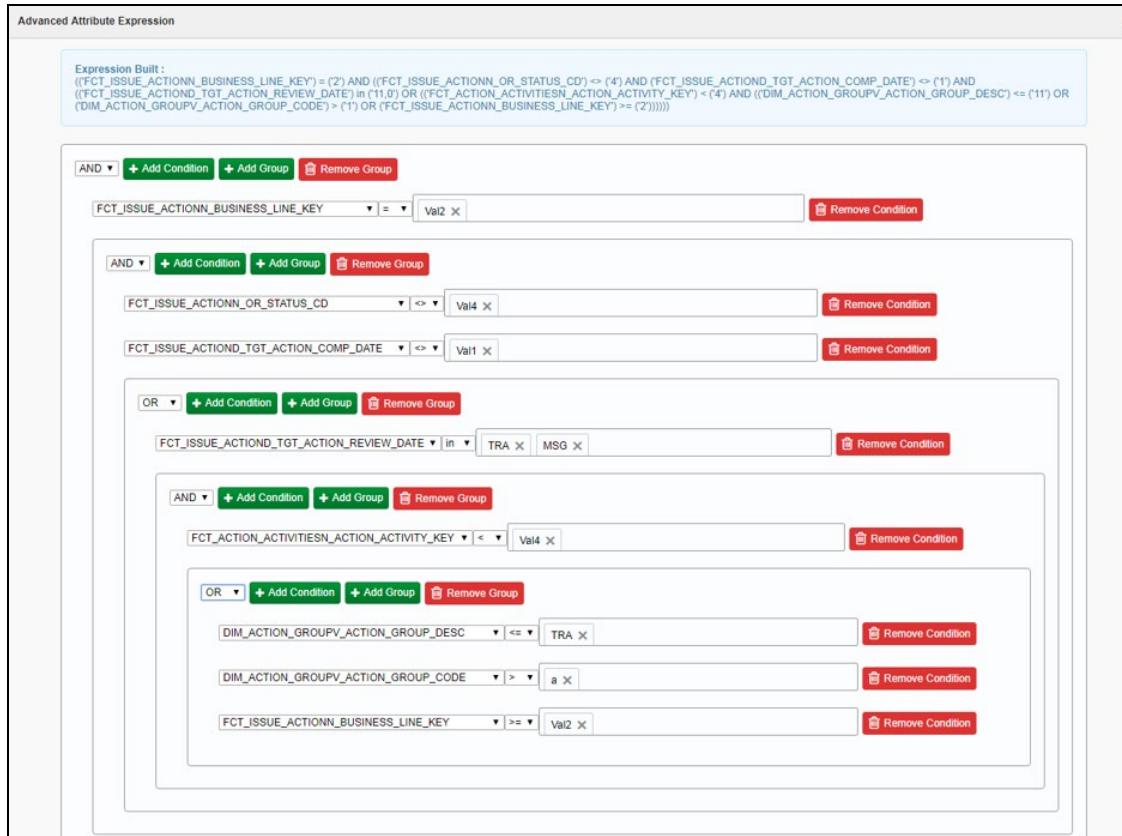
Field Name	Description
Attribute	<p>Select the attribute for which you want to define the application rule, from the drop-down list. The list displays the attributes configured for the selected application and component. For more information, see Configuring Application Object Model (AOM) section.</p> <p>Click Add to add values to the selected attributes. A row is added in the <i>Attribute Values</i> pane. Click the Value column to select the values for the attribute from the drop-down. You can select one or more values.</p> <p>You can delete a row by clicking  button.</p> <p>You can select multiple attributes and click Add to assign values to those attributes.</p>

4.4.10 Advanced Attribute Expression Application Rule

This is an advanced version of Attribute Expression Application Rule with additional logical conditions and assignment operators. The expression can be dynamically built and returns 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.

Figure 21: Advanced Attribute Expression window



The screenshot shows the 'Advanced Attribute Expression' window with the following details:

Expression Built:

```
((FCT_ISSUE_ACTIONN_BUSINESS_LINE_KEY) = ('2') AND ((FCT_ISSUE_ACTIONN_OR_STATUS_CD) <> ('4') AND (FCT_ISSUE_ACTIONID_TGT_ACTION_COMP_DATE) <> ('1') AND (FCT_ISSUE_ACTIONID_TGT_ACTION_REVIEW_DATE) in ('11,0') OR ((FCT_ACTION_ACTIVITIESN_ACTION_ACTIVITY_KEY) <> ('4') AND ((DIM_ACTION_GROUPV_ACTION_GROUP_DESC) <= ('11') OR (DIM_ACTION_GROUPV_ACTION_GROUP_CODE) > ('1') OR (FCT_ISSUE_ACTIONN_BUSINESS_LINE_KEY) >= ('2')))))
```

Logical Structure:

- Top Level:** AND
 - FCT_ISSUE_ACTIONN_BUSINESS_LINE_KEY = Val2
- Second Level (Nested AND):**
 - AND
 - FCT_ISSUE_ACTIONN_OR_STATUS_CD <> Val4
 - FCT_ISSUE_ACTIONID_TGT_ACTION_COMP_DATE <> Val1
- Third Level (Nested OR):**
 - OR
 - IN (FCT_ISSUE_ACTIONID_TGT_ACTION_REVIEW_DATE) in TRA, MSG
 - AND
 - FCT_ACTION_ACTIVITIESN_ACTION_ACTIVITY_KEY <= Val4
- Fourth Level (Nested OR):**
 - OR
 - AND
 - DIM_ACTION_GROUPV_ACTION_GROUP_DESC <= TRA
 - DIM_ACTION_GROUPV_ACTION_GROUP_CODE > a
 - FCT_ISSUE_ACTIONN_BUSINESS_LINE_KEY >= Val2

Table 13: Advanced Attribute Expression Description

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.
Add Condition	<p>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.</p> <ul style="list-style-type: none"> Attribute- The drop-down list displays the attributes configured for the selected application and component. For more information, see Configuring Application Object Model (AOM) section. Operator- Available options are in,=,<>,<,<=,>,>=. Value- Displays the values configured for the selected attributes. Select the required value. <p>Click Remove Condition to delete an already added condition.</p>
Add Group	<p>Click Add Group if you want to have nested conditions. For each group, select the required logical operator as AND or OR.</p> <p>Click Remove Group to delete a group of conditions.</p>

4.4.11 JSON Path Expression Application Rule

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.

Figure 22: JSON Path Expression Rule Details window

Rule Details

Name	Get Customer Id
RuleType	ExecutionRule
ExecutionType	JSON Path Expression
JSON input	OB_CUST_INFO
JSON Path Expression	\$.CustomerUniqueId
Output Datafield	CUST_SEQ_ID
Scope	Process
Covert to Type	JSON ARRAY
Save	

Table 14: JSON Path Expression Rule Details Description

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 is compared with RHS expression and rule returns as 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. For more information, see the JsonPath Expressions section.
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 <=.

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: <ul style="list-style-type: none"> • 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.

4.4.12 JSON Read From DB Application Rule

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

Figure 23: JSON Read from DB Rule Details window

JSON Read from DB

Name: Read DB Data

RuleType: ExecutionRule

ExecutionType: JSON Read From DB

Table Name: DIM_ACCOUNT

Column List: N_ACCT_SKEY Account_ID,v_account_desc Account_Nar

Where Condition: V_PROD_CODE='CME'

Return JSON Type: JSON Array

Output Datafield: JSON_DB_DATA

Scope: Process

Save

Table 15: JSON Read from DB Rule Details Description

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. <ul style="list-style-type: none"> • 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.
Scope	Select the scope of the Application Rule from the drop-down list. The options are: <ul style="list-style-type: none"> • 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.

4.4.13 JSON Write To DB Application Rule

This Application Rule is used to write the data in JSON format into the database.

Figure 24: JSON Write to DB Rule Details window

The screenshot shows the 'Rule Details' window with the following configuration:

- Name:** WriteArrayToDB
- Rule Type:** ExecutionRule
- Execution Type:** JSON Write To DB
- Table Name:** DIM_COMPONENT_INFO
- Source JSON:**

```
[{"N_COMP_ID": "1", "N_COMP_CODE": "1", "V_COMP_NAME": "RAO R"}, {"N_COMP_ID": "2", "N_COMP_CODE": "2", "V_COMP_NAME": "Compliance Investigation"}]
```
- Output Datafield:** (empty)
- Scope:** Process

Table 16: JSON Write to DB Rule Details Description

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 Write To DB.
Table Name	Enter the table name to which you want to write the data in JSON format.
Source JSON	Enter the data in the JSON format that you want to write to the database.
Output DataField	This field is not applicable.

Field Name	Description
Scope	<p>Select the scope of the Application Rule from the drop-down list. The options are:</p> <ul style="list-style-type: none"> • 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.

4.5 Configuring Application Object Model (AOM)

This module helps in creating a set of attributes for a given application in an abstract way so that frameworks like PMF and other modules can leverage to retrieve application attributes and their values.

Each application is identified using an application package ID. For configuring package IDs, see the [Application Package](#) section.

Against each package id, the set of attributes needs to be seeded in the “AAI_AOM_APP_COMP_ATTR_MAPPING” table.

4.5.1 AAI_AOM_APP_COMP_ATTR_MAPPING Table

In this table, make entries for each attribute as given in the following table:

Table 17: AAI_AOM_APP_COMP_ATTR_MAPPING Table Description

Column Name	Description
APP_COMP_ATTR_MAP_ID	<p>Enter a unique ID for the attribute.</p> <p>You need to enter the Attribute name and description for each attribute ID entered here in the AAI_AOM_APP_COMP_ATTR_TL table. See the AAI_AOM_APP_COMP_ATTR_TL Table section.</p>
V_ATTR_CODE	Name of the attribute.
N_ATTR_TYPE_ID	<p>The ID of the attribute type.</p> <p>The values of the attributes are fetched based on the attribute type.</p> <p>1001- Static 1002- SQL Query 1003- JavaAPI 1004- Hierarchy 1005- Multi Select Hierarchy 103- Date field 102- Text box field</p> <p>For more information, see Attribute Types.</p>

Column Name	Description
V_ATTRIBUTE_VALUE1 V_ATTRIBUTE_VALUE2	Values to be fetched for the attribute. Based on the attribute type, you need to pass the values.
N_APP_ID	Application code for which the current attribute is configured. For example, if you are configuring Run execution parameters for IFRS application, enter the application ID of IFRS here.
N_COMP_ID	Component code for which the attribute is configured.
V_UDP_CODE	Special property used by applications (user-defined). For example, 'GET_STATUS' -to get the status for the workflow.
V_ATTR_CONTROL_TYPE	Enter the Control type ID to be used for the attribute. For example, 3 is used for drop-down list, 7 for textbox, 11 for date control, 41 is for hierarchy and 42 for Multi Select hierarchy

4.5.2 Example for Run Pipeline

Following figure shows an example of entries in AAI_AOM_APP_COMP_ATTR_MAPPING table for Run Pipeline:

Figure 25: AAI_AOM_APP_COMP_ATTR_MAPPING table

AAI_AOM_APP_COMP_ATTR_MAPPING											
APP_COMP_ATTR_MAP_ID	N_ATTRIB	V_ATTR_CODE	N_ATTR_TYPE_ID	V_ATTRIBUTE_VALUE1	V_ATTRIBUTE	N_APP_ID	N_COMP_ID	V_UDP_CODE	V_ATTR_CONTROL_TYPE		
1	134	LE	1004	HLLFP029	...	DFS_LLFP	-1	GET_RUN_PARAMS	41		
2	135	CONSOTYPE	1001	1	...	DFS_LLFP	-1	GET_RUN_PARAMS	3		
3	136	RCY	1004	HLLFP026	...	DFS_LLFP	-1	GET_RUN_PARAMS	41		
4	137	STAGEDETERRUN	1001	4	...	DFS_LLFP	-1	GET_RUN_PARAMS	7		
5	138	FIC_MIS_DATE	103	DFS_LLFP	-1	GET_RUN_PARAMS	11		
6	139	V_RUN_MAIN_DESC	102	DFS_LLFP	-1	GET_RUN_PARAMS	7		

4.5.3 Attribute Types

The values of attributes are fetched based on the attribute types. Following are the attribute types with their IDs:

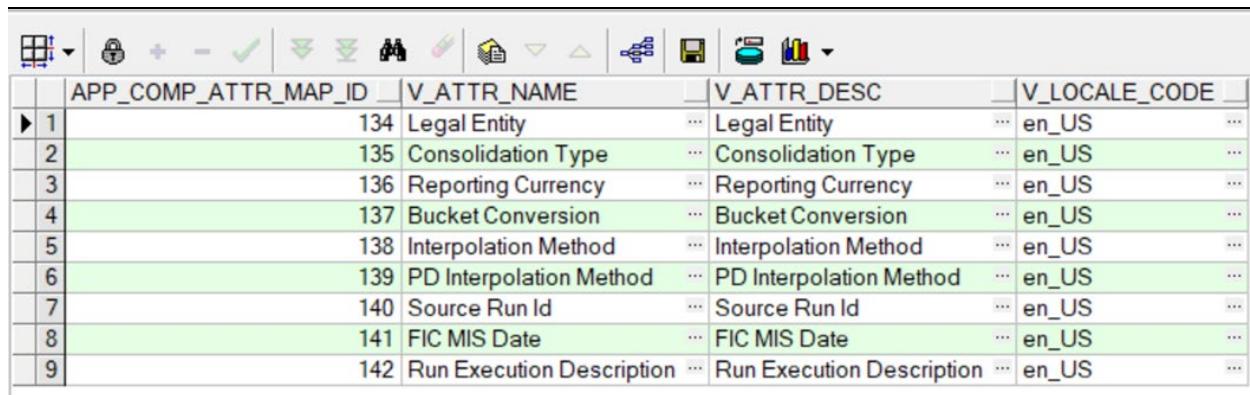
- **1001 (Static)** - Store attribute values in the AAI_AOM_STATIC table as V_STATIC_ID and V_STATIC_VAL.
- **1002 (Query)** - Enter the SQL query in V_ATTRIBUTE_VALUE1 in the AAI_AOM_APP_COMP_ATTR_MAPPING table, which has to be fired to fetch the attribute values.
- **1003 (JavaAPI)** – Enter the method that is configured for V_ATTRIBUTE_VALUE1 for the required attribute . The configured method in the classpath is invoked to get the attribute values in this case.
- **1004 (Hierarchy)** – Specify the Hierarchy code to be fetched in V_ATTRIBUTE_VALUE1 in the AAI_AOM_APP_COMP_ATTR_MAPPING table.

- **1005 (Multi Select Hierarchy)** – Specify the Multi Select Hierarchy entries in the AAI_AOM_APP_COMP_ATTR_TL table.
- **103 (DATE)** - This is used for configuring FIC_MIS_DATE. The V_ATTR_CONTROL_TYPE value should be 11.
- **102 (Text Box)** - This is used for the Description field. The V_ATTR_CONTROL_TYPE value should be 7.

4.5.4 AAI_AOM_APP_COMP_ATTR_TL Table

In this table, for each APP_COMP_ATTR_MAP_ID table, enter the locale specific Attribute Name, Description as shown:

Figure 26: APP_COMP_ATTR_MAP_ID table



	APP_COMP_ATTR_MAP_ID	V_ATTR_NAME	V_ATTR_DESC	V_LOCALE_CODE
▶	1	134	Legal Entity	en_US
	2	135	Consolidation Type	en_US
	3	136	Reporting Currency	en_US
	4	137	Bucket Conversion	en_US
	5	138	Interpolation Method	en_US
	6	139	PD Interpolation Method	en_US
	7	140	Source Run Id	en_US
	8	141	FIC MIS Date	en_US
	9	142	Run Execution Description	en_US

4.5.5 Usage of AOM Attributes in Run Pipeline

While executing Run pipeline, the Select Run Params window displays the AOM fields that are marked as mandatory.

For example, for the attributes stored in APP_COMP_ATTR_MAP_ID table as shown in the previous figure, the Select Run Params window is displayed as shown:

Figure 27: Select Run Params window

Select Run Params

TASK_STATUS	<input type="text"/>
Bucket Conversion	Mid of Bucket
PD Interpolation Method	Non-Linear Geometric
Interpolation Method	Cubic Spline
FIC MIS Date	<input type="text"/> mm/dd/yy <input type="button" value="Calendar"/>
Consolidation Type	Solo
Source Run Id	<input type="text"/>
Legal Entity	<input type="text"/> <input type="button" value="X"/>
Reporting Currency	<input type="text"/> <input type="button" value="X"/>
Run Execution Description	<input type="text"/>
<input type="button" value="OK"/>	

5

Design a Pipeline

Business pipelines are defined in OFSAA to design and execute the sequence of tasks that are either OFSAA tasks or external tasks, to derive a well-defined outcome. This flow is defined by using various OFSAA artifacts from the component toolbar.

Using Process Modeler, we can perform as follows:

- Orchestrate a Business pipeline.
- Orchestrate a Run pipeline using PMF modeling.

5.1

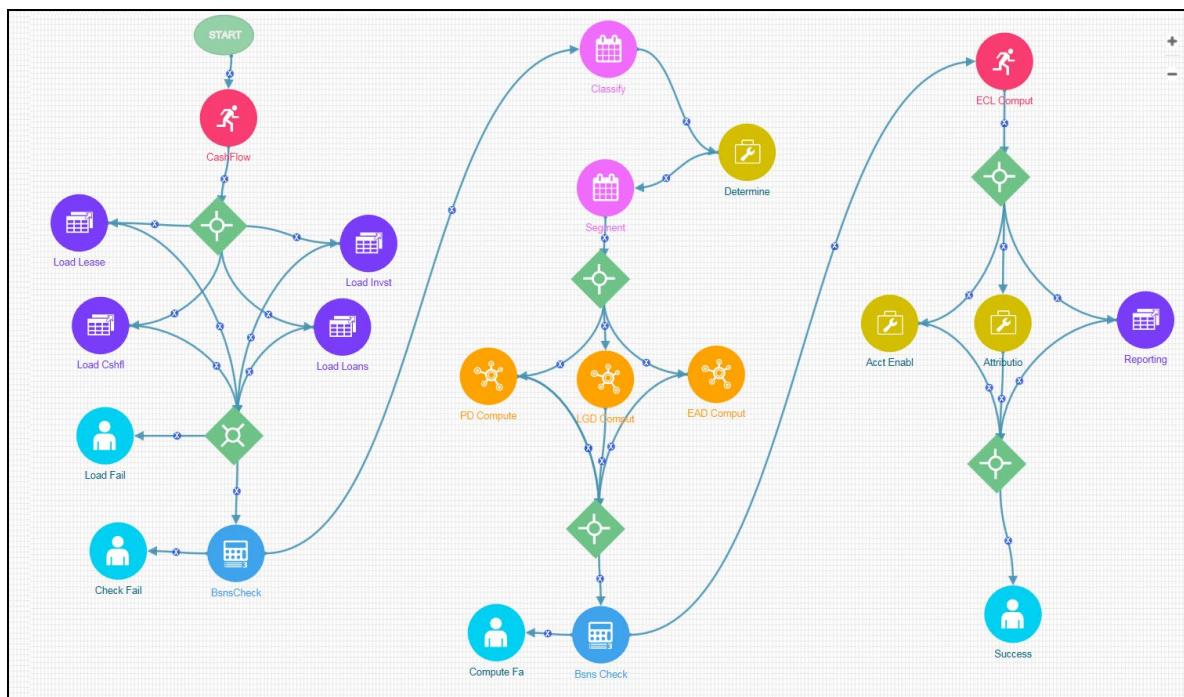
Orchestrate a Business Pipeline

Business Pipeline is used to design a Business Process that consists of a sequence of tasks either internal or external tasks through well-defined interfaces. Using the designer, you can design the entire business flows consisting of various types of tasks or another business pipeline.

5.1.1

An Example of a Business Pipeline

Figure 28: Business Pipeline example



In the example shown, we use various OFSAA widgets like Run, LoadT2T, RuleType3, TransformDT, MFModel, and activities like Human Tasks and Service tasks, which are related to each other through transitions. For executing tasks in parallel, we used Parallel Gateways.

5.1.2

Creating a Business Pipeline

Click  in the *Process Modeler* Summary window.

Figure 29: Process Details window to create a Business Pipeline

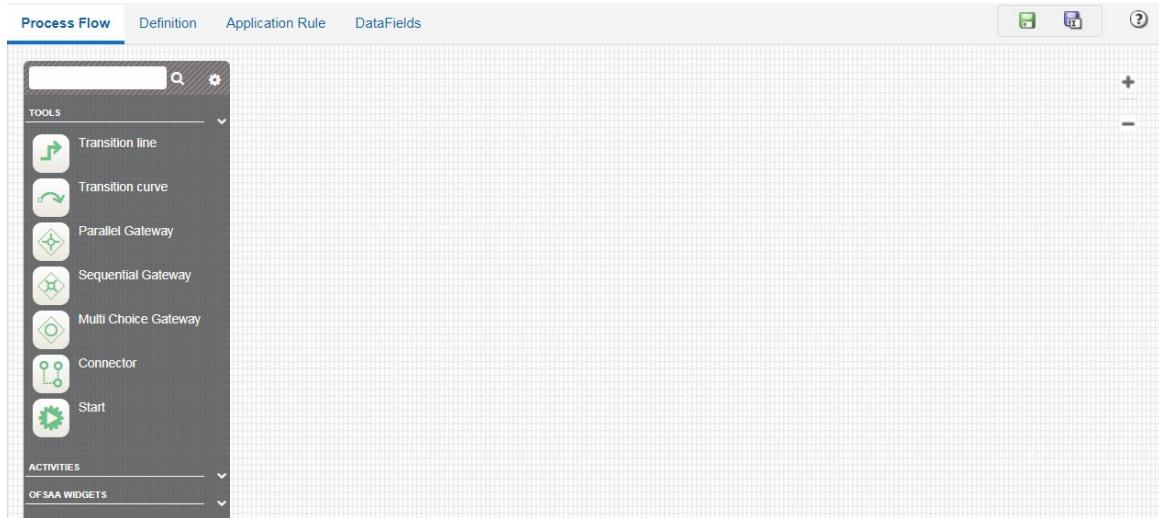
The screenshot shows the 'Process Details' window with the following fields:

- Process ID: MD
- Process Name: Model_Deployment
- Process Description: Model Deployment
- App Package ID: Platform
- Type: Business Process Pipeline
- Registered Topics: (empty dropdown)
- Spark DB: (toggle switch)
- Infodom: OFSAAINFO
- Tag: (empty dropdown)
- Segment: INDSEG

At the bottom are two buttons: 'Save And Close' and 'Save And Launch'.

1. Enter a unique Process ID, Process Name and a description.
2. Select the appropriate app package in which you want to create the process. For more information, see [Application Package](#).
3. Select **Business Process Pipeline** from the **Type** drop-down list.
4. Select the information domain in which you want to create the Business Pipeline, from the **Infodom** drop-down list. The list displays all the Infodoms that are mapped to the applications configured in your OFSAA instance.
5. Click **Save and Launch**. The Process Flow canvas is displayed.

Figure 30: Process Flow canvas



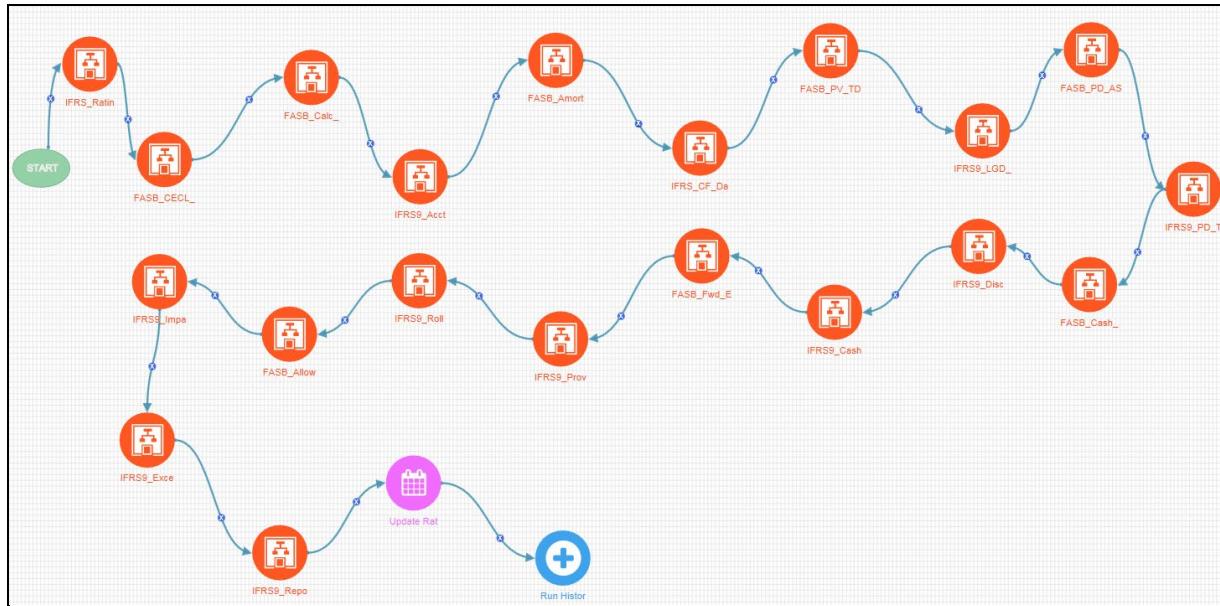
6. Click  **Start** from the *Tools* pane in the floating toolbar and click on the drawing canvas where you want to place it. This Start activity indicates the beginning of the Process.
7. Design your Process with various components available in the *Process Flow* tab. For more information on each component, see [Components for Designing Your Process Flow](#) section.

5.2 Run Pipeline

Run Process is used to create a Run definition in Rule Run Framework (RRF) using PMF Process. Visual representation of the Run is enabled through PMF by the construction of a Run Pipeline. Various OFSAA widgets that enable the construction of a Run Pipeline are available in the Component toolbar.

5.2.1 An Example of Run Pipeline

Figure 31: Run Pipeline Example



5.2.2 Creating a Run Pipeline

1. Seed AOM Data Fields. For more information, see [Configuring Application Object Model \(AOM\) section](#).
2. Click in the *Process Modeller* Summary window.

Figure 32: Process Details window to create a Run Pipeline

Process Details

Process ID	53756756
Process Name	CECL_RUN
Process Description	CECL_RUN
App Package ID	Platform
Type	Run Pipeline
Registered Topics	
Spark DB	<input type="checkbox"/>
Infodom	OFSAAAINFO
Tag	
Segment	ORECSEG
<input type="button" value="Save And Close"/> <input type="button" value="Save And Launch"/>	

3. Enter a unique Process ID, Process Name and a description.

4. Select the appropriate app package in which you want to create the process. For more information, see the [Application Package](#) section.
5. Select **Run Pipeline** from the **Type** drop-down list.
6. Select the information domain in which you want to create the Run Process, from the **Infodom** drop-down list. The list displays all the Infodoms that are mapped to the applications configured in your OFSAA instance.
7. Click **Save & Launch**.
8. Click  **Start** from the *Tools* pane in the floating toolbar and click on the drawing canvas where you want to place it. This Start activity indicates the beginning of the Process.

5.2.3 Design your Run Pipeline using OFSAA Widgets

You can construct a Run pipeline using only OFSAA Widgets. For details, see [Configuring OFSAA Tasks in Your Process Flow](#).

5.2.4 Design your Run Pipeline using Sub Pipeline

You can construct Run Pipeline using already constructed pipelines. The same pipeline can be used across different Run pipelines. For more information, see the [Calling another Pipeline from Your Parent Pipeline](#) section.

5.2.5 Design your Run Pipeline using combinations of OFSAA Widgets and Sub Pipeline

You can construct Run pipeline by combining OFSAA widgets and Sub pipeline.

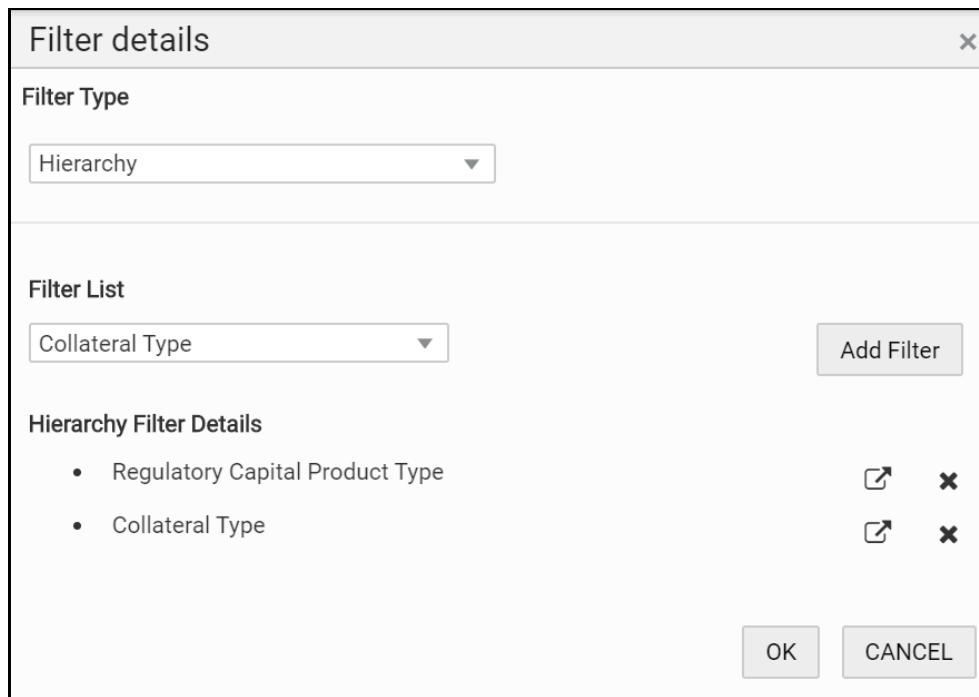
5.2.6 Applying Filter Condition on Run Pipeline

This section details about how to apply filter condition at Run Pipeline level. You can apply filter condition on Sub Pipeline level also. For more information, see [Calling another Pipeline from Your Parent Pipeline](#).

To apply filter on a Run Pipeline

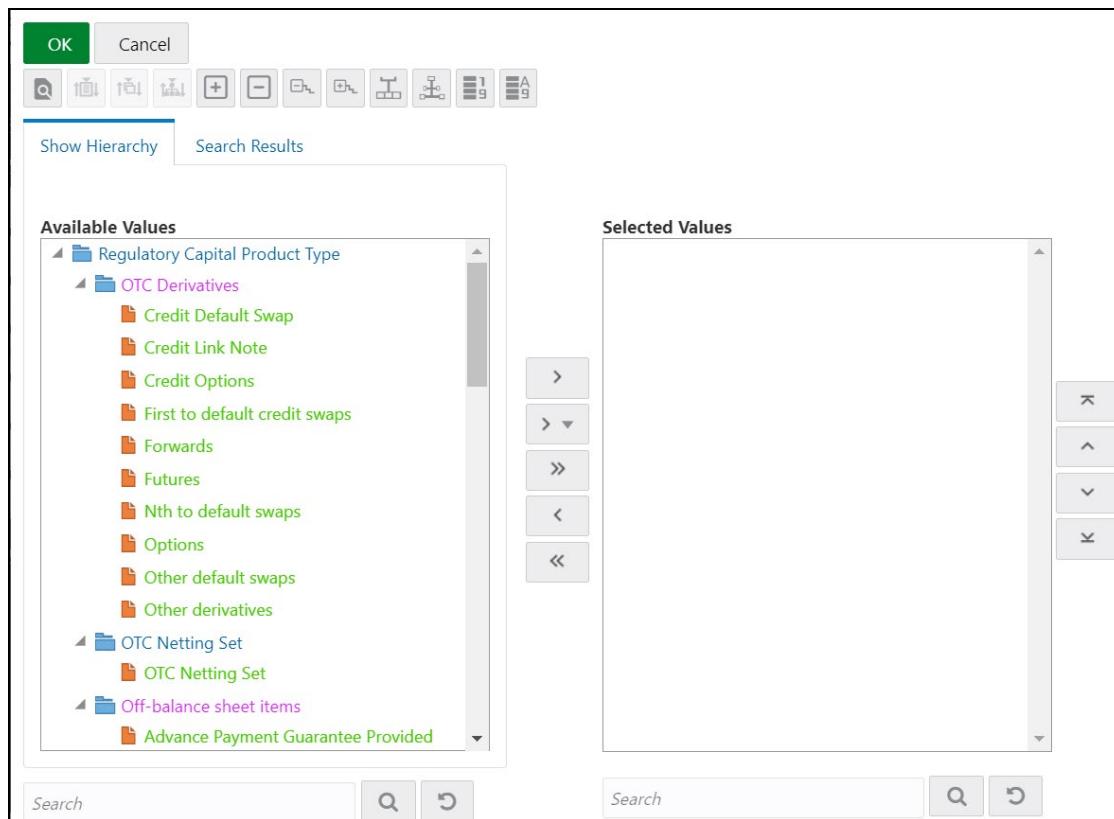
1. From the *Process Modeler Summary* window, click  corresponding to the Run Pipeline for which you want to apply filter condition and click **Filter**. The *Filter Details* window is displayed.

Figure 33: Filter Details window



2. Select the **Filter Type** as **Hierarchy** from the drop-down list. Currently only Hierarchy Filter is supported.
3. Select the Filter from the **Filter List** drop-down list. This list displays all Business Hierarchies defined in the information domain.
4. Click **Add Filter**. The Filter is displayed under *Hierarchy Filter Details* pane.
5. Click  to view the *Hierarchy Browser* window and select the hierarchy members.

Figure 34: Hierarchy Browser window



6. Select a member/node and click to select the same. Click to select the member as Self, Self Children, Parent, Siblings, and Children. For more information, see [Hierarchical Member Selection Modes](#).

In the *Hierarchy Browser* window you can also:

- Click to sort members based on path.
- Click to sort hierarchy (top to bottom).
- Click to sort based on level.
- Click or to expand or collapse the members under a node.
- Click or to expand a branch or collapse the selected branch.
- Click to focus only on the selected branch. The *Available Values* pane shows the members of the selected branch only. Click to go back to normal view.
- Click to display member's numeric codes in the right. The icon changes to .
- Click to display member's numeric codes in the left. The icon changes to .

- Click  to show only member names. This is the default view. The icon changes to .
- Click  to display member's alphanumeric codes in the right. The icon changes to .
- Click  to display member's alphanumeric codes in the left. The icon changes to .
- Click  to display only member names. This is the default view. The icon changes to .
- Select a member and click  or  to re-arrange the members in the *Selected Values* pane.
- Select a member and click  to move it to the top or click  to move it to the bottom.
- Click  to launch the Search panel. Here you can search based on **Dimension Member Numeric Code**, **Dimension Member Name** or **Dimension Member Alphanumeric Code**. You can also search in the grid based on member name using the **Search** field.

7. Click **OK**.
8. Click  to remove any selected filters.

5.2.6.1 Hierarchical Member Selection Modes

To aid the selection process, certain standard modes are offered through a drop-down. The available modes are **Self**, **Self Children**, **Parent**, **Siblings**, and **Children**.

Based on the hierarchy member security applied, the nodes/members of the hierarchy are displayed in enabled or disabled mode. The members that are in enabled mode only can be selected. That is, the members that are mapped to your user group only can be selected. For example, if you choose **Self Children**, the immediate children of the selected hierarchy that are mapped to your user group only will be moved to the RHS pane.

- The **Self** mode is the default mode displayed. In this mode, only the specific member selected in the LHS pane is selected on the RHS pane.
- Choose the **Self Children** mode when you want a specific member and only its immediate children to be selected onto the RHS pane.
- Choose the **Parent** mode when you want to select only the parent member of a selected member onto the RHS pane.
- Choose the **Siblings** mode when you want to select all the sibling members of the selected member (those members under the same parent) onto the RHS pane.
- Choose the **Children** mode when you want only the immediate children of a specific member to be selected onto the RHS pane mode.

You can also click  to select all the members to the *Selected Values* pane. Click  to deselect a selected member from the *Selected Values* pane or click  to deselect all the members.

5.2.7 Executing Run Pipeline

You can execute a Run Pipeline using the following options:

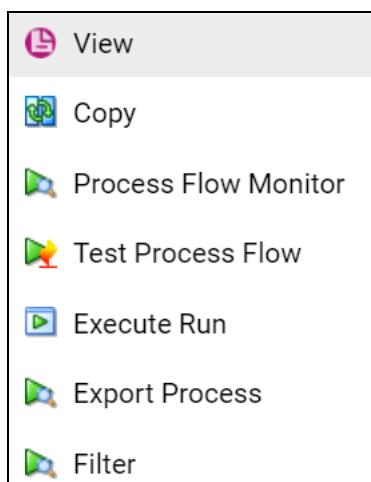
- [From UI](#)
- [Using Command Line Utility](#)
- [Using ICC Batch](#)

5.2.7.1 From UI

To execute Run Pipeline from UI:

1. From the *Process Modeller Summary* window, click  corresponding to the Run Pipeline you want to execute:

Figure 35: Sub-menu corresponding to the Run Pipeline



2. Click **Execute Run** to execute the Run Pipeline. The *Select Run Params* window is displayed.

Figure 36: Select Run Params window

The dialog box is titled "Select Run Params". It contains the following fields:

- Bucket Conversion: Mid of Bucket
- PD Interpolation Method: Non-Linear Geometric
- Interpolation Method: Cubic Spline
- FIC MIS Date: 10/17/18
- Consolidation Type: Solo
- Source Run Id: (empty)
- Legal Entity: Bank of Australia
- Reporting Currency: Afghanistan Afghani
- Run Execution Description: (empty)

At the bottom is an "OK" button.

3. Select values for the Run Parameters and click **OK**.

The execution of the Run Pipeline 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.

5.2.7.2 Using Command Line Utility

A command line utility `./wfExecExternal.sh` is available in `$FIC_DB_HOME/bin` folder.

To execute Run Pipeline using command line utility

1. Navigate to `$FIC_DB_HOME/bin` folder.
2. Execute the script file using the following command:

```
./wfExecExternal.sh processInstanceId processId '$objectId' objectType
infodom userID segment locale 'applicationparams' 'securityparams'
```

- `processInstanceId` - Instance Id of the Process or Run Pipeline
- `processId` - Process ID of the Run Pipeline. This is a mandatory parameter.
- `objectId` - This is an auto-generated unique Object ID. Enter '`$objectId`' as mandatory parameter value.
- `objectType` - Specify the Object Type if it is defined in the `aaI_wf_app_definition_map` table.
- `infodom` - Information Domain Name

- `userID` - Specify the user ID
- `segment` - Segment Name
- `locale` - Locale selected. For example, `en_US`
- `'applicationparams'` - Specify values for the Run execution parameters stored in `APP_COMP_ATTR_MAP_ID` within single quotes separated by comma.
- `'securityparams'` - Specify any security parameters within single quotes separated by comma

For example:

```
./wfExecExternal.sh null '$objectID' CECLShellLatest null OFSIFRSINFO
OFSIFRSINFO AAAIUSER en_US
" hierDetailsLE": [{"hierValueCode": "AU", "value": "Bank of
Australia", "hierNodeCode": "[HLLFP029].[0].[AU]", "leafCondition": "CASE
WHEN DIM_ORG_STRUCTURE.f_latest_record_indicator = 'Y' THEN
DIM_ORG_STRUCTURE.v_entity_code END='AU'", "hierNodeDesc": "Bank of
Australia"}], "CONSOTYPE": "S", "hierDetailsRCY": [{"hierValueCode": "DZD", "value": "Algerian
Dinar", "hierNodeCode": "[HLLFP026].[DZD]", "leafCondition": "DIM_CURRENCY.
v_iso_currency_cd = 'DZD'", "hierNodeDesc": "Algerian
Dinar"}], "FIC_MIS_DATE": "2019-04-19", "V_RUN_MAIN_DESC": "Test"}' null
```

TIP Null needs to be passed if you do not want to pass value for a parameter.

5.2.7.3 Using ICC Batch

To execute a Run Pipeline from the **Operations** menu as an ICC batch, the

`./pipelineExecutor.sh` script file is made available in the `$FIC_DB_HOME/bin` directory. You must enter the specified parameters in the `./pipelineExecutor.sh` file required for **Run Pipeline** execution and then create an ICC Batch to call this script file through the Run Executable component.

To update the parameters in the `./pipelineExecutor.sh` script file for Run Pipeline execution, follow these steps:

1. Navigate to the `$FIC_DB_HOME/bin` directory.
2. Update the script file by modifying the values that are enclosed within the `##` symbols. The following is an example for the format:

```
pmfscriptexe=".wfExecExternal.sh null '##PROCESS_NAME##' $objectID
null '##INFODOM##' '##SEGMENT##' '##USER##' en_US'{##EXECUTION PARAM
JSON##, \"FIC_MIS_DATE\": \"$datevar\"}' null"
```

The following list provides descriptions of the script file parameters:

- `Process_Name` - Enter the process name of the Run Pipeline. This is a required parameter.
- `$objectID` - This is an auto-generated unique Object ID that you must not modify. Retain `$objectID` in the script as this mandatory parameter is created by the PMF application during the execution of the Run Pipeline.

- **INFODOM** - Enter the Information Domain name here.
- **SEGMENT** - Enter the Segment name here.
- **USER** - Enter the user ID here.
- **locale** - Modify the locale as required. For example, for English-United States, enter **en_US**.
- **EXECUTION PARAM JSON** - Enter the Run Execution parameters in JSON format.

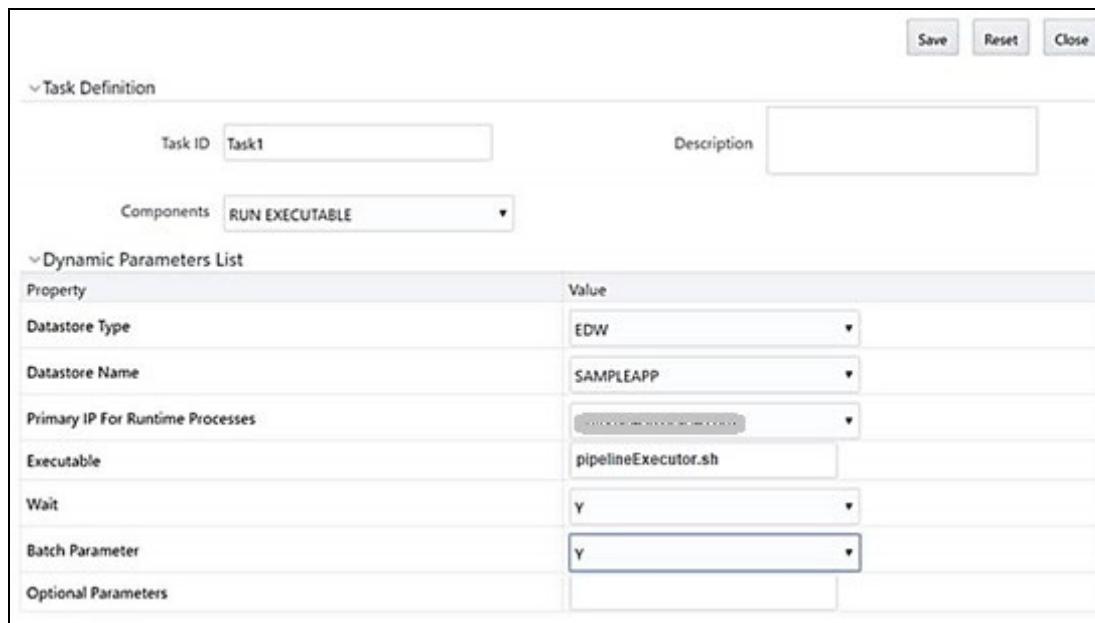
The following code snippet is an example:

```
{\"hierDetailsLE\":[{\\"hierValueCode\":\"ORG_NAME\", \"value\":\"ORG_NAME\", \"hierNodeCode\":\"ORG_NAME\", \"leafCondition\\":\"CASE WHEN DIM_ORG_STRUCTURE.f_latest_record_indicator = 'Y' THEN DIM_ORG_STRUCTURE.v_entity_code END='ORG_NAME'\", \"hierNodeDesc\":\"ORG_NAME\"}], \"CONSOTYPE\\":\"SOL0\", \"hierDetailsRCY\":[{\\"hierValueCode\":\"INR\", \"value\":\"India n Rupee\", \"hierNodeCode\":\"INR\", \"leafCondition\\":\"DIM_CURRENCY.v_iso_currency_cd = 'INR'\", \"hierNodeDesc\\":\"Indian Rupee\"}], hierDetailsUDRS: [{\"hierValueCode\": \"BISNONSECSTD-SETUP1\", \"value\": \"BIS Non Sec STD - Setup 1\", \"hierNodeCode\": \"BISNONSECSTD-SETUP1\", \"leafCondition\\": \"USR_DEFINED_RUN_PARAMETERS.V_RUN_PARAMETER_SETUP_CODE = 'BISNONSECSTD-SETUP1'\", hierNodeDesc: \"BIS Non Sec STD - Setup 1\"}], \"FIC_MIS_DATE\":\"$datevar\", \"V_RUN_MAIN_DESC\\":\"RUN MARCH 09\"}
```

- **\$datevar** - Retain **\$datevar** variable in the execution param JSON specified in the preceding list without any modification as this gets replaced by the **FIC_MIS_DATE** selected during batch execution as the date value for the Run execution.

3. Create a new batch in the **Batch Maintenance** window in the **Operations** menu. For more information, see the *Adding Batch Definition* section in the [OFS Analytical Applications Infrastructure User Guide](#).
4. Create a new task with the **Run Executable** task component. For more information, see the *Adding Task Details* section in the [OFS Analytical Applications Infrastructure User Guide](#).

Figure 37: Call pipelineExecutor.sh in the Executable Field



The screenshot shows the 'Task Definition' configuration window. At the top right are 'Save', 'Reset', and 'Close' buttons. The 'Task ID' is set to 'Task1' and the 'Components' dropdown is set to 'RUN EXECUTABLE'. The 'Dynamic Parameters List' section contains the following entries:

Property	Value
Datastore Type	EDW
Datastore Name	SAMPLEAPP
Primary IP For Runtime Processes
Executable	pipelineExecutor.sh
Wait	Y
Batch Parameter	Y
Optional Parameters	

- Call the `pipelineExecutor.sh` script file in the **Executable** field.
- Select **Y** in the **Batch Parameter** drop-down list for all cases.
- Select **Y** in the **Wait** drop-down list if you want the Run executable to wait to finish the task execution and then update the task status.

5. Click **Save** to execute the Batch.
6. Schedule a batch through the **Batch Scheduler** window in the **Operations** menu or schedule and execute the batch through the External Scheduler (ESIC).
For more information, see the *Batch Scheduler* and *External Scheduler Interface Component* sections in the [OFS Analytical Applications Infrastructure User Guide](#).

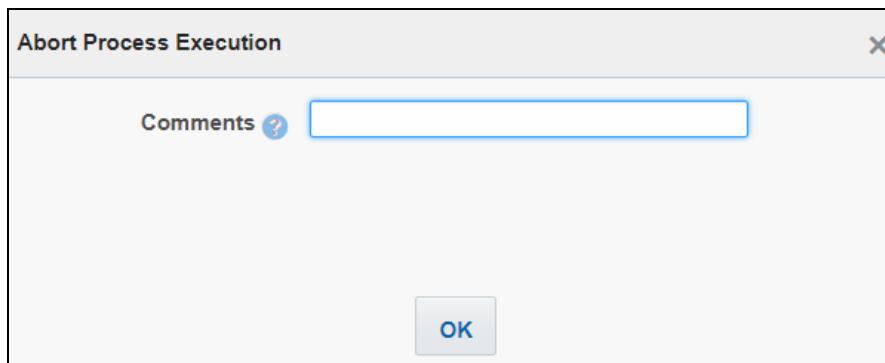
5.2.8 Abort Run Pipeline

The Abort feature facilitates you to abort a Run Pipeline that is in the process of execution.

To abort a Run Pipeline

1. From the *Process Monitor* window, click  corresponding to the Run Pipeline you want to abort and click **Abort**.

Figure 38: Abort Process Execution window



2. Enter comments if any, for aborting the Run Pipeline execution.

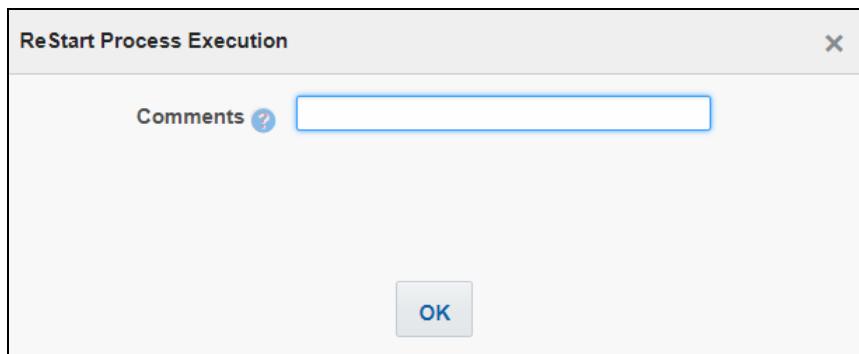
5.2.9 Resume Run Pipeline

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

1. From the *Process Monitor* window, click  corresponding to the Run Pipeline you want to resume and click **Resume**.

Figure 39: Re-Start Process Execution window



2. Enter comments if any, for restarting the Run Pipeline execution.

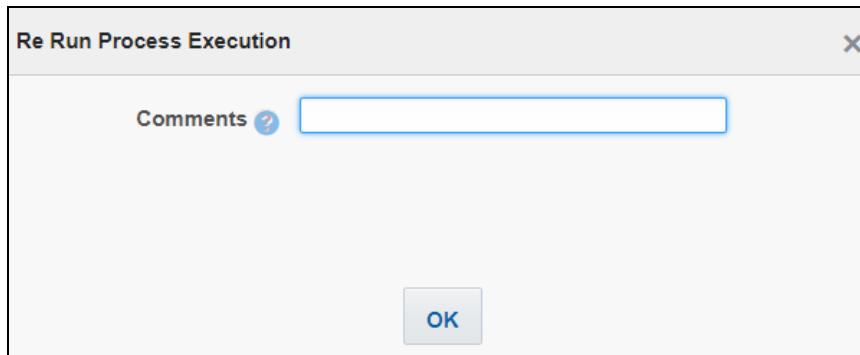
5.2.10 Re-run Run Pipeline

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

To re-run a Run Pipeline

1. From the *Process Monitor* window, click  corresponding to the Run Pipeline you want to re-run and click **Re-Run**.

Figure 40: Re-Run Process Execution window



2. Enter comments if any, for re-running the Run Pipeline execution.

5.3 Additional Functionalities

This section describes the additional functionalities that can be performed in the *Process Modeler* window.

Topics:

- [Modifying a Pipeline](#)
- [Testing a Pipeline](#)
- [Viewing a Pipeline](#)
- [Copying a Pipeline](#)
- [Deleting a Pipeline](#)

5.3.1 Modifying a Pipeline

This option allows you to modify a Pipeline.

To modify a Sub Pipeline perform the following:

1. From the *Process Modeler* window, search for the Pipeline (Business or Run Pipeline) and click the Pipeline name. The *Process Flow* tab is displayed.
2. Modify the Process Flow, Definition, Application Rules, and Data Fields, as required.

NOTE

When you try to delete a component in the Business Pipeline that is used as a Sub Pipeline, a confirmation message is displayed. After you click **OK** to confirm deletion, an alert is displayed with the list of the Business Pipelines where the Sub Pipeline is used. You should click **OK** again to delete the component.

5.3.2 Testing a Pipeline

This option allows you to check whether the Process flow you designed works as expected.

To test a Process flow

1. From the *Process Modeler* window, click  to view the sub-menu and select **Test Process Flow**. The Execute Workflow window is displayed.

Figure 41: Execute Workflow window



2. Enter an **Object ID** to identify the Process flow. This is displayed as Entity ID in the *Process Monitor* window.
3. Enter the Application Parameters that are required in the Process flow you are testing. Specify parameters in JSON format. This is an optional field.
For example, `{ "WF_RUNSK": "15", "WF_MISDATE": "12/31/1999" }`
4. Click **OK**.

5.3.3 Viewing a Pipeline

This option allows you to view the workflow of an already existing Business Process.

From the *Process Modeler* window, click  to view the sub-menu and select **View**. You can view the Process Flow of the Pipeline.

5.3.4 Copying a Pipeline

This option facilitates you 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

1. From the *Process Modeler* window, click  to view the sub-menu and select **Copy**. The *Process Details* window is displayed.
2. 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. Select the newly created Process and click the Process Name. Modify the Process flow and other details if required.

5.3.5 Deleting a Pipeline

This option allows you to delete a Business Pipeline or Run Pipeline.

To delete a Pipeline perform the following:

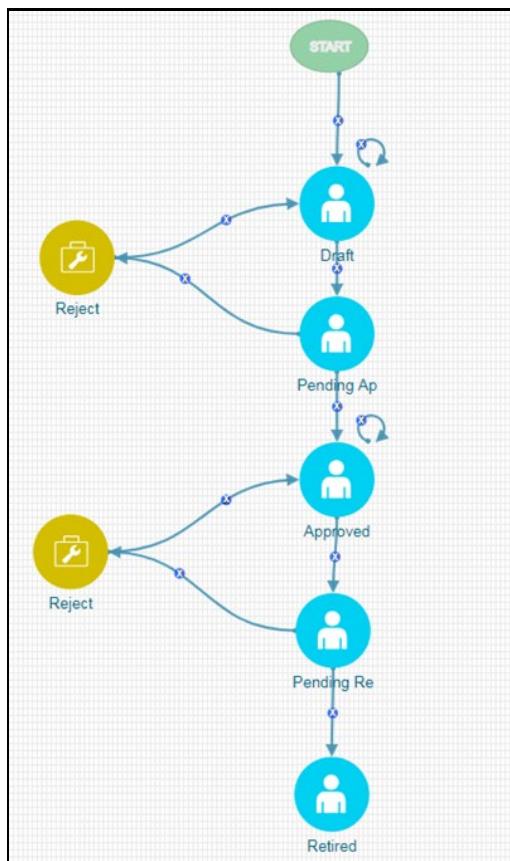
1. From the *Process Modeler* window, click  corresponding to the Pipeline you want to delete.
2. A confirmation message is displayed.
3. Click **OK** to confirm deletion. If this Pipeline is used as a Sub Pipeline in other Business Pipelines, an alert is displayed with the list of Business Pipelines where this is used.
4. Click **Delete Anyway** to delete the Pipeline or click **Cancel** to cancel the delete operation.

6

Human Tasks

Human Task is used if an activity requires a human intervention to go to the next activity.

Figure 42: Human Tasks Flow



6.1

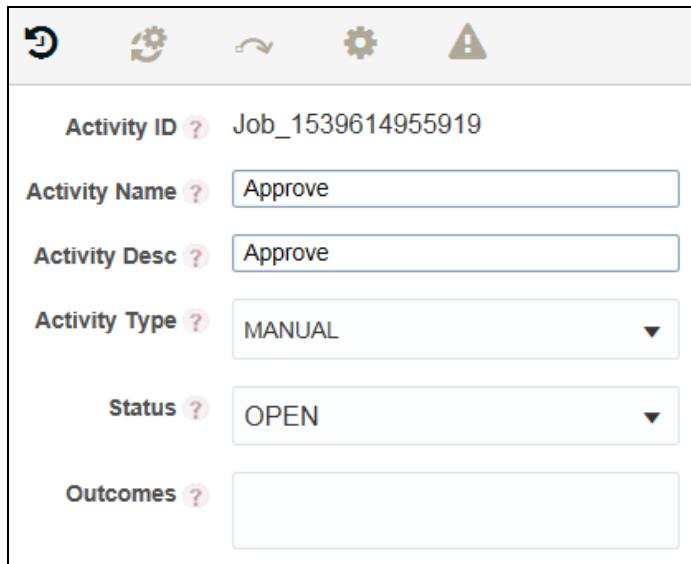
How to Use Human Task

1. From the *Process Flow* tab, click  Human Task from the *Activities* pane in the floating toolbar and click on the drawing canvas where you want to place it.
2. Double-click  icon. The *Activity* tab is displayed.

6.1.1

Activity Tab

Figure 43: Activity Tab



The screenshot shows a configuration form for an activity. At the top are five icons: a circular arrow, a gear, a double arrow, another gear, and a warning sign. Below these are five input fields with labels and dropdowns:

- Activity ID**: Job_1539614955919
- Activity Name**: Approve
- Activity Desc**: Approve
- Activity Type**: MANUAL
- Status**: OPEN

Below these fields is a large empty text area labeled **Outcomes**.

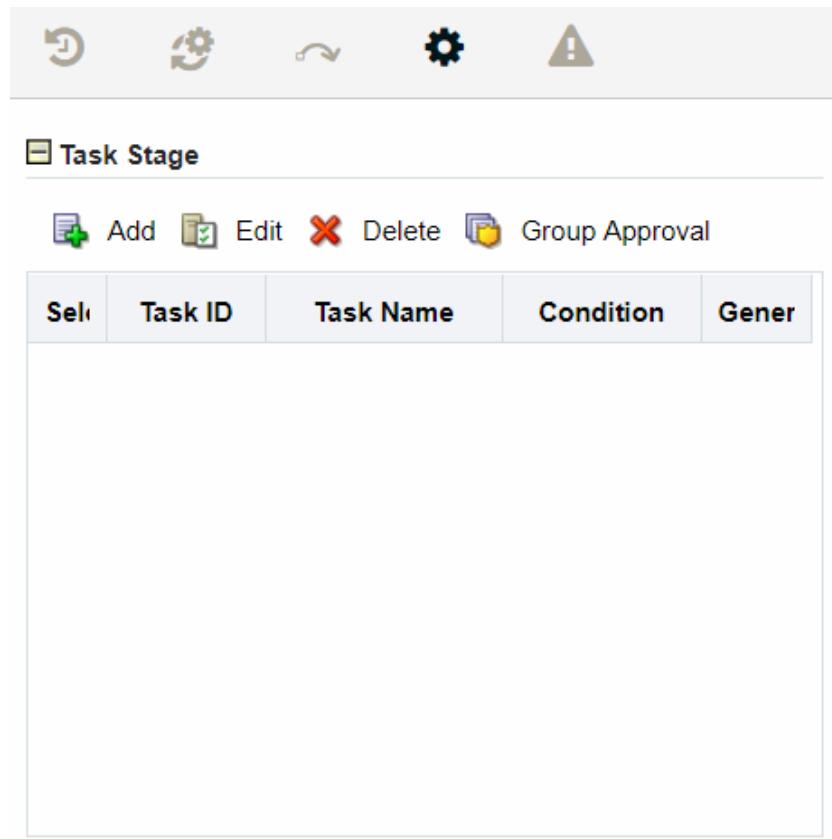
1. **Activity ID** is auto populated.
2. Enter Activity Name and Description.
3. Select the **Status** of the activity from the drop-down list. The list displays the seeded values in the AAI_WF_STATUS_B table.

6.1.2 Action Tab for Creating Tasks/Notification

Action/Task is used to inform the assigned user about an action to be completed in the current stage of Workflow. You can add multiple tasks for an activity. A task can be assigned to a user, user group, users with a particular role, or users selected by an application rule. The task is sent to the Application Inbox of the assigned users.

1. Click  to define Actions.

Figure 44: Action Tab for Creating Tasks/Notification



2. Click **Add** from the Action tab to add a new Task.

6.1.2.1 Defining Task Details

Figure 45: Task Group Details window - Defining Task Details

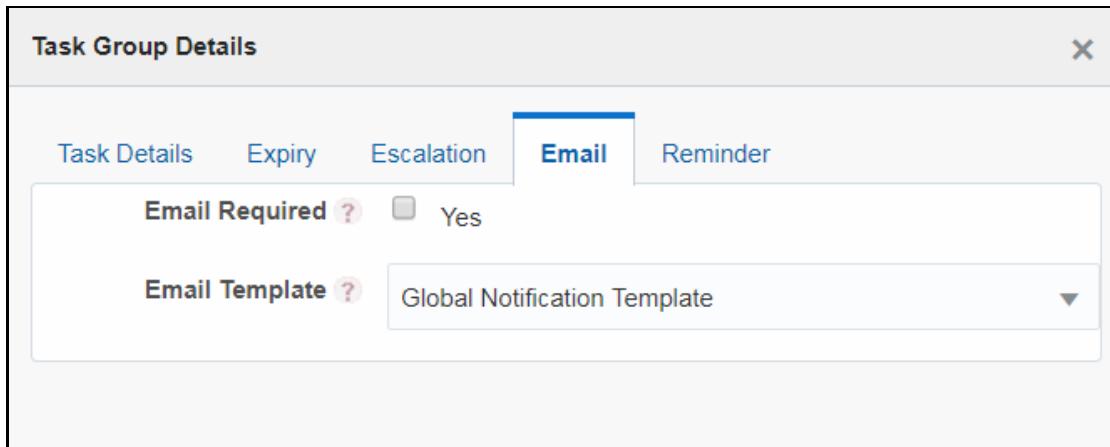
A screenshot of the "Task Group Details" window. The "Task Details" tab is selected. The window contains the following fields:

- Task ID:** 1513333455171
- Task Name:** TaskName_
- Condition:** Default
- Generate:** On Entry
- Enabled:** Yes (checkbox checked)

1. Task ID is automatically generated.
2. Enter a Task Name. By default, the task name is prefixed with “TaskName_”.
3. Select the decision rule so that when the **Condition** is satisfied, the Task is sent to the users associated with this task.
4. Select the required option for **Generate**:
 - On Entry- Task is sent before executing the activity business logic.
 - On Exit- Task is sent before leaving the activity and moving to the next activity.
 - On Stage- When there is a transition where the target and source are the same activity, and if that transition happens, then OnStage tasks are triggered.
5. Select the **Yes** checkbox to enable the Task.

6.1.2.2 Setting Email Notification

Figure 46: Task Group Details window - Setting Email Notification



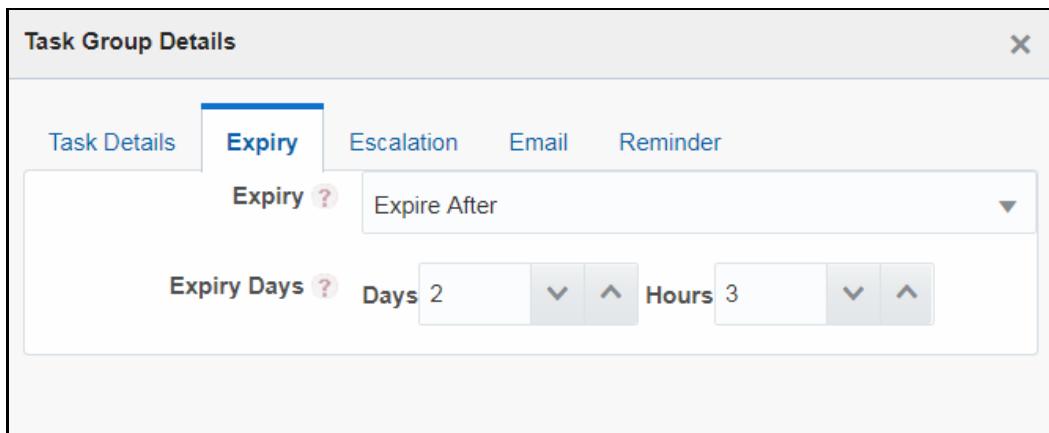
You should have configured Email settings before you set an email notification for your tasks. For more details on how to configure email settings, see [Configuring Email for Human Tasks](#) section.

1. Select the **Yes** checkbox if an email needs to be sent for the task.
2. Select the email template to be used from the drop-down list. Templates are populated from AAI_WF_Email_Template table.

6.2 Additional Functionalities for Human Tasks

6.2.1 Setting Task Expiry

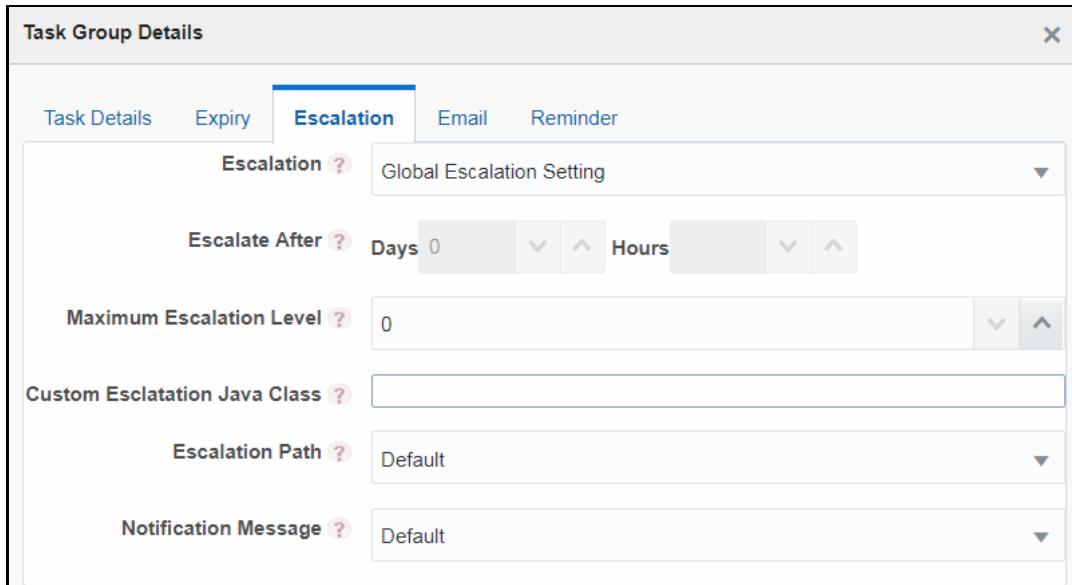
Figure 46: Task Group Details window - Setting Task Expiry



1. Select the required option from the **Expiry** drop-down list:
 - Global Expiry Setting- Select this option to set the task expiry based on a global setting. Global setting can be set from the backend.
 - Never Expire- Select this option if the task should not expire.
 - Expire After- Select this option if you want to set the task expiry after some days and/or hours.
 - Dynamic Value – Select this option if you want the user assigned to the task to set the task expiry date and time dynamically. This value needs to be entered in the code.
2. If **Expiry** is selected as **Expire After**, enter the number of days and/or hours after which the task should be expired.

6.2.2 Setting Task Escalation

Figure 47: Task Group Details window - Setting Task Escalation



1. Select the **Escalation** criteria:

- Global Escalation Setting- Select this option to set the task escalation based on a global setting. Global setting can be set from the backend.
- Never Escalate - Select this option if escalation is not required for the task.
- Escalate After- Select this option if you want to escalate if the task is not addressed after some days and/or hours.

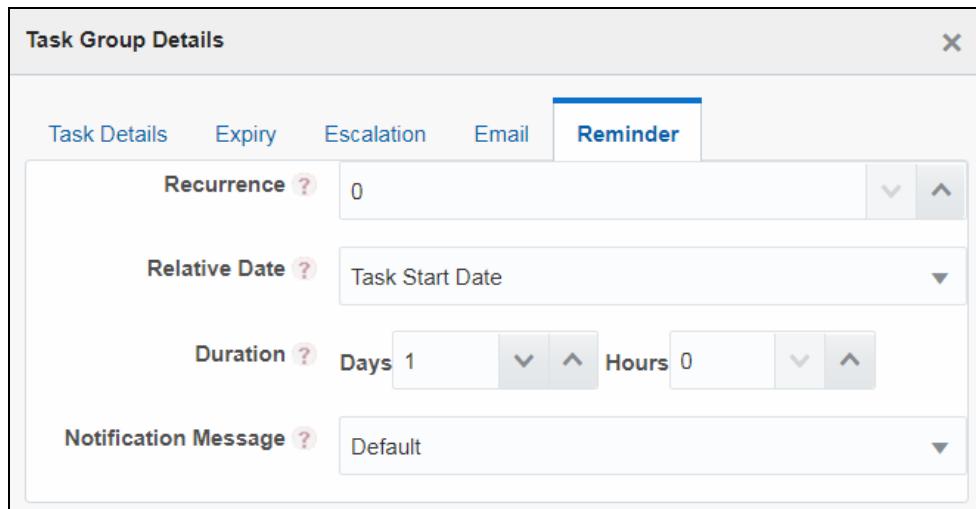
NOTE

Escalation should be set after the expiry of the task. If you have selected Never Expire option for Expiry, you cannot set escalation.

2. If **Escalate After** is selected for **Escalation**, enter the number of days and/or hours after which the escalation should be triggered.
3. Enter the maximum number of escalation level. 1 indicates escalation to the immediate manager, 2 indicates escalation to the manager's manager and so on.
4. Enter the custom escalation Java Class that you want to call.
5. Select the escalation path from the drop-down list. The options are Default, People Hierarchy, and Custom Rule.
6. Select the type of notification message from the drop-down list. Notification messages are populated from the `AAI_WF_Templates` table.

6.2.3 Setting Reminder for Your Task

Figure 48: Task Group Details window - Setting Reminder for Your Task



Reminders are sent to the assigned user as an open task in their inbox.

1. Enter the number of times you want to set the reminder from the **Recurrence** field.
2. Select **Task Start Date** if you want to send a reminder after the defined number of days and/or hours, from the start date of the task. Select **Task Expiration Date** if you want to send a

reminder before the defined number of days and/or hours from the end date/expiry date of the task.

3. Select the number of days and/or hours from the **Relative Date** after/before which you want to set the reminder.
4. Select the **Notification Message** you want to send, from the drop-down list. Notification messages are populated from the `AAI_WF_Templates` table.
5. Click **OK**.

7

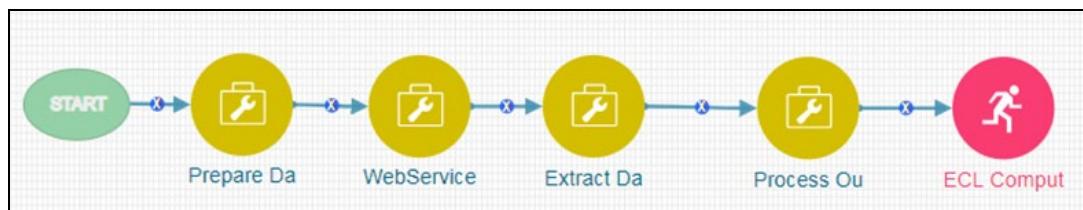
Service Tasks

Service Task is an automatic task that gets triggered in the Process flow. It is used to execute the Business Logic that is defined through an Application Rule of Execution Rule type. For more information on configuring the Application Rules, see the [Application Rule](#) section.

Service Tasks are used to invoke External Model Service through Rest API, External Java APIs, Stored Procedures, and Functions.

A more detailed explanation of invoking external model service is available in [How to invoke External Model through Web Service](#) section.

Figure 49: Service Tasks flow



7.1

How to Use a Service Task



1. From the *Process Flow* tab, click **Service Task** from the *Activities* pane in the floating toolbar and click on the drawing canvas where you want to place it.
2. Double-click  icon. The *Activity* tab is displayed.

7.1.1

Activity Tab

Figure 50: Activity Tab

Activity ID	Job_1522122432701
Activity Name	WebService
Activity Desc	Activity for calling external model
Activity Type	AUTOMATIC
Status	(dropdown menu)
Outcomes	(dropdown menu)

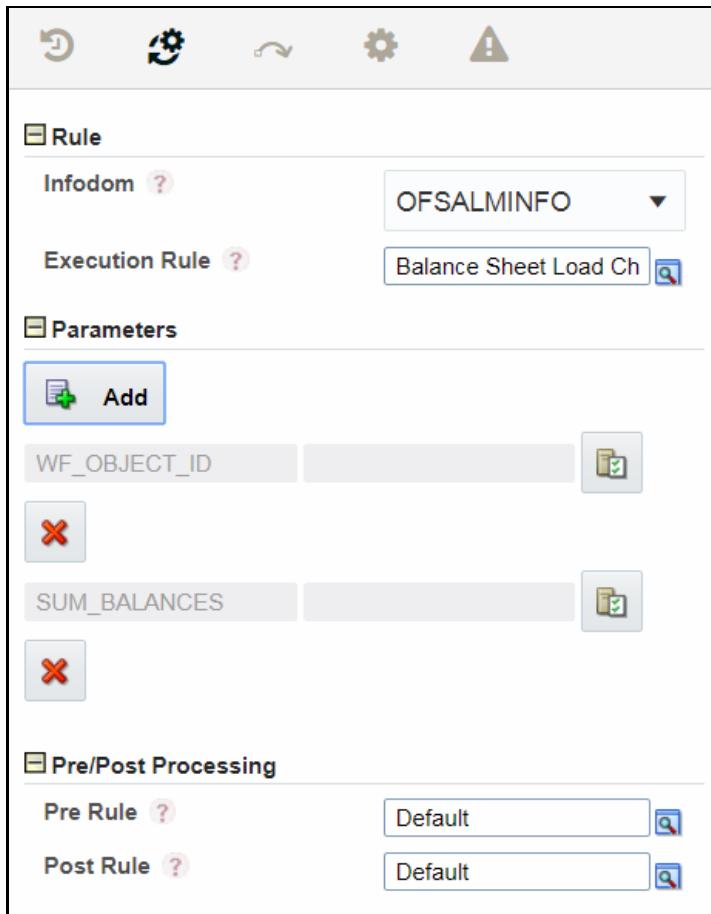
3. **Activity ID** is auto populated.

4. Enter Activity Name and Activity Description.

7.1.2 Implementation Tab

1. Click . The *Implementation* tab is displayed.

Figure 51: Implementation tab

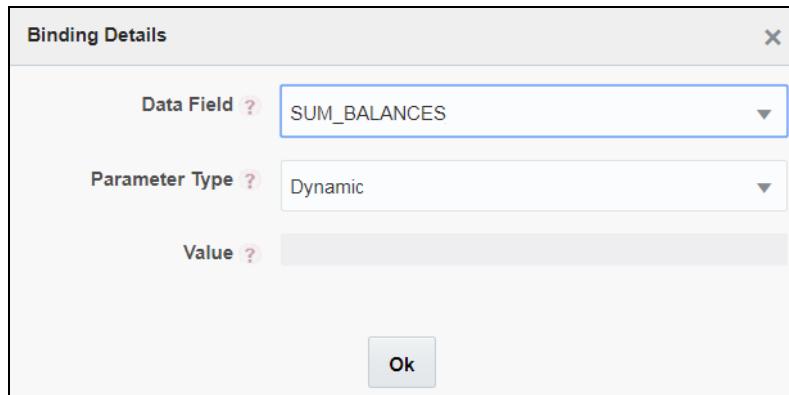


The screenshot shows the 'Implementation' tab of a service task configuration. At the top, there are several icons: a gear, a magnifying glass, a double arrow, a gear, and a warning sign. Below these are two sections: 'Rule' and 'Parameters'. In the 'Rule' section, 'Infodom' is set to 'OFSALMINFO' and 'Execution Rule' is set to 'Balance Sheet Load Ch'. In the 'Parameters' section, there are two data fields: 'WF_OBJECT_ID' and 'SUM_BALANCES', each with a 'Remove' button (a red 'X') and a 'Details' button (a magnifying glass). Below these are sections for 'Pre/Post Processing' with 'Pre Rule' and 'Post Rule' both set to 'Default'.

2. Select the information domain where the Execution Rule you want to execute is available, from the **Infodom** drop-down list. The list displays all the Infodoms mapped to the applications configured in your OFSAA instance.
3. Select the **Execution Rule** that needs to be executed for this activity. For more information on how to define an Application Rule, see the [Application Rules](#) section.
 - Click . The *Participant Details* window is displayed with all Application Rules of Execution Rule types available in your Process.
 - Click the Name link of the Application Rule to view the details.
 - Select the required Rule and click **Ok**.
4. Add Parameters you want to pass to the Execution Rule using Data Fields. You can pass Static values or Dynamic Values. In case of Dynamic, the value needs to be entered during the execution of the workflow.

- Click  **Add** under **Parameters**. The *Binding Details* window is displayed.

Figure 52: Binding Details window



- Select the **Data Field** to which you want to pass the value. The list displays all Data Fields for the current Process or Package.
- Select the **Parameter Type** as Static to pass a static value to the selected Data Field in the **Value** field or as Dynamic to pass the value during execution of the workflow.

NOTE

The added parameters are displayed under Parameters pane. You can click  corresponding to a parameter to edit it or click  to delete it.

5. Select an application rule that you want to execute before executing the Execution Rule.
 - Click  corresponding to **Pre Rule** and select the required application rule.
6. Select an application rule that you want to execute after executing the Execution Rule.
 - Click  corresponding to **Post Rule** and select the required application rule.

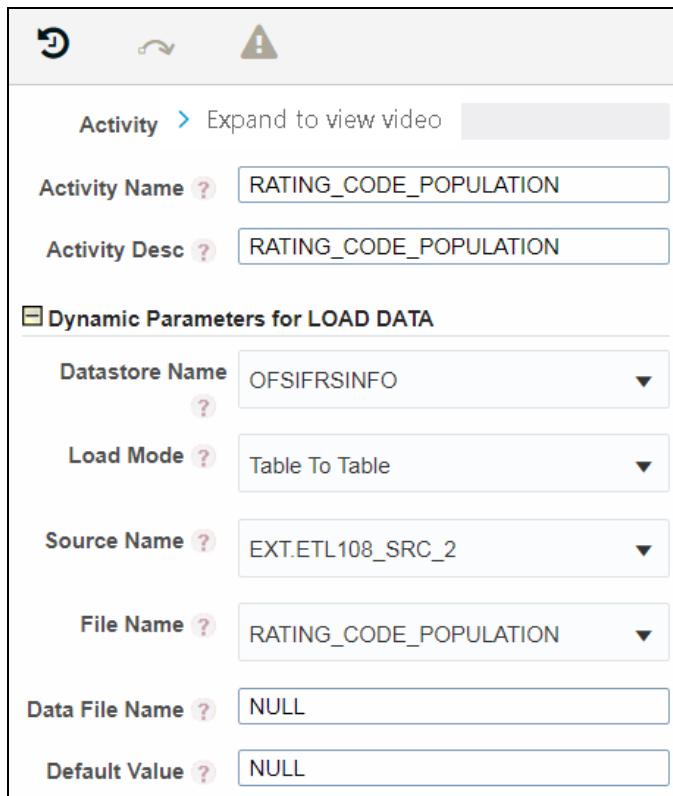
8 Configuring OFSAA Tasks in Your Process Flow

OFSAAs widgets are used to orchestrate OFSAA components like T2T definitions, PLC definitions (DT), Rules (Classification Rule and Computation Rule), Models in EMF, Run, Run Executable, and Data Quality Groups into your Process Flow using Process Modeller.

8.1 How to Use an OFSAA Widget

1. From the *Process Flow* tab, click the required widget under the **OFSAAs Widget** toolbar in the left pane and click the canvas. The available OFSAA Widgets are LoadT2T, TransformDT, RuleType2, RuleType3, MFModel, Run, RunExecutable, and DataQualityGroups.
2. Double-click the widget. On the Right Pane, the *Activity* tab is displayed.

Figure 53: OFSAA Widget Activity tab

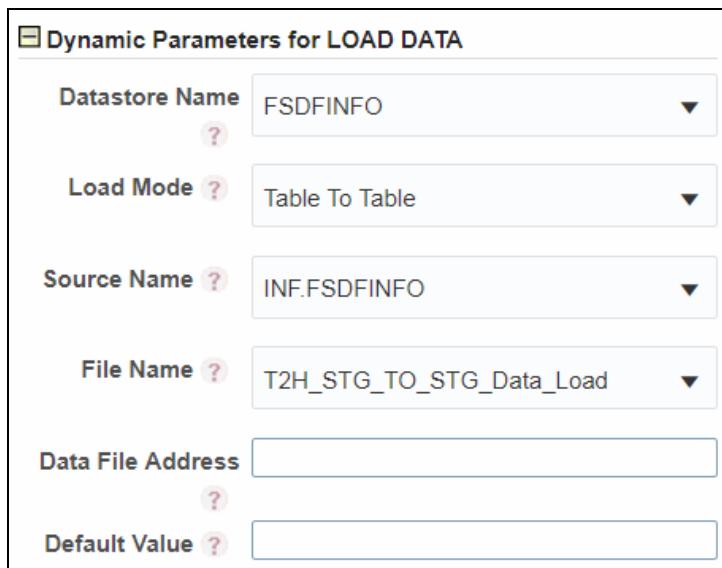


3. Enter the **Activity Name** and **Activity Description**.
4. Based on the OFSAA widget, the **Dynamic Parameters** are displayed. For more information, see [Dynamic Parameters for OFSAA Widgets](#) section.
5. To view the definition of the OFSAA component you are using, mouse over its icon and click **View**. The *Definition* window of the OFSAA component is displayed.

8.2 Dynamic Parameters for OFSAA Widgets

8.2.1 LoadT2T

Figure 54: Dynamic Parameters for LOAD DATA



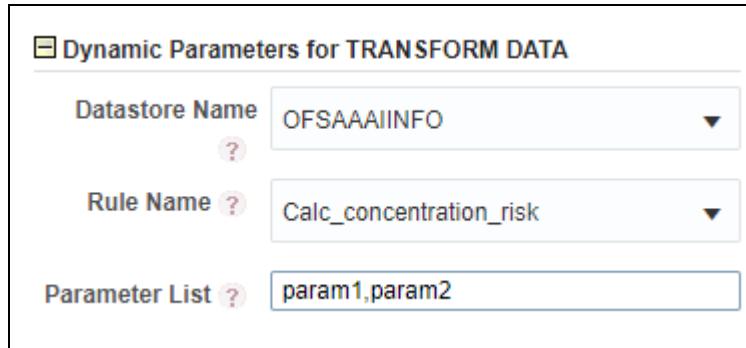
The screenshot shows a configuration dialog for 'Dynamic Parameters for LOAD DATA'. The fields are as follows:

- Datastore Name:** FSDFINFO
- Load Mode:** Table To Table
- Source Name:** INF.FSDFINFO
- File Name:** T2H_STG_TO_STG_Data_Load
- Data File Address:** (empty)
- Default Value:** (empty)

Table 18: Dynamic Parameters for LOAD DATA Description

Field Name	Description
Datastore Name	Select the Information Domain in which the Data Mapping you want to execute is present, from the drop-down list.
Load Mode	Table to Table needs to be selected for Data Mapping definitions such as T2T, T2H, H2T, H2H, and L2H definitions.
Source Name	Select the required source on which the Data Mapping definition you want to execute is defined, from the drop-down list.
File Name	Select the Data Mapping definition you want to execute, from the drop-down list. Based on the selected Load Mode and Source Name, the list displays the corresponding definitions.
Data File Address	This field is not applicable to Load Mode selected as Table-to-Table.
Default Value	Select the Data Fields you want to pass as parameters for the selected Data Mapping definition. For information on creating Data Fields, see the Data Fields section. For additional information, see Task Component Parameters section in the OFS Analytical Applications Infrastructure User Guide .

8.2.2

TransformDT **Figure 54: Dynamic Parameters for TRANSFORM DATA**


Dynamic Parameters for TRANSFORM DATA

Datastore Name: OFSAAINFO

Rule Name: Calc_concentration_risk

Parameter List: param1,param2

Table 19: Dynamic Parameters for TRANSFORM DATA Description

Field Name	Description
Datastore Name	Select the datastore name in which the PLC you want to execute is present from the drop-down list.
Rule Name	Select the Post Load Changes (DT) definition you want to execute from the drop-down list. The list displays the Post Load Changes definitions in the selected Information Domain.
Parameter List	Enter the Data Fields you want to pass as parameters for the selected Data Mapping definition. Use comma-separated values if you want to enter more than one Data Field. For information on creating Data Fields, see Data Fields section.

8.2.3

RuleType2  **or RuleType3** **Figure 55: Dynamic Parameters for RULE_EXECUTION**

Dynamic Parameters for RULE_EXECUTION

Datastore Name	OFISIFRSINFO
Rule Code	External Rating to Internal Rating Re-cl.
Build Flag	No
Optional Parameters	

Table 20: Dynamic Parameters for RULE_EXECUTION Description

Field Name	Description
Datastore Name	Select the Information Domain in which the RRF Rule you want to execute is present, from the drop-down list.
Rule Code	Display the codes of the RRF Rules defined under the selected Infodom. Select the required Rule from the drop-down list.
Build Flag	Select the required option from the drop-down list as "Yes" or "No". Build Flag refers to the pre-compiled rules, which are executed with the query stored in the database. While defining a Rule, you can make use of Build Flag to fasten the Rule execution process by making use of existing technical metadata details wherein the rule query is not rebuilt again during Rule execution. Built Flag status set to "No" indicates that the query statement is formed dynamically retrieving the technical metadata details. If the Build Flag status is set to "Yes", then the relevant metadata details required to form the rule query are stored in the database on "Save" of a Rule definition. When this rule is executed, the database is accessed to form the rule query based on stored metadata details, thus ensuring performance enhancement during Rule execution. For more information, see Significance of Pre-Built Flag section in OFS Analytical Applications Infrastructure User Guide .
Optional Parameters	Select the Data Fields you want to pass as parameters for the selected Data Mapping definition. For information on creating Data Fields, see the Data Fields section.

8.2.4**MFModel**

Figure 56: Dynamic Parameters for MODEL

The screenshot shows a configuration dialog box titled "Dynamic Parameters for MODEL". It contains four fields: "Datastore Name" with a dropdown menu showing "FSDFINFO", "Model Code" with a dropdown menu showing "PD Model_0", "Operation" with a dropdown menu showing "ALL", and "Optional Parameters" which is an empty text input field.

Table 21: Dynamic Parameters for MODEL Description

Field Name	Description
Datastore Name	Select the Information Domain in which the RRF Rule you want to execute is present, from the drop-down list.
Model Code	Display the codes of the EMF Models defined under the selected Infodom.
Operation	The All definition for the Operation field conveys the process of extracting the data from the flat files and applying the run regression on the data extracted. For Batches that are being built for the first time the data is extracted from the flat files and the run regression is applied on it.
Optional Parameters	Select the Data Fields you want to pass as parameters for the selected Data Mapping definition. For information on creating Data Fields, see the Data Fields section.

8.2.5 Run

Figure 57: Dynamic Parameters for RUN

Dynamic Parameters for RUN

Datastore Name: HIVEDOM1

Run Type: Base Run

Execution Rule

Run Parameters: datafield1, datafield2

Table 22: Dynamic Parameters for RUN Description

Field Name	Description
Datastore Name	Select the required datastore from the drop-down list.
Run Type	Select Base Run or Simulation Run based on the type of the Run you want to execute, from the drop-down list.
Execution Rule	Select the Run you want to execute from the drop-down list.
Run Parameters	<p>Enter the Data Fields you want to pass as parameters for the selected Data Mapping definition. Use comma-separated values if you want to enter more than one Data Field.</p> <p>For information on creating Data Fields, see Data Fields section.</p>

8.2.6 DataQualityGroups

Figure 58: Dynamic Parameters for RUN DQ RULE

Dynamic Parameters for RUN DQ RULE

Datastore Name	OFSLALMINFO
User Id	AAAIUSER
DQ Group Name	LRMDQGRP22
Rejection Threshold	
Additional Parameters	
Fail if Threshold Breaches	No
Optional Parameters	-1

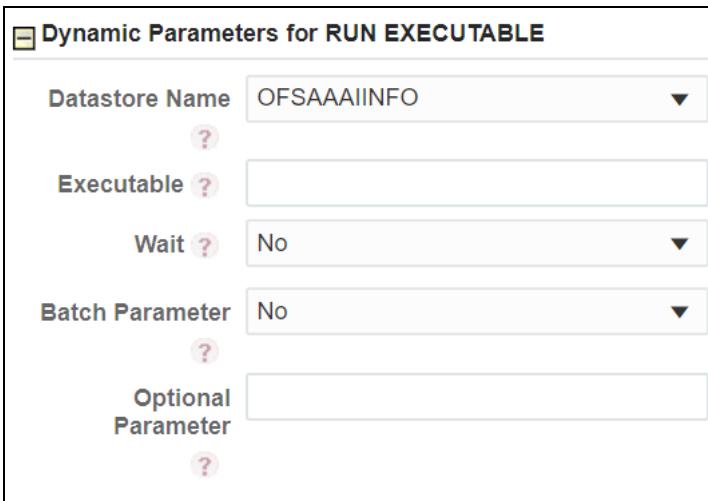
Table 23: Dynamic Parameters for RUN DQ RULE Description

Property	Description
DQ Group Name	Refers to the Data Quality Groups consisting of associated Data Quality Rule definition(s). Select the required DQ Group from the drop-down list.
Rejection Threshold	Specify the percentage of the Rejection Threshold (%) limit in numeric value. This refers to the maximum percentage of records that can be rejected in a job. If the percentage of failed records exceeds the Rejection Threshold, the job fails. If the field is left blank, the default the value is set to 100%.
Additional Parameters	Specify the Additional Parameters as filtering criteria for execution in the pattern Key#Data type#Value; Key#Data type#Value;...etc. Here the Data type of the value should be "V" for Varchar/Char, or "D" for Date with "MM/DD/YYYY" format, or "N" for numeric data. For example, if you want to filter some specific region codes, you can specify the Additional Parameters value as \$REGION_CODE#V#US;\$CREATION_DATE#D#07/06/1983;\$ACCOUNT _BAL#N#10000.50; Note: In case the Additional Parameters are not specified, the default value is fetched from the corresponding table in configuration schema for execution.

Property	Description
Fail if Threshold Breaches	Select Yes or No from the drop-down list. If Yes is selected, execution of the task fails if the threshold value is breached. If No is selected, the execution of the task continues. Note: For Custom Check type DQ Rules in Hive Infodoms, the execution of the task does not fail even if the threshold is breached. This is a limitation.
Parameters	Select the Data Fields you want to pass as parameters for the selected Data Mapping definition. For information on creating Data Fields, see the Data Fields section.

8.2.7 RunExecutable

Figure 59: Dynamic Parameters for RUN EXECUTABLE



The screenshot shows a configuration dialog for the 'RUN EXECUTABLE' task. It contains five input fields:

- Datastore Name:** OFSAAAIINFO
- Executable:** (empty)
- Wait:** No
- Batch Parameter:** No
- Optional Parameter:** (empty)

Each field has a question mark icon next to it, indicating help or documentation.

Table 24: Dynamic Parameters for RUN EXECUTABLE Description

Field Name	Description
Datastore Name	Refers to the name of the Information Domain. By default, the Information Domain to which you are connected is selected.

Field Name	Description
Executable	<p>Refers to the executable path on the DB Server. The Executable parameter contains the executable name as well as the parameters to the executable. These executable parameters have to be specified as they are specified at a command line. In other words, the Executable parameter is the exact command line required to execute the executable file.</p> <p>The path to the executable has been entered in quotes. Quotes have to be used if the exe name has a space included in it. In other words, the details entered here should look exactly as you would enter it in the command window while calling your executable. The parameter value is case-sensitive. So, ensure that you take care of the spaces, quotes, and case. Additionally, commas are not allowed while defining the parameter value for executable.</p> <p>To pass parameters like \$RUNID, \$PHID, \$EXEID, \$RUNSK to the RUN EXECUTABLE component, specify RRFOPT=Y or rrfopt=y along with other executable details.</p>
Wait	<p>When the file is being executed you have the choice to either wait till the execution is completed or proceed with the next task.</p> <p>Select Y (Yes) or N (No) from the drop-down list.</p> <ul style="list-style-type: none"> • Y- Select this if you want to wait for the execution to be completed • N- Select this if you wish to proceed. <p>If the task is using FICGEN/RUN EXECUTABLE component and there is no precedence set for this task, then the WAIT should always be set to 'N'.</p>
Batch Parameter	<ul style="list-style-type: none"> • Y- Select Yes if you want to pass the Batch parameters to the shell script file being executed. <ul style="list-style-type: none"> ▪ If Wait is selected as Y and Batch Parameter is selected as Y, following parameters are passed to the executable: <code>NIL <BatchExeRunID> <ComponentId> <Task> <Infodate> <Infodom> <DatstoreType> <IPAddress></code> ▪ If Wait is selected as N and Batch Parameter is selected as Y, following parameters are passed to the executable: <code><BatchExeRunID> <ComponentId> <Task> <Infodate> <Infodom> <DatstoreType> <IPAddress></code> • N- Select No if the Batch parameters should not be passed to the shell script.

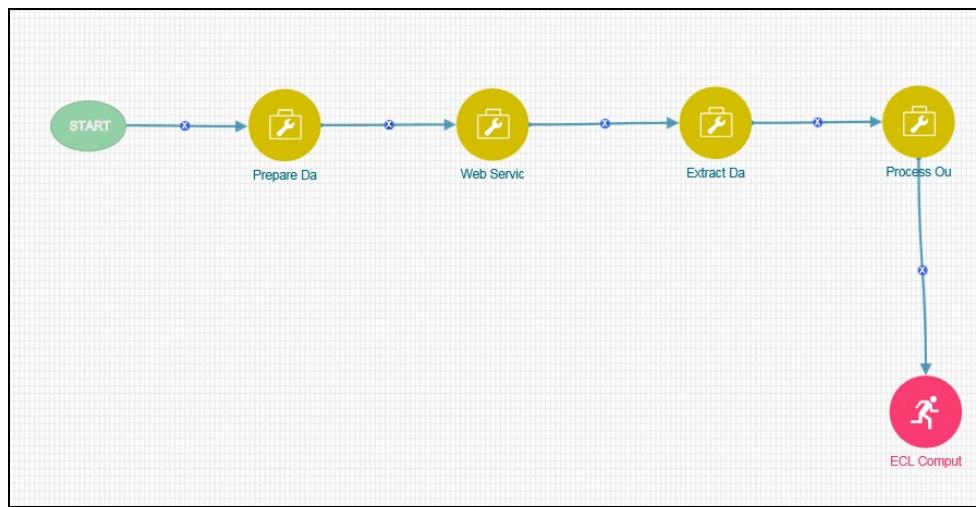
Field Name	Description
Optional Parameter	<p>The following runtime parameters can be passed during run execution:</p> <ul style="list-style-type: none">• \$RUNID• \$PHID• \$EXEID• \$RUNSK• \$MISDATE• \$BATCHRUNID <p>Values for the runtime parameters are implicitly passed while executing the Run definition.</p> <p>Note: The length of PHID is 255 characters and TASKID is 100 characters.</p>

9 Orchestrating External Models/Components in Your Process Flow

You can use external models or external components in your process flow by using Rest Service Application Rule or External Java API Application Rule.

9.1 How to invoke External Model through Web Service

Figure 60: Invoking External Web Service flow



A typical External Web Service invocation have the following steps:

1. Data Preparation
2. Web Service Invocation
3. Data Extraction

9.1.1 Data Preparation

Before invoking Web Service, we need to prepare data to be passed across to the Web Service. This can be done by configuring the Application Rule.

In the above example, data preparation is done through JSON Read From DB Application Rule as shown below, which reads data from a table, converts into JSON and stores output to JSON_DB_DATA Data Field.

Figure 61: Edit API Details

Edit API Details

Name	Read DB Data
RuleType	ExecutionRule
ExecutionType	JSON Read From DB
Table Name	DIM_ACCOUNT
Column List	N_ACCT_SKEY Account_ID,v_account_desc Account_Name
Where Condition	V_PROD_CODE='CME'
Return JSON Type	JSON Array
Output Datafield	JSON_DB_DATA
Scope	Process

Save

For more information, see [JSON Read From DB Application Rule](#) section.

9.1.2

Webservice Invocation

Consider you want to use an external model called ForecastModelPost, which is available in `http://<IP Address/hostname of the Web Server>:<servlet port>/<context name>/rest-api/v1/ModelWebService/ForecastModelPost`.

1. Launch the Process for which you want to add an external model.
2. Go to the *Application Rule* tab and click  **Add >Rest Service**.

Figure 62: Webservice Invocation Rule Details

Rule Details

Name: ModelWebService

Rule Type: ExecutionRule

Execution Type: Rest Service

Method Type: POST

Url: http://[REDACTED]/ofsaa/rest-api/v1/ModelWebService/ForecastModelPost

Query Param:

Headers:

Data: {"owner": {"id": 3730}, "scenario_name": "Baseline", "forecast_periods": 2, "data": ~~JSON_DB_DATA~~}

Return Parameter: TASK_RESPONSE

Scope: Process

Is Proxy Required: Yes

Save

3. Enter the details as shown. For more information on configuring a Rest Service, see the [Rest Service Application Rule](#) section.
4. In this example, the Data sent to the web service is a combination of both static and dynamic value (~~JSON_DB_DATA~~ is data field holding a JSON string which is prepared as the previous step in the pipeline.)
5. Click **Save**. The ModelWebService Application Rule gets created in your logged-in Information Domain.
6. Go to the Process Flow tab.
7. Click  **Service Task** from the *Activities* pane and click on the drawing canvas where you want to place it.
8. Double-click the Service Task icon.

Figure 63: Webservice Invocation Activity details

Activity ID: Job_1522122432701

Activity Name: WebService

Activity Desc: Activity for calling external model

Activity Type: AUTOMATIC

Status: (dropdown menu)

Outcomes: (dropdown menu)

9. Enter the Activity details such as Activity Name and Activity Description.
10. Click . The *Implementation* tab is displayed.
11. Select the information domain where the ModelWebService Application Rule is defined, from the **Infodom** drop-down list. The list displays all the Infodoms mapped to the applications configured in your OFSAA instance.
12. Click in the **Execution Rule** field. The *Participant Details* window is displayed with all Application Rules of Execution Rule types available in your Process.
13. Select ModelWebService and click **Ok**.

Figure 64: Webservice Invocation Participant Details

Rule

Infodom: FSDFINFO

Execution Rule: ModelWebService

Parameters

Add

Pre/Post Processing

Pre Rule: Default

Post Rule: Default

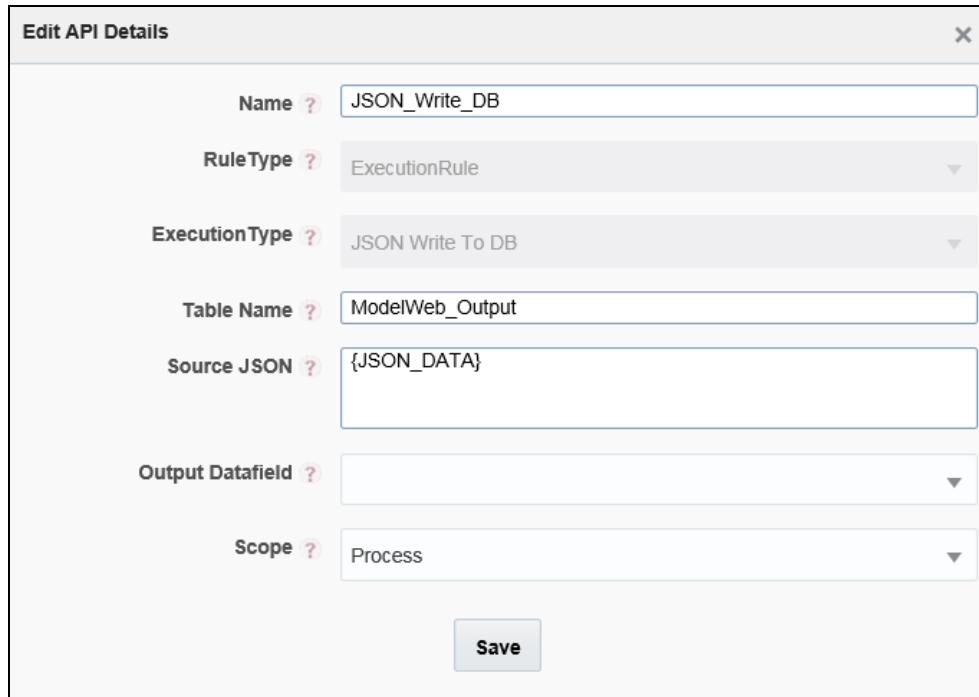
9.1.3 Data Extraction

The response from the Web Service needs to be processed depending on the application requirement. PMF has capabilities to process the JSON and store in the output table.

You can use JSON PATH expressions to extract the relevant information from the Web Service response. For more information on JSON PATH expressions, see [For JSON Path Expression Application Rule](#) section.

Similarly, the response can be stored back to the database using JSON Write To DB Application Rule as shown.

Figure 65: JSON Write To DB Application Rule- Edit API Details window



The screenshot shows the 'Edit API Details' window for a JSON Write To DB Application Rule. The window has a title bar 'Edit API Details' and a close button 'X'. It contains the following fields:

- Name**: JSON_Write_DB
- RuleType**: ExecutionRule
- ExecutionType**: JSON Write To DB
- Table Name**: ModelWeb_Output
- Source JSON**: {JSON_DATA}
- Output Datafield**: (empty)
- Scope**: Process

At the bottom is a 'Save' button.

For more information, see [JSON Write To DB Application Rule](#) section.

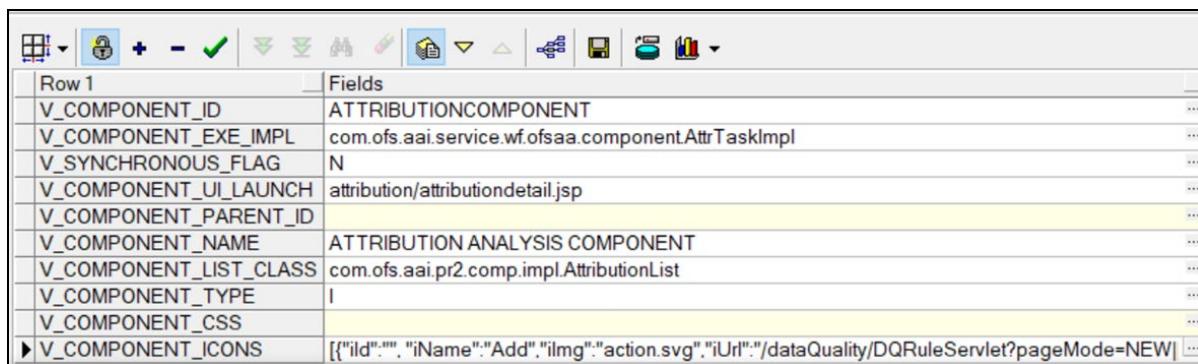
10 Configuring Custom Components

For configuring a new custom component as an OFSAA Widget, you need to have entries in the AAI_WF_COMPONENT_REGISTRATION table and AAI_WF_COMPONENT_PARAMETERS table. Additionally, the executeComponent() method needs to be implemented in the Implementation class and the jar should be present in web-inf/lib path.

10.1 AAI_WF_COMPONENT_REGISTRATION Table

An entry into the AAI_WF_COMPONENT_REGISTRATION table will create a new custom component in the PMF OFSAA widgets.

Figure 66: AAI_WF_COMPONENT_REGISTRATION table

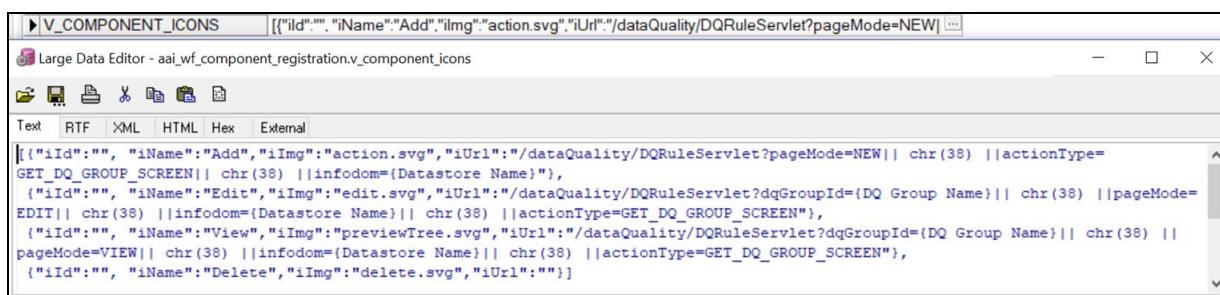


Row 1	Fields
V_COMPONENT_ID	ATTRIBUTIONCOMPONENT
V_COMPONENT_EXE_IMPL	com.ofs.aai.service.wf.ofsaa.component.AttrTaskImpl
V_SYNCHRONOUS_FLAG	N
V_COMPONENT_UI_LAUNCH	attribution/attributiondetail.jsp
V_COMPONENT_PARENT_ID	
V_COMPONENT_NAME	ATTRIBUTION ANALYSIS COMPONENT
V_COMPONENT_LIST_CLASS	com.ofs.aai.pr2.comp.impl.AttributionList
V_COMPONENT_TYPE	I
V_COMPONENT_CSS	
V_COMPONENT_ICONS	[{"iId": "", "iName": "Add", "iImg": "action.svg", "iUrl": "/dataQuality/DQRuleServlet?pageMode=NEW"}]

- V_COMPONENT_ID – Specify a unique ID for the component.
- V_COMPONENT_EXE_IMPL - This is the interface that needs to be implemented by the component owner, so that during the execution of this component, this API is invoked.
- V_COMPONENT_UI_LAUNCH - This is the URL of the page that needs to be opened when the component is double clicked in the canvas.
- V_COMPONENT_NAME – Specify a name for the component, which is displayed for the custom component in the OFSAA Widgets.
- V_COMPONENT_ICONS - All the icons and menu are configured in this field.

For example:

Figure 67: AAI_WF_COMPONENT_REGISTRATION Example table



V_COMPONENT_ICONS	[{"iId": "", "iName": "Add", "iImg": "action.svg", "iUrl": "/dataQuality/DQRuleServlet?pageMode=NEW"}]
Large Data Editor - aai_wf_component_registration.v_component_icons	
	
Text RTF XML HTML Hex External	
<pre>[{"iId": "", "iName": "Add", "iImg": "action.svg", "iUrl": "/dataQuality/DQRuleServlet?pageMode=NEW"}]</pre>	

10.2 AAI_WF_COMPONENT_PARAMETERS Table

The AAI_WF_COMPONENT_PARAMETERS table needs to be populated with the relevant parameters:

Figure 68: AAI_WF_COMPONENT_PARAMETERS table

	V_COMPONENT_ID	V_PARAMETER_NAME	I_PARAMETER_ORDER	I_DISPLAY_ORDER	V_PARAMETER_CODE	TYPE_OF_DISPLAY
► 1	LOAD DATA	... IP Address	...	1	3 SMSLB.IP.ADDR	... 2 ...
2	LOAD DATA	... Datastore Type	...	2	1 ICC.FE.LBL_DATA_STORE_TYPE	... 2 ...
3	LOAD DATA	... Datastore Name	...	3	2 ICC.FE.LBL_DATA_STORE	... 2 ...
4	LOAD DATA	... Load Mode	...	4	4 ICC.FE.LBL_LOADMODE	... 2 ...
5	LOAD DATA	... Source Name	...	5	5 SMSLB.SRC_NAME	... 2 ...
6	LOAD DATA	... File Name	...	6	6 GLOBAL.FIL_NAM	... 2 ...
7	LOAD DATA	... Data File Name	...	7	7 ETLPROP_DEF_LABEL_OR_TEXT_DATA_FILE_NAME	... 1 ...
8	LOAD DATA	... Default Value	...	8	8 DTDQ.DFLT_VAL	... 1 ...

V_DEFAULT_VALUE	V_CLASS_NAME	V_METHOD_NAME
...
...
...
...
...	com.ofs.aai.service.wf.ofsaa.component.T2TTaskImpl	...
...	com.ofs.aai.service.wf.ofsaa.component.T2TTaskImpl	...
...
...

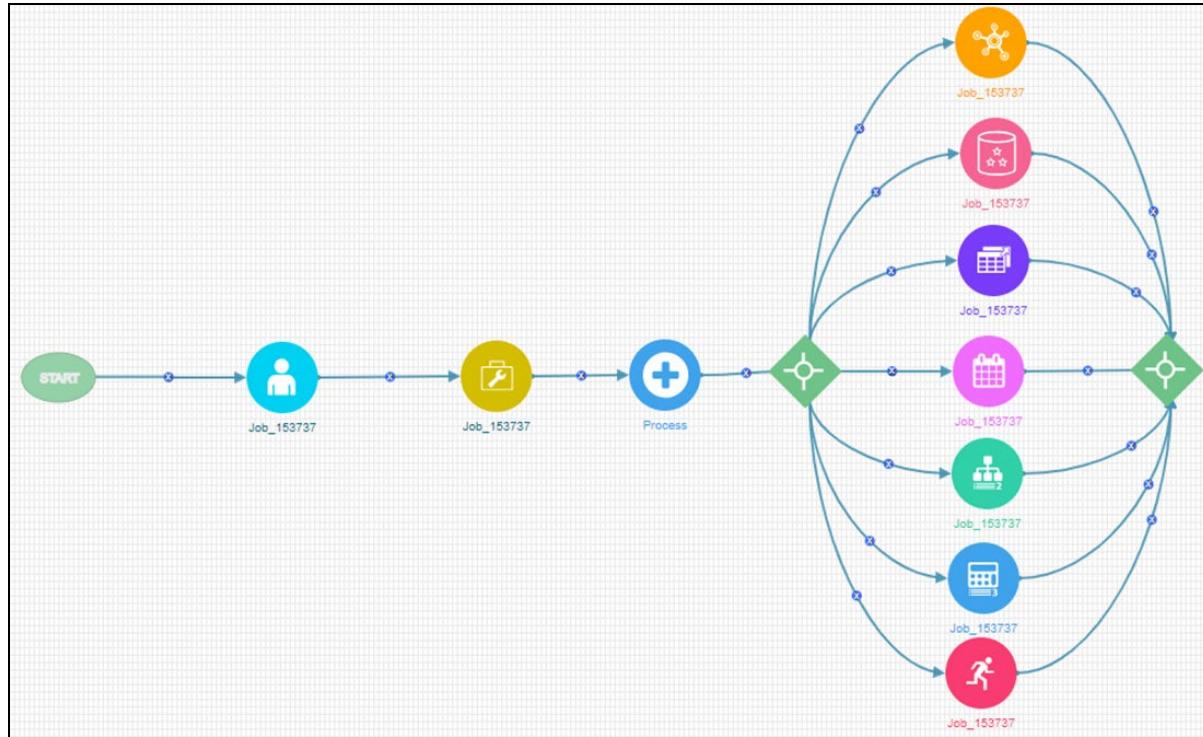
- V_COMPONENT_ID - Specify the ID for the component. It should be same as that is given in the AAI_WF_COMPONENT_REGISTRATION table.
- V_PARAMETER_NAME – Specify the Parameter name which will be required by the component]
- I_PARAMETER_ORDER – Specify the Parameter order.
- I_DISPLAY_ORDER -- Specify the order in which the Parameters need to be displayed in the UI.
- V_PARAMETER_CODE – Specify Parameter code.
- TYPE_OF_DISPLAY – Specify the type of display for the parameter. For example: if text input required the value should be 1 or selection drop-down value should be 2]
- V_DEFAULT_VALUE – Specify if any default value needs to be provided for the parameter.
- V_CLASS_NAME – Specify the class name for listing the drop-down values.
- V_METHOD_NAME - Specify the name of the method for listing values.

11

Executing Parallel Tasks

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

Figure 69: Executing Parallel Tasks flow

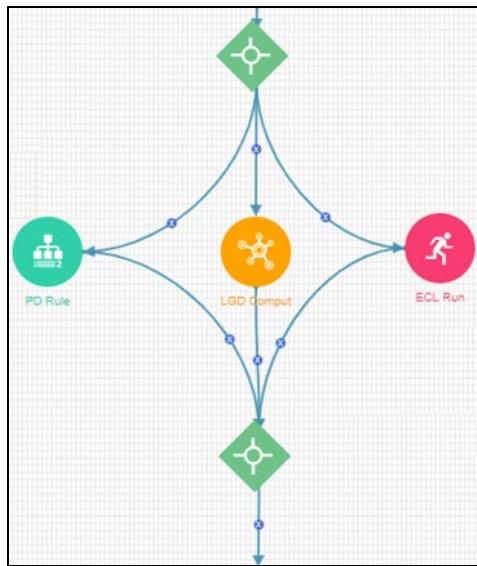


In the shown example, after  is executed, Parallel Gateway  gets invoked. All the OFSAA components, which are placed between Parallel Gateways,  are executed simultaneously. It waits until all components are executed, and then moves to the next activity in the Process Flow.

11.1

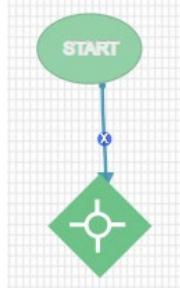
How to Use Parallel Gateways

Figure 70: RuleType 2, MFModel and Run widget Parallel Execution flow



This section explains how to design three tasks such as RuleType 2, MFModel and Run widget to be executed in parallel.

1. Create a Process Modelling Process.
2. Launch the Process. It displays the *Process Flow* tab.
3. Click  **Start** from the *Tools* pane in the floating toolbar and click on the drawing canvas where you want to place it. This Start activity indicates the beginning of the Process.
4. Click  **Parallel Gateway** from the *Tools* pane in the floating toolbar and click on the drawing canvas where you want to place it.
5. Click  **Transition Curve** and click **Start** (from where the transition starts) and click  (the activity to which you want to have the transition).



6. Click  **RuleType2** from the *OFSAW Widgets* pane, and then click the drawing canvas where you want to place it.

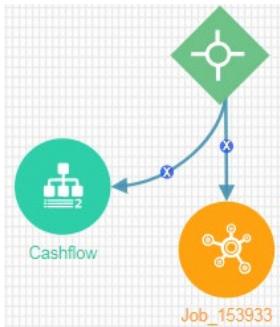
7. Click  **Transition Curve** from the *Tools* pane and click  Parallel Gateway and then click



8. Configure the RuleType2 widget. For more information, see [Configuring OFSAA Tasks in Your Process Flow](#) section.

9. Click  **MFModel** from the *OFSAA Widgets* pane, and then click the drawing canvas where you want to place it.

10. Click  **Transition Curve** from the *Tools* pane and click  Parallel Gateway and then click

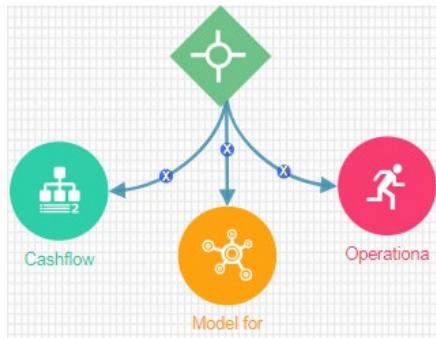


11. Configure the MFModel widget. For more information, see [Configuring OFSAA Tasks in Your Process Flow](#) section.

12. Click  **Run** from the *OFSAA Widgets* pane, and then click the drawing canvas where you want to place it.

13. Click  **Transition Curve** from the *Tools* pane and click  Parallel Gateway and then click





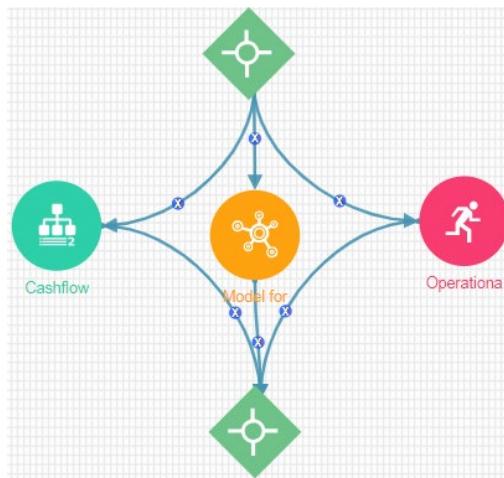
14. Configure the Run widget. For more information, see [Configuring OFSAA Tasks in Your Process Flow](#) section.



15. Click Parallel Gateway from the *Tools* pane in the floating toolbar and click on the drawing canvas where you want to place it.



16. Draw transitions from RuleType2, MFModel and Run widgets to the merging Parallel Gateway as shown:

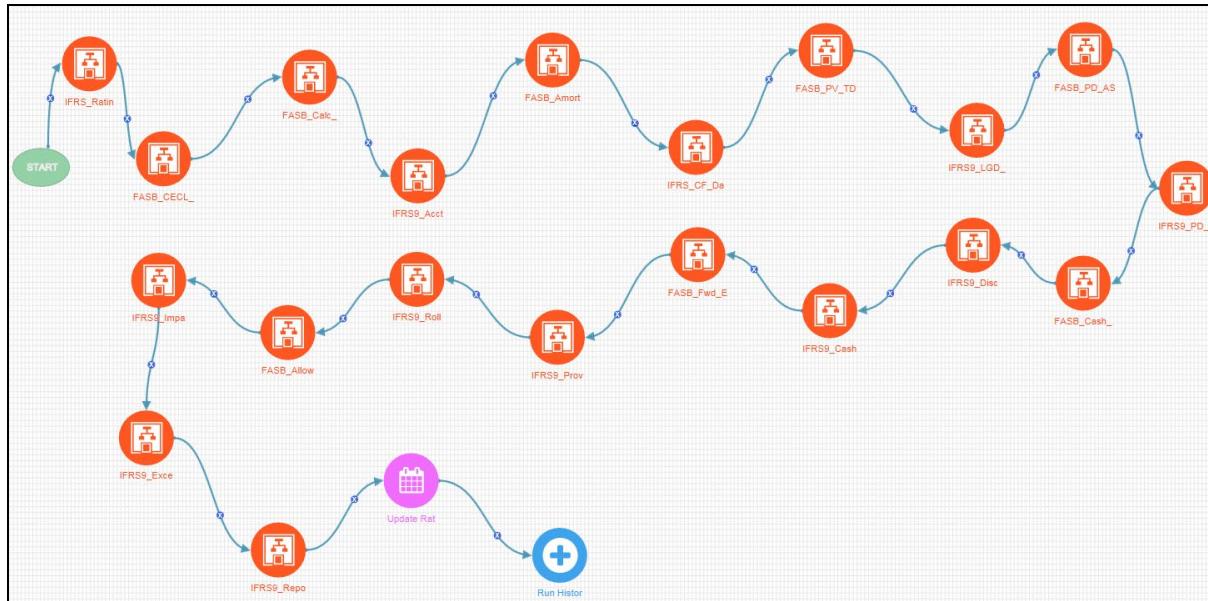


You can drag and position the widgets to avoid overlapping of widgets or transition lines.

12 Calling another Pipeline from Your Parent Pipeline

Reusability is important while designing your pipeline. Sub Pipeline is the mechanism in the PMF to call another pipeline from your parent pipeline. You can add filters to a Sub Pipeline. Then all Run Pipelines that are using the Sub Pipeline inherit those filters used in the Sub Pipeline.

Figure 71: Calling another Pipeline from Your Parent Pipeline



12.1 How to Configure Sub Pipeline

1. From the *Process Flow* tab, click  **Sub Pipeline** under *Activities* toolbar in the left pane and click the drawing canvas where you want to place it.



2. Double-click  **IFRS_Ratin**. The *Sub Process Details* window is displayed.

Figure 72: Sub Process Details window

Sub Process Details

Activity Name	Regulatory Hub
Activity Desc	Actual Data Load
App Package ID	Platform
Process ID	Actual_Pipeline-0
Object ID	USERID
Object Type	STATUS
Data field pass	<input type="checkbox"/> Yes
Calling Mode	<input checked="" type="radio"/> Synchronous <input type="radio"/> Asynchronous

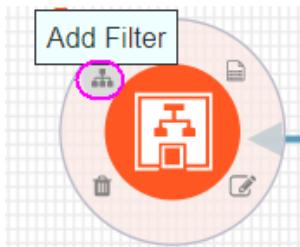
3. Enter the details as given in the table:

Table 25: Sub Process Details Description

Field Name	Description
App Package ID	Select the Application Package from which you want to call a process, from the drop-down list. The package IDs that are seeded from backend are displayed in the list.
Process ID	Select the Process that you want to call within your workflow, from the drop-down list. The list displays all processes defined for the selected Application Package.
Object ID	Select the Object ID from the drop-down list. Object ID is the entity ID used to identify if a workflow needs to be started from the beginning or from the current stage. Object ID drop-down list is populated from the Data Fields.
Object Type	Select the object type from the drop-down list. Object Type is used to identify a workflow that is passed by the application.
Data field pass	Select Yes to pass the parameters passed to the parent workflow to be passed to the selected sub pipeline.

Field Name	Description
Calling Mode	<ul style="list-style-type: none">• Synchronous – Select this option if you want the sub pipeline to complete before the flow moves to the next activity.• Asynchronous – Select this option if you do not want to wait for the sub pipeline to complete. In addition, the parent workflow proceeds to the next activity.

4. Mouse-over the Sub Pipeline icon to display the sub menu.



5. Click  to display the Filter details. For more information on selecting Filters, see [Filter Details](#).

13 Configuring Email for Human Tasks

Following table entries should be made for setting Email notification:

- [AAI EMAIL CONFIG Table](#)
- [AAI WF APP PACKAGE B Table](#)
- [AAI WF APP REGISTRATION Table](#)
- [AAI WF ACTIVITY TASK B Table](#)
- [AAI USER PREFERENCE Table](#)
- [AAI WF EMAIL TEMPLATE Table](#)
- [AAI WF BULK MAIL TRIGGER Table](#)
- [CSSMS USR PROFILE Table](#)
- [AAI MAIL AUDIT TRAIL Table](#)

13.1 AAI_EMAIL_CONFIG Table

This table holds the SMTP server configurations.

- V_PROTOCOL - SMTP
- V_HOST - SMTP/ Mail Server ID

NOTE

If the SMTP HostName does not function or displays as invalid, use the IP address of the SMTP server in the AAI_EMAIL_CONFIG table.

- V_PORT - SMTP Server Port
- V_AUTHENTICATION - Either False or True
- V_USER_NAME - Login name to SMTP/ Mail Server ID from which mail is triggered. This is required if V_AUTHENTICATION is set as True.
- V_PASSWORD - Password to login into SMTP/ Mail Server. This is required if V_AUTHENTICATION is set as True.

Figure 73: AAI_EMAIL_CONFIG Table



	V_PROTOCOL	V_HOST	V_PORT	V_AUTHENTICATION	V_USER_NAME	V_PASSWORD	V_SECURITY
▶ 1	smtp	internal-mail-router.oracle.com	25	false

13.2 AAI_WF_APP_PACKAGE_B Table

The AAI_WF_APP_PACKAGE_B table entry is for configuring email at Application Level.

- Set the V_EMAIL_REQUIRED parameter value to Y in the AAI_WF_APP_PACKAGE_B table.

- Set `V_EMAIL_TYPE` as:
 - 0 – To receive no notification mails
 - 1 – To get mails instantly
 - 2 – To get bulk mail (Additionally, you need to set `V_BULK_MAIL_TRIGGER` value to Y in the `AAI_WF_BULK_MAIL_TRIGGER` table). A single mail is sent with all the pending notifications from the last trigger, as a PDF attachment. After the bulk mail is sent, `V_BULK_MAIL_TRIGGER` value is automatically set to N.
 - 3 – To get mail with attachment

`V_BULK_TEMP` – This is used to set the template for the bulk emails. You can keep this as blank if bulk email is not set.

Figure 74: AAI_WF_APP_PACKAGE_B Table

V_APP_PACKAGE_ID	V_APP_PACKAGE_DESC	V_IS_EMAIL_REQUIRED	V_EMAIL_TYPE	V_BULK_TEMP	V_LANDING_PAGE_URL	V_DEFINITION_PAGE_URL
1 100	Platform	N
2 10	Business Restructure	Y
3 11	Questionnaire	Y
4 3	Expense Management	Y	1	5

13.3 AAI_WF_APP_REGISTRATION Table

This table holds email configuration at module or entity type level.

Set the `V_IS_EMAIL_REQUIRED` parameter value to Y in `AAI_WF_APP_REGISTRATION` table.

Figure 75: AAI_WF_APP_REGISTRATION Table

V_APP_PACKAGE_ID	V_OBJECT_TYPE	V_IS_EMAIL_REQUIRED
1 10	BR	...
2 11	QTNR	...
3 100	MD_1	...
4 3	1000	Y

13.4 AAI_WF_ACTIVITY_TASK_B Table

The `AAI_WF_ACTIVITY_TASK_B` table holds the email configuration at each activity (Task & notification) level.

Set the `V_EMAIL_REQUIRED` parameter value to Y against the task.

Additionally, you can set the `V_EMAIL_TEMPLATE` value based on `AAI_WF_EMAIL_TEMPLATE` table.

Figure 76: AAI_WF_ACTIVITY_TASK_B Table

	V_PROCESS_ID	V_ACTIVITY_ID	V_TASK_ID	V_CONDITION_TYPE	V_EMAIL_REQUIRED	V_EMAIL_TEMPLATE
1	New	Job_1539062920201	1539062927787	2	Y	4
2	BR1	Activity_1430138133131	1430304038218			
3	BR1	Activity_1430138133131	1430304084815			
4	QTNR	Job_1468916517232	1496309128751	2	Y	2
5	QTNR	Job_1468916574725	1496295700502		Y	2
6	QTNR	Job_1496226571444	1496296080165	2	Y	2
7	QTNR	Job_1496226679422	1496309653931	2	Y	3
8	MD_1	Job_1472554718819	1472554888526	2	N	1

13.5 AAI_USER_PREFERENCE Table

In this table, you can set the user preference for receiving the notification emails.

Table 26: Email Notification User Preference

V_USER_ID	N_EMAIL_NOTIF_REQ
USER1	1
USER2	2

- 0 – To receive no notification emails
- 1 – To get mails instantly
- 2 – To get bulk mail (Additionally, you need to set V_BULK_MAIL_TRIGGER value to Y in the AAI_WF_BULK_MAIL_TRIGGER table). A single mail is sent with all the pending notifications from the last trigger, as a PDF attachment. After the bulk mail is sent, V_BULK_MAIL_TRIGGER value is automatically set to N.
- 3 – To get mail with attachment

13.6 AAI_WF_EMAIL_TEMPLATE Table

This table is used to provide the template for the email that needs to be sent.

Figure 77: AAI_WF_EMAIL_TEMPLATE Table

V_MAIL_FROM	V_MAIL_MESSAGE	V_MAIL_SUBJECT	V_APP_PACKAGE_ID	V_MAIL_TYPE	N_TEMPLATE_ID
1 workflow_test@oracle.com	<html><body><p>Hi [USERID].</p><p>This is to inform you that a [TASK_TYPE] [ENTITYTYPE] [ENTITYID] [TITLE] ...</p>	[TASK_TYPE] [ENTITYTYPE] [ENTITYID] [TITLE] ...	0	Notification	1
2 workflow_test@oracle.com	<html><body><p>Hi [USERID].</p><p>This is to inform you that a [TASK_TYPE] [ENTITYTYPE] [ENTITYID] [TITLE] ...</p>	[TASK_TYPE] [ENTITYTYPE] [ENTITYID] [TITLE] ...	0	Task	4
3 workflow_test@oracle.com	<html><body><p>Hi [USERID].</p><p>This is to inform you that a mail dig ...</p>	[ENTITYTYPE] [ENAME] Mail Digest	0	BulkTask	5
4 workflow_test@oracle.com	<html><body><p>Hi [USERID].</p><p>This is to inform you that tasks are del ...</p>	Delegation Notification	0	Delegation	2

13.7 AAI_WF_BULK_MAIL_TRIGGER Table

If you have set N_EMAIL_NOTIF_REQ parameter to 2 in AAI_USER_PREFERENCE table, additionally you need to set V_BULK_MAIL_TRIGGER value to Y in the AAI_WF_BULK_MAIL_TRIGGER table.

13.8 CSSMS_USR_PROFILE Table

This table is used to store the email id of the users, to which the notification emails need to be sent.

Table 27: Email ID and User ID for Email Notification

V_USR_ID	V_EMAIL
USER1	user1@oracle.com
USER2	user2@oracle.com

13.9 AAI_MAIL_AUDIT_TRAIL Table

This is where all mail trails are stored. The status changes after mail is sent. This can be used for debugging purposes.

14

Process Monitor

The Process Monitor is used to monitor the current stage of the Process for different instances. After integration with an Application, the workflow can be invoked. After invoking it goes through all the stages defined. Process Monitor shows all the stages finished, current stage and stages to come if any. Your user group needs to be mapped to the function role WFMACC (Workflow Monitor Access) to access the Process Monitor window.

Figure 78: Process Monitor Window

Entity Name	Process Name	Process Description	Execution Start Time	Last Updated By	Status
SubProcess_Job_1550204890362	SubProcess_Job_1550204890362	Process Name: added Process Description: desc	14-FEB-19 08:32:35		0
SubProcess_Job_1550204890362	SubProcess_Job_1550204890362	Process Name: added Process Description: desc	18-FEB-19 07:25:50		0
SubProcess_Job_1550204890362	SubProcess_Job_1550204890362	Process Name: added Process Description: desc	14-FEB-19 08:32:03		0
SUB_003	SUB_PIPELINE_001	Process Name: SUB_PIPELINE_001 Process Description: SUB_PIPELINE_001	01-MAR-19 01:19:43	AAAIUSER	COMPLETED
SubProcess_Job_1551267767559	SubProcess_Job_1551267767559	Process Name: TEST_RESTART Process Description: TEST_RESTART	01-MAR-19 01:34:10	AAAIUSER	COMPLETED
SubProcess_Job_1551431328947	SubProcess_Job_1551431328947	Process Name: TEST_SEQ_SUB Process Description: TEST_SEQ_SUB	01-MAR-19 01:32:38	AAAIUSER	COMPLETED
SubProcess_Job_1551260772157	SubProcess_Job_1551260772157	Process Name: TEST_RESTART Process Description: TEST_RESTART	01-MAR-19 01:33:45	AAAIUSER	COMPLETED

Page 1 of 15 (1-10 of 145 items) | [K](#) [<](#) [>](#) [X](#)

This window displays all the Workflows that are invoked from the Application with details such as Entity Name, Entity ID, Process Name, Process Description, Execution Start Time, Last Execution Time, Last Updated By and Status.

Click to display the following sub menu for Run Pipeline:



- Abort- to abort an ongoing Run Pipeline.
- Resume- to resume a Run Pipeline
- Re-run- to execute a Run Pipeline again irrespective of the previous execution status.

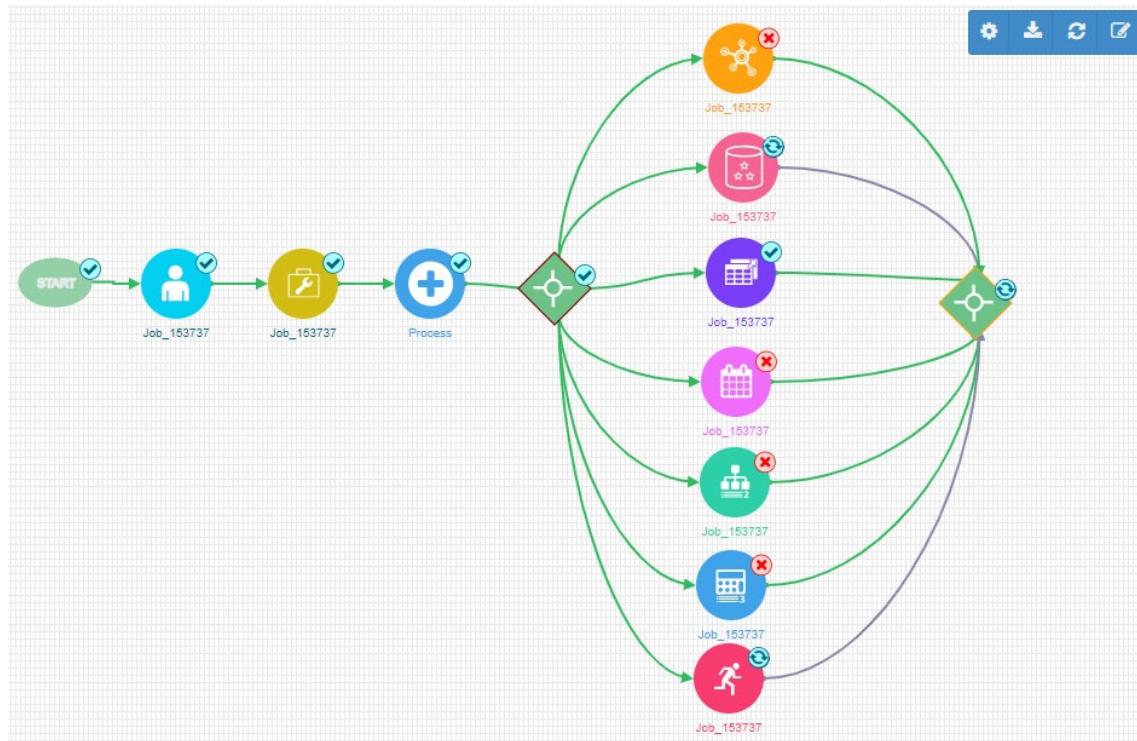
Using the *Search* grid, you can search for a specific Pipeline by providing a keyword from Process ID, Process Name or Process Description and clicking . Click  to reset the Search fields.

You can sort the Processes displayed in the *Process Monitor* window based on Entity Id, Entity Name, Process ID or Process Name. Click  to go to *Process Modeller* window.

14.1 Monitoring a Business Process

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

Figure 79: Monitoring a Business Process



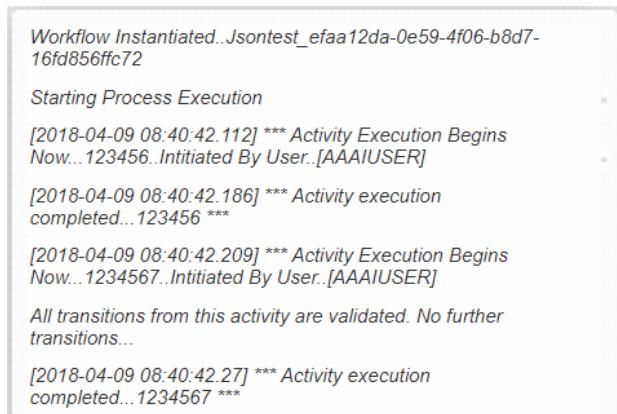
In the *Process Monitor* window, the status of the activity is represented as given:

-  - This indicates that the execution of activity is successful.
-  - This indicates that the activity is currently running or waiting for user's input to proceed.
-  - This indicates that the execution of activity is failed.
-  - This indicates that the activity is yet to be executed.

Double click the Sub Pipeline icon to monitor the tasks inside them.

- Click  to refresh the pane. Click  to view the *Process Logs* pane.

Figure 80: Process Logs pane



The *Process Logs* pane shows all the execution stages of the process. Click the Process Monitor pane to close the *Process Logs* pane.

You can use  and  to zoom in and out the *Process Monitor* window.

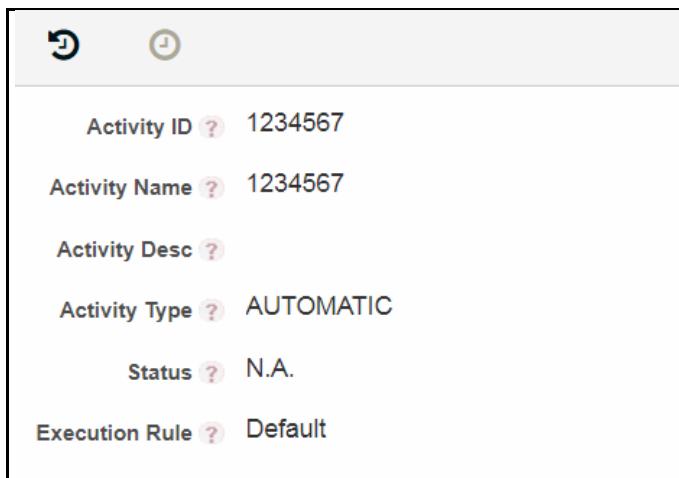
14.2 Viewing Activity Logs

This feature allows you to view logs of each activity from the Process Monitor window.

To view Activity logs

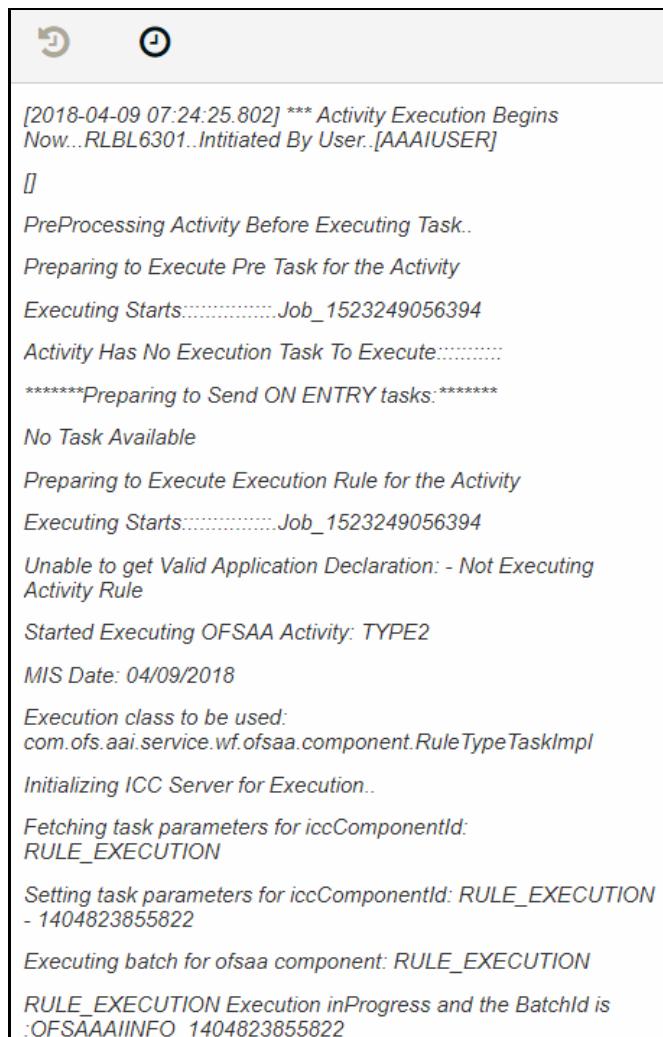
- Double-click on the activity icon whose logs you want to view. The Activity Definition details are displayed.

Figure 81: Activity Definition details window to View the Activity Logs



- Click  to view the Activity Logs.

Figure 82: Activity Logs pane



```

[2018-04-09 07:24:25.802] *** Activity Execution Begins
Now...RLBL6301..Initiated By User.[AAAIUSER]
[]

PreProcessing Activity Before Executing Task..
Preparing to Execute Pre Task for the Activity
Executing Starts.....Job_1523249056394
Activity Has No Execution Task To Execute.....
*****Preparing to Send ON ENTRY tasks:*****
No Task Available
Preparing to Execute Execution Rule for the Activity
Executing Starts.....Job_1523249056394
Unable to get Valid Application Declaration: - Not Executing
Activity Rule
Started Executing OFSAA Activity: TYPE2
MIS Date: 04/09/2018
Execution class to be used:
com.ofs.aai.service.wf.ofsaa.component.RuleTypeTaskImpl
Initializing ICC Server for Execution..
Fetching task parameters for iccComponentId:
RULE_EXECUTION
Setting task parameters for iccComponentId: RULE_EXECUTION
- 1404823855822
Executing batch for ofsaa component: RULE_EXECUTION
RULE_EXECUTION Execution inProgress and the BatchId is
:OFSAAINFO_1404823855822

```

The log shows all the execution stages of the selected Activity, if it is already executed. Otherwise, it is blank.

14.3 Viewing Execution Log for OFSAA Tasks

This option allows you to view the execution logs for OFSAA widgets. This option is not available for Service tasks or human tasks.

To view Execution logs

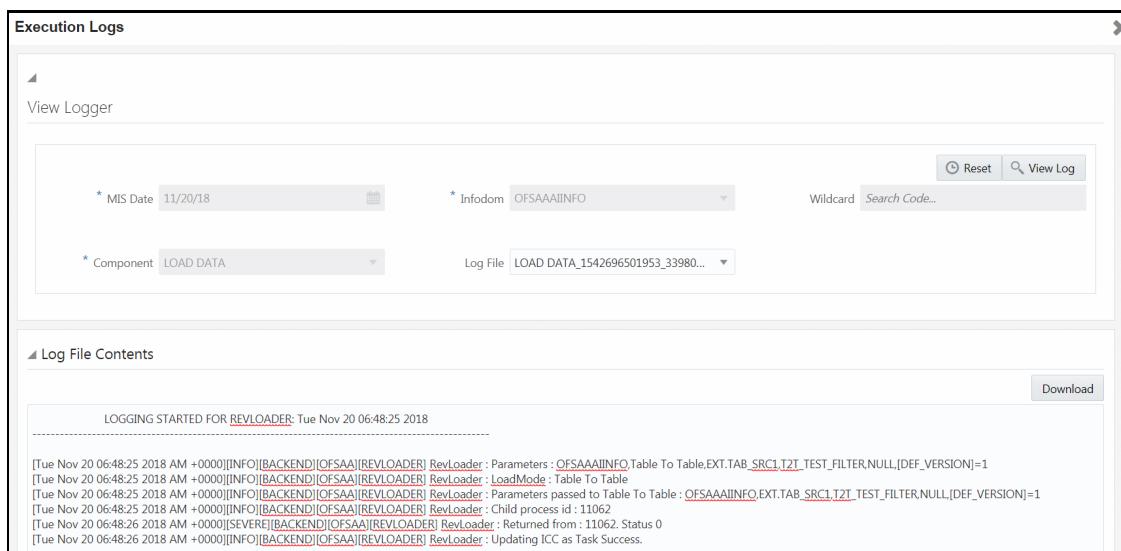
1. Double-click on the activity icon whose logs you want to view. The Activity Definition details are displayed.

Figure 83: Activity Definition details window to View the Execution Log for OFSAA Tasks



2. Click **Execution Log**.
3. Select the required **Log File** from the drop-down list and click **View Log**. The log information is displayed in the *Log File Contents* pane

Figure 84: Log File Contents pane



4. Click **Download** to download the execution log details.

15 Invoking PMF Pipeline

You can invoke a PMF pipeline using the following ways:

- [Application UI](#)
- [Within PMF Summary Screen UI](#)
- [Operations Module](#)
- [Command Line Execution](#)

15.1 Application UI

Application can invoke PMF in the following ways:

15.1.1 Java API

```
WorkflowEngineAPI.startWorkflowProcess( String objectId, String
objectType , String infodom, String segment, String userID, String
locale,Map<String, String> applicationParams,Map securityMap)
```

15.1.2 Stored Procedure

```
create or replace procedure startWorkflowProcessAsynch (objectId      IN
VARCHAR2,
          objectType  IN VARCHAR2,
          infodom     IN VARCHAR2,
          segments    IN VARCHAR2,
          userID      IN VARCHAR2,
          locale      IN VARCHAR2,
          appParams   IN array_varchar,
          secMap      IN array_varchar) is
```

15.1.3 Rest Service

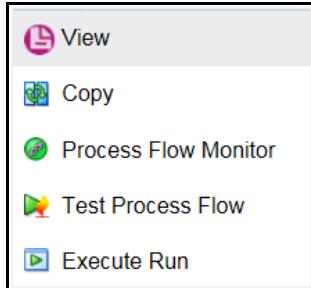
```
URL: <contextPath>/PMFService/startWorkflowProcess
Method: POST
Consumes("application/json")
Produces("text/plain")
Sample Input Params:
"{{\"SummaryPayload\":{\"objectid\":\"123\",\"objecttype\":\"QTNR\",\"infodom\":\"OFSAAAIINFO\",\"segment\":\"OFSAAAIINFO\",\"userid\":\"AAAIUSER\",\"locale\":\"en_US\",\"securitymap\":
:{},\"applicationparams\":
:{},\"testparam\":\"value1\",\"testparam2\":\"value2\"}}}"
```

15.2 Within PMF Summary Screen UI

15.2.1 Using Test Workflow

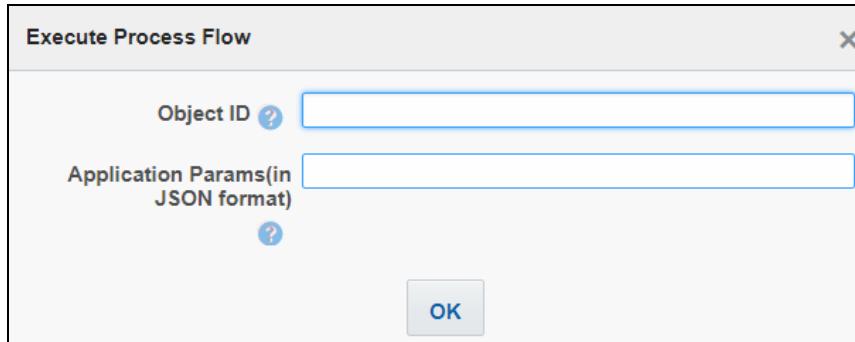
Testing Process Flow option is used to check whether the Process flow you designed works as expected.

1. From the *Process Modeller* window, click  corresponding to the Pipeline you want to test. The Submenu is displayed.



2. Click **Test Process Flow**. The *Execute Process Flow* window is displayed.

Figure 85: Execute Process Flow window



3. Enter an **Object ID** to identify the Process flow. This is displayed as Entity ID in the *Process Monitor* window.
4. Enter the **Application Parameters** that are required in the Process flow that you are testing. Specify parameters in JSON format. This is an optional field.
For example, `{"WF_RUNSK": "15", "WF_MISDATE": "12/31/1999"}`

5. Click **OK**.

15.2.2 Using Execute Run

See the [Executing Run Pipeline](#) section on how to invoke a Run pipeline from the PMF screen.

15.3 Operations Module

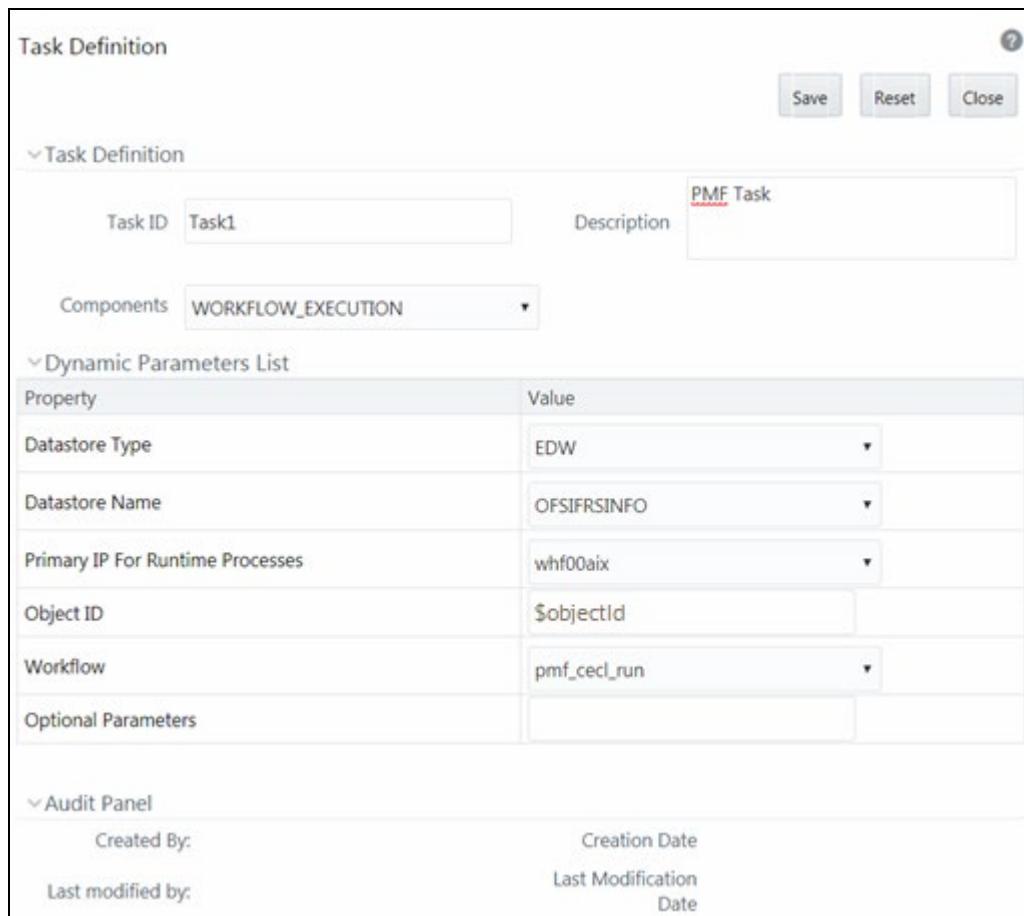
Execution of a PMF Pipeline can be triggered from the Operations module as a batch.

1. From the *Batch Maintenance* window under the Operations module, create a new Batch. For more information, see *Adding Batch Definition* section in the [OFS Analytical Applications Infrastructure User Guide](#).
2. Create a new Task with task component as Workflow Execution. For more information, see *Adding Task Details* section in the [OFS Analytical Applications Infrastructure User Guide](#).

NOTE

To avoid entering the Object ID each time you run a batch, enter the value \$objectId in the **Object ID** field. This automatically generates the ID in the OFSAA system.

Figure 86: Task Definition Window to Create a New Task



The screenshot shows the 'Task Definition' window. At the top, there are 'Save', 'Reset', and 'Close' buttons. The 'Task ID' field is set to 'Task1'. The 'Components' dropdown is set to 'WORKFLOW_EXECUTION'. The 'Dynamic Parameters List' section contains the following entries:

Property	Value
Datastore Type	EDW
Datastore Name	OFSSIFRSINFO
Primary IP For Runtime Processes	whf00aix
Object ID	\$objectId
Workflow	pmf_cecl_run
Optional Parameters	

The 'Audit Panel' section shows 'Created By:' and 'Last modified by:' fields, and 'Creation Date' and 'Last Modification Date' fields.

- a. Select the PMF Pipeline you want to execute from the **Workflow** drop-down list.
- b. Enter any parameters you want to pass during the execution of the Pipeline in the **Optional Parameters** field.
- c. Click **Save**.
3. Execute the Batch from the *Batch Execution* window. For more information, see the *Batch Execution* section in the [OFS Analytical Applications Infrastructure User Guide](#).

15.4 Command Line Execution

A shell script file `wfExec.sh` is available in the `ficdb/bin` folder.

To execute the utility, navigate to `$FIC_DB_HOME/bin` and execute `wfExec.sh` with parameters such as `objectId`, `objectType`, `Infodom`, `segments`, `userID`, `locale`, `appParams`, and `secMap`.

16 Appendix A

Topics:

- [Configuring Group Approval for Human Tasks](#)
- [JsonPath Expressions](#)
- [Delegation](#)

16.1 Configuring Group Approval for Human Tasks

Group approval/Group Consensus can be used to decide whether the flow has to move to a particular activity based on the response of a single member, majority of members of the group, or all members of the group. If more than one group is present, then you can design to move the flow to a particular activity based on the response of a single group, all groups or the majority of groups.

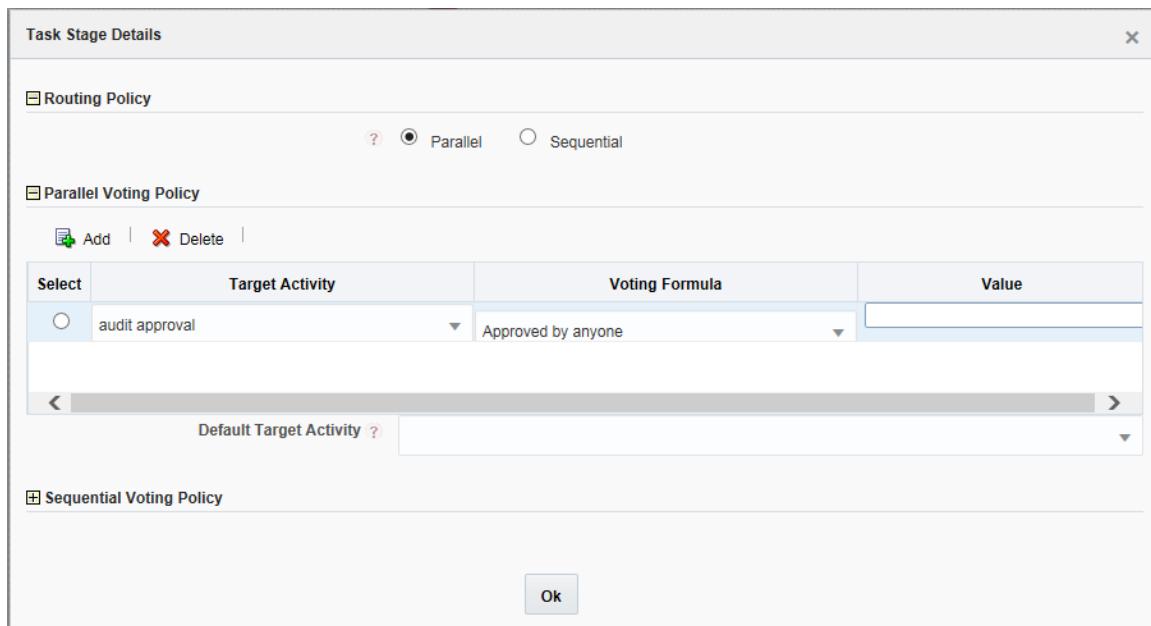
16.1.1 Configuring Parallel Group Approval

Parallel group approval is used when you want to send the task to all users in the task group simultaneously.

To configure parallel group approval:

1. From the *Process Modeller* window, select the required Process and click **Edit**. The *Process Flow* tab is displayed.
2. Double-click the Activity for which you want to configure user approval.
3. On the right pane, click . The *Actions* tab is displayed.
4. Click **Group Approval** to configure group approval. The *Task Stage Details* window is displayed.

Figure 87: Task Stage Details Window to configure Parallel Group Approval



5. Select **Parallel** to configure parallel group approval.
6. Click **Add**. A row is added to define the voting formula for target activity.

NOTE

It is recommended that you define the voting formula for all activities. If a voting formula is not defined for an activity and if someone in a task group selects that activity, the workflow moves to that activity.

7. Select the required **Target Activity** from the drop-down list.
8. Select the required option from the **Voting Formula** drop-down list. The options are:
 - **Approved by anyone**- If any one of the users from any task group chooses the selected **Target Activity**, the flow moves to the selected activity. If no one chose it, it checks for the voting formula defined for the next Target Activity.
 - **Approved by anyone from every Group**- If at least one user from every task group chooses the selected **Target Activity**, the flow moves to that activity.
 - **Approved by overall majority**- If the majority of the users from all task groups choose the selected **Target Activity**, the flow moves to that activity. For example, if there are 2 task groups and 15 users in each group, then at least 16 users (majority of 30 users) should choose the selected **Target Activity**, for the flow to proceed to that activity.
 - **Approved by majority from each Group**- If the majority of the users from each task group choose the selected **Target Activity**, the flow moves to that activity. For example, consider there are 3 task groups and each group has 15 users, then from each group, at least 8 users should favor the Target Activity to move the flow to that activity.
 - **Approved by everyone**- All the users in all the task group should choose the selected **Target Activity** for the flow to move to that activity.
 - **Approved by overall percentage**- If the specified percentage of users in the task group chooses the selected **Target Activity**, the flow moves to that activity. Enter the percentage in the **Value** field.
9. Select the **Default Target Activity** from the drop-down list. This is the activity that is executed if none of the condition is satisfied.
10. Click **Ok**.

16.1.2 Configuring Sequential Group Approval

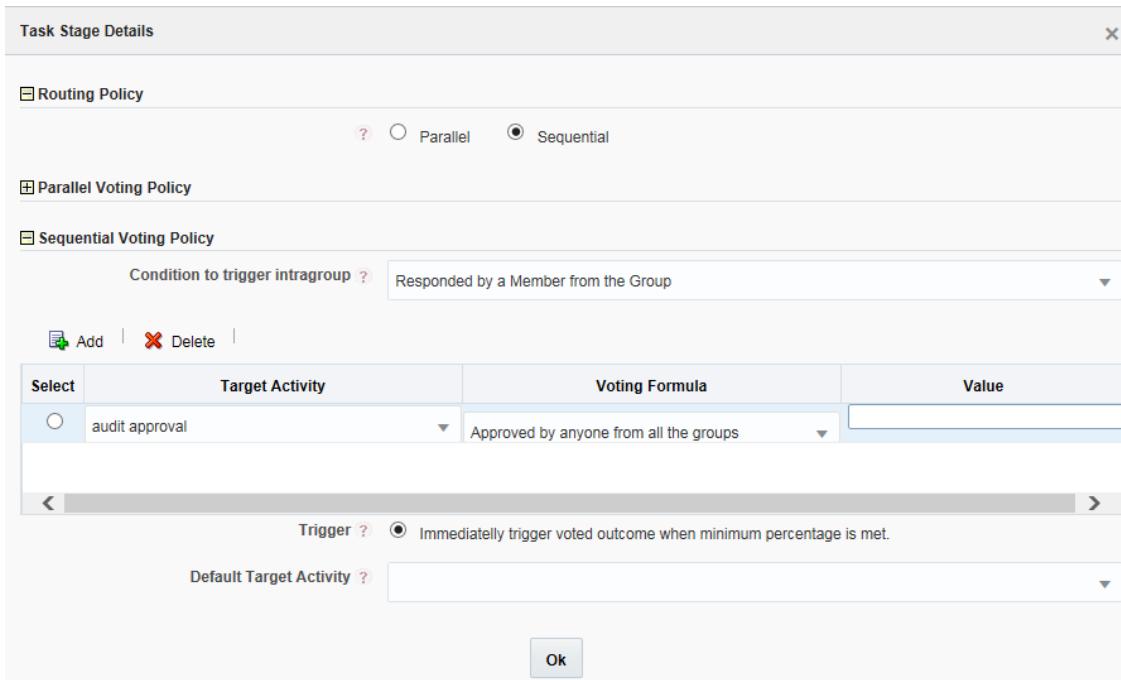
Sequential Group approval is used when you have multiple tasks for an activity.

To configure sequential group approval:

1. From the *Process Modeller* window, select the required Process and click **Edit**. The *Process Flow* tab is displayed.
2. Double-click the Activity for which you want to configure user approval.
3. On the right pane, click . The *Actions* tab is displayed.

4. Click **Group Approval** to configure group approval. The *Task Stage Details* window is displayed.

Figure 88: Task Stage Details Window to configure Sequential Group Approval



5. Select **Sequential** to configure sequential group approval.

6. Select the **Condition to trigger intragroup** from the drop-down list. Intragroup is the task group of each task in a particular activity. The sequence in which each intragroup will be considered for voting is based on the sequence in which the tasks are added in the *Task Stage* pane in the *Actions* tab. The options are:

- **Responded by a Member from the Group**- If anyone from the task group of the first task responds, it goes to the next task group and waits till someone from that task group respond and so on.
- **Responded by Overall Majority**- If majority of users in the task group of the first task respond, it goes to the next task group and waits till the majority of users in the task group of the second task respond and so on.
- **Responded by Everyone in each Group**- Once all users in the task group of the first task have responded, it moves to the next task group. Then it waits till everyone in the second task group responds and so on.

7. Click **Add**. A row is added to define the voting formula for target activity.

NOTE

It is recommended that you define a voting formula for all activities. If a voting formula is not defined for an activity and if someone in a task group selects that activity, the workflow moves to that activity.

8. Select the required **Default Target Activity** from the drop-down list. This is the activity which will be executed if none of the condition is satisfied.
9. Select the required option from the **Voting Formula** drop-down list. The options are:
 - **Approved by anyone from all the groups**- If anyone selects the Target Activity, the flow moves to that activity.
 - **Approved by Overall Majority**- If the majority of the users select the Target Activity, the flow moves to that activity.
 - **Approved by Everyone in each group**- If everyone in the group selects the Target Activity, the flow moves to that activity.
 - **Approved by Overall Percentage**- Provide the percentage in the Value field. If the specified percentage of the users select the Target Activity, the flow moves to that activity.
10. Select the activity which needs to be executed if the condition fails, from the **Default Target Activity** drop-down list.

16.2 JsonPath Expressions

JsonPath expressions always refer to a JSON structure in the same way as XPath expression is used in combination with an XML document. The "root member object" in JsonPath is always referred to as \$ regardless if it is an object or array.

JsonPath expressions can use the dot-notation

`$.store.book[0].title`

or the bracket-notation

`$['store']['book'][0]['title']`

16.2.1 Operators

Table 28: Operator Description table

Operator	Description
\$	The root element to query. This starts all path expressions.
@	The current node being processed by a filter predicate.
*	Wildcard. Available anywhere a name or numeric are required.
..	Deep scan. Available anywhere a name is required.
.<name>	Dot-notated child
['<name>', '<name>']	Bracket-notated child or children
[<number>, <number>]	Array index or indexes
[start:end]	Array slice operator

Operator	Description
[?(<expression>)]	Filter expression. The expression must evaluate to a boolean value.

16.2.2 Functions

Functions can be invoked at the tail end of a path - the input to a function is the output of the path expression. The function output is dictated by the function itself.

Table 29: Function Description table

Function	Description	Output
min()	Provides the min value of an array of numbers	Double
max()	Provides the max value of an array of numbers	Double
avg()	Provides the average value of an array of numbers	Double
stddev()	Provides the standard deviation value of an array of numbers	Double
length()	Provides the length of an array Integer	Integer

16.2.3 Filter Operators

Filters are logical expressions used to filter arrays. A typical filter would be [?(@.age > 18)] where @ represents the current item being processed. More complex filters can be created with logical operators && and ||. String literals must be enclosed by single or double quotes

([?(@.color == 'blue')] or [?(@.color == "blue")])

Table 30: Filter Operators table

Operator	Description
==	Left is equal to right (note that 1 is not equal to '1')
!=	Left is not equal to right
<	Left is less than right
<=	Left is less or equal to right
>	Left is greater than right
>=	Left is greater than or equal to right
=~	Left matches regular expression [?(@.name =~ /foo.*?/i)]
in	Left exists in right [?(@.size in ['S', 'M'])]

Operator	Description
nin	Left does not exist in right
subsetof	Left is a subset of right [?(@.sizes subsetof ['S', 'M', 'L'])]
size	Size of left (array or string) should match right
empty	Left (array or string) should be empty

16.2.4 Path Examples

```
{
  "store": {
    "book": [
      {
        "category": "reference",
        "author": "Nigel Rees",
        "title": "Sayings of the Century",
        "price": 8.95
      },
      {
        "category": "fiction",
        "author": "Evelyn Waugh",
        "title": "Sword of Honour",
        "price": 12.99
      },
      {
        "category": "fiction",
        "author": "Herman Melville",
        "title": "Moby Dick",
        "isbn": "0-553-21311-3",
        "price": 8.99
      },
      {
        "category": "fiction",
        "author": "J. R. R. Tolkien",
        "title": "The Lord of the Rings",
        "isbn": "0-395-19395-8",
        "price": 22.99
      }
    ]
  }
}
```

```

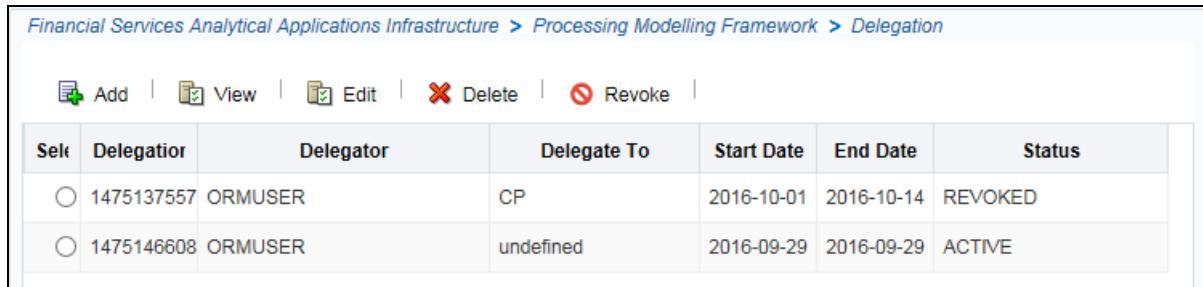
        }
    ],
    "bicycle": {
        "color": "red",
        "price": 19.95
    }
},
"expensive": 10
}

```

16.3 Delegation

This feature facilitates you to delegate the tasks/notifications assigned to you to another user. The delegate can be your peer, someone from your immediate subordinate or someone from your all subordinates. Additionally, you can revoke active delegations whenever required. Your user group should be mapped to the function role WFDELACC (Process Delegation User) if you want to define delegation.

Figure 89: Delegations defined by the Logged-in user



The screenshot shows a table titled 'Delegation' with the following data:

Sel	Delegator	Delegator	Delegate To	Start Date	End Date	Status
<input type="radio"/>	1475137557	ORMUSER	CP	2016-10-01	2016-10-14	REVOKED
<input type="radio"/>	1475146608	ORMUSER	undefined	2016-09-29	2016-09-29	ACTIVE

This window displays all the delegations that are defined by the logged-in user with details such as Delegation ID, Delegator, Delegate To, Start Date, End Date, and Status. You can add a new Delegation, view, modify, delete and revoke a delegation.

16.3.1 Adding a Delegate

To add a delegate

1. From the *Delegation* window, click  **Add**. The *Delegation Details* window is displayed.

Figure 90: Delegation Details Window

Delegation Details

Delegation ID ? 1477473481896

Delegator ? Tom Harley

Identify Delegate ?

Delegate To ?

Start Date ? 10/26/16

End Date ? 10/26/16

Notification Required ?

Filter ?

Notification Message ?

Scope ?

Application ?

Process ? All

Comments ?

2. Enter the details as tabulated:

Table 31: Delegation Details Description

Field Name	Description
Delegation ID	Displays the auto-generated Delegation ID.
Delegator	Displays the User ID of the logged-in user. If your user group is mapped to the function role WFDELGADM, you can select the delegator from the drop-down list.

Field Name	Description
Identify Delegate	<p>Select the required option from the drop-down list. The options are:</p> <ul style="list-style-type: none"> • Peers – Select this option if you want to delegate your tasks to your peer, who reports to your own manager. • Subordinates- Select this option if you want to delegate your tasks to your immediate subordinates. • All subordinates – Select this option if you want to delegate your tasks to someone who comes under you in your organization.
Delegate To	<p>Select the user to whom you want to delegate your tasks from the drop-down list. Based on the selected option from the Identify Delegate drop-down list, the users are displayed in this drop-down list. For example, if Peers is selected as Identify Delegate, this drop-down list displays all the peers in your organization. The data is fetched from the AAI_EMPLOYEE_MASTER table.</p>
Start Date and End Date	<p>Specify the duration for which you want to delegate your tasks by selecting the Start Date and End Date from the calendar.</p>
Notification Required	<p>Turn ON the toggle button if you want to send a notification to the delegate or delegator.</p>
Filter	<p>This field is enabled only if the Notification Required toggle button is turned ON.</p> <p>Select to whom you want to send the notification. You can set to send notification to Delegator and Delegate.</p> <p>Select when you want to send the notification. The options are On Defining and On Execution.</p>
Notification Message	<p>This field is enabled only if the Notification Required toggle button is turned ON.</p> <p>Select the notification message you want to send to the delegate or delegator.</p>
Scope	<p>Select the scope of the delegation from the drop-down lists. The options are:</p> <ul style="list-style-type: none"> • All- Select this option to delegate all your tasks. • Application- Select this option if you want to delegate all your tasks for a particular Application only. • Process- Select this option if you want to delegate all your tasks for a particular Process only.
Application	<p>This field is enabled only if Application or Process is selected as Scope.</p> <p>Select the required Application from the drop-down list. All your tasks related to the selected application is delegated to the selected user.</p>
Process	<p>This field is enabled only if Process is selected as Scope.</p> <p>Select the required Process from the drop-down list. The list displays all processes related to the selected Application. All your tasks related to the selected process is delegated to the selected user.</p>
Comments	<p>Enter if you want to add any comments for the delegation.</p>

3. Click **Save**.

16.3.2 Viewing Delegation

This option allows you to view the details of existing delegations.

From the *Delegation* window, select the required delegation and click **View**. You can view the Delegation details.

16.3.3 Modifying Delegate Details

To modify delegate details

1. From the *Delegation* window, select the delegation you want to modify and click  **Edit**. The *Delegation Details* window is displayed.
2. Modify the required details. For more information, see [Adding a Delegate](#) section.

16.3.4 Revoking Delegation

You can revoke only active delegations.

To revoke delegation

1. From the *Delegation* window, select the delegation you want to revoke and click  **Revoke**.
2. Click **OK** in the confirmation message box.

16.3.5 Deleting Delegation

You cannot delete active delegations.

To delete a delegation

1. From the *Delegation* window, select the delegation you want to delete and click  **Delete**.
2. Click **OK** in the confirmation message box.

17 Appendix B: Support APIs for Java External APIs

Topics:

- [Connection API](#)
- [Logging API](#)

17.1 Connection API

For establishing a connection with the Database, the ConnectionAdapter class provided by PMF can be used.

17.1.1 Jar Files Required

The following jar available at `$FIC_HOME/ficweb/webroot/WEB-INF/lib` folder contains the ConnectionAdapter class that contains connection related API's.

`aai-pmf-common.jar`

Referenced Files in Jar for Connection:

The `aai-pmf-common.jar` provides the following classes that can be used for query execution.

- ConnectionAdapter
- PreparedStatementDecorator

17.1.2 ConnectionAdapter Methods

The ConnectionAdapter class has the following API's:

```
public static Connection getDBConnections() - For Config Connection
public static Connection getDBConnections(String infodom, Boolean
isMetaConnection) - For Atomic Connection
public static void closeResultSet(ResultSet rs)
public static void closePreparedStatement(PreparedStatement ps)
public static void closeConnection(Connection conn)
public static void commitTransaction(Connection conn)
public static void rollBackTransaction(Connection conn)
```

17.1.3 Connection to Config Schema

To open a Config Schema connection, the `getDBConnections` method of ConnectionAdapter has to be invoked.

```
Connection configConn = ConnectionAdapter.getDBConnections();
```

For example:

```
public boolean testMethod(String attr1) {
    Connection configConn = null;
```

```

PreparedStatementDecorator prepStatement = null;
ResultSet rs = null;
try {
    configConn = ConnectionAdapter.getDBConnections();
    prepStatement = new PreparedStatementDecorator(configConn, query);
    prepStatement.setString(1, attr1);
    rs = prepStatement.executeQuery();
    while (rs.next()) {
        return true;
    }
}
catch (Exception e) {
    WorkflowUtil.logDebug("Error while updating process execution status...+ e");
}
finally {
    ConnectionAdapter.closeResultSet(rs);
    ConnectionAdapter.closePreparedStatement(prepStatement);
    ConnectionAdapter.closeConnection(configConn);
}
return false;
}

```

17.1.4 Connection to Atomic Schema

Opening Connection: To open an atomic connection the getDBConnections method of ConnectionAdapter has to be invoked with infodom and isMetaConnection as parameters.

```
Connection atomicConn = ConnectionAdapter.getDBConnections(infdom, false);
```

For example:

```

public boolean testMethod(String attr1) {
    Connection atomicConn = null;
    PreparedStatementDecorator prepStatement = null;
    ResultSet rs = null;
    try {
        atomicConn = ConnectionAdapter.getDBConnections(infdom, false);
        prepStatement = new PreparedStatementDecorator(configConn, query);
        prepStatement.setString(1, attr1);
        rs = prepStatement.executeQuery();
        while (rs.next())

```

```

    {
        return true;
    }

}

catch (Exception e)
{
    WorkflowUtil.logDebug("Error while updating process execution status...+ e");
}

finally {
    ConnectionAdapter.closeResultSet(rs);
    ConnectionAdapter.closePreparedStatement(preparedStatement);
    ConnectionAdapter.closeConnection(configConn);
}

return false;
}

```

17.2 Logging API

For logging into an application, the WorkflowUtil class provided by PMF can be used.

17.2.1 Jar Files Required

The following jar available at `$FIC_HOME/ficweb/webroot/WEB-INF/lib` folder contains the WorkflowUtil class that contains Logging related API's.

`aai-pmf-common.jar`

Referenced File in Jar:

The `aai-pmf-common.jar` provides the following class that can be used to implement Logging.

- `WorkflowUtil`

17.2.2 Debug Message

Debug messages can be used to log information that is required for debugging.

Signature

```
public static void logDebug(String logMessage)
```

Examples:

```
WorkflowUtil.logDebug( "Your Message");
WorkflowUtil.logDebug( "Message" + variableName);
```

17.2.3 Error Message

An error message can be used to log an exception.

Signature:

```
public static void logError(Exception e)
```

For example:

```
WorkflowUtil.LogError(e);
```

OFSA Support

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

Send Us Your Comments

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

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

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

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

ORACLE