Oracle Financial Services Analytical Applications Infrastructure

Process Modelling Framework Orchestration Guide

Release 8.0.8.0.0

August 2022

ORACLE
FINANCIAL SERVICES

ORACLE

OFSAAI Process Modelling Framework Orchestration Guide

Document Control

Version Number	Revision Date	Change Log
2.3	August 2022	Inline process is not supported from versions 8.0.8.5.0 onwards. All references to Inline process are removed.
2.2	July 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).
2.1	June 2020	Added note in the Operations Module section for Object ID field (Doc 31460787).
2.0	Modified Nov 2019	Updated the following sections: • Configuring Application Object Model (AOM) • Executing Run Pipeline using Command Line Utility • Rest Service Application Rule
1.0	Created May 2019	Updated the following sections for the enhancements done in 8.0.8.0.0 ML: • Configuring Application Object Model (AOM) • Executing Run Pipeline • Abort Run Pipeline • Resume Run Pipeline • Re-run Run Pipeline • Added a new Appendix for Support APIS for Java External APIs. • Added new section Application Registration.

Table of Contents

1 Introduction

Process Modeling 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.

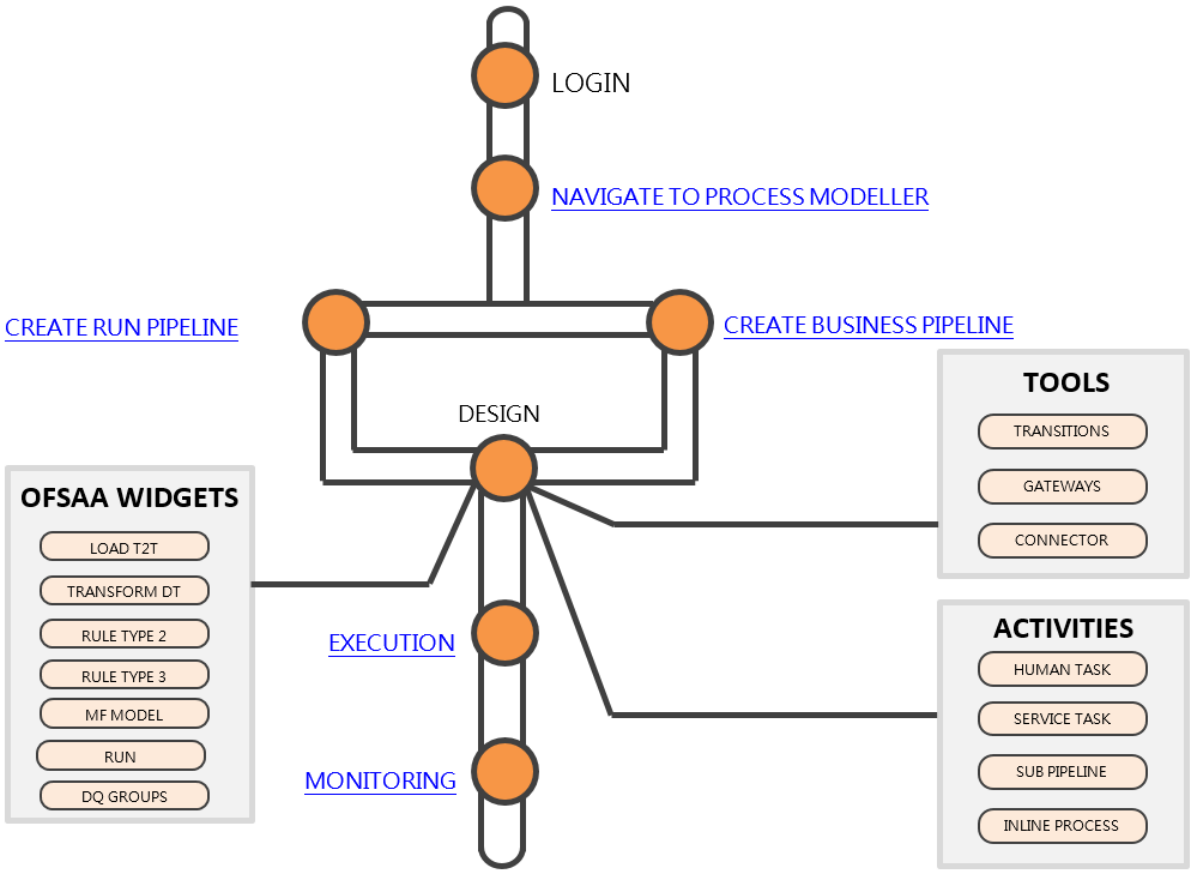
Process Modeller aids in representing the various artifacts required for modeling and provides implementation details of the OFSAA process artifacts.

- **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 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.

1.2 Process Pipeline Flow



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

1.2.1 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:

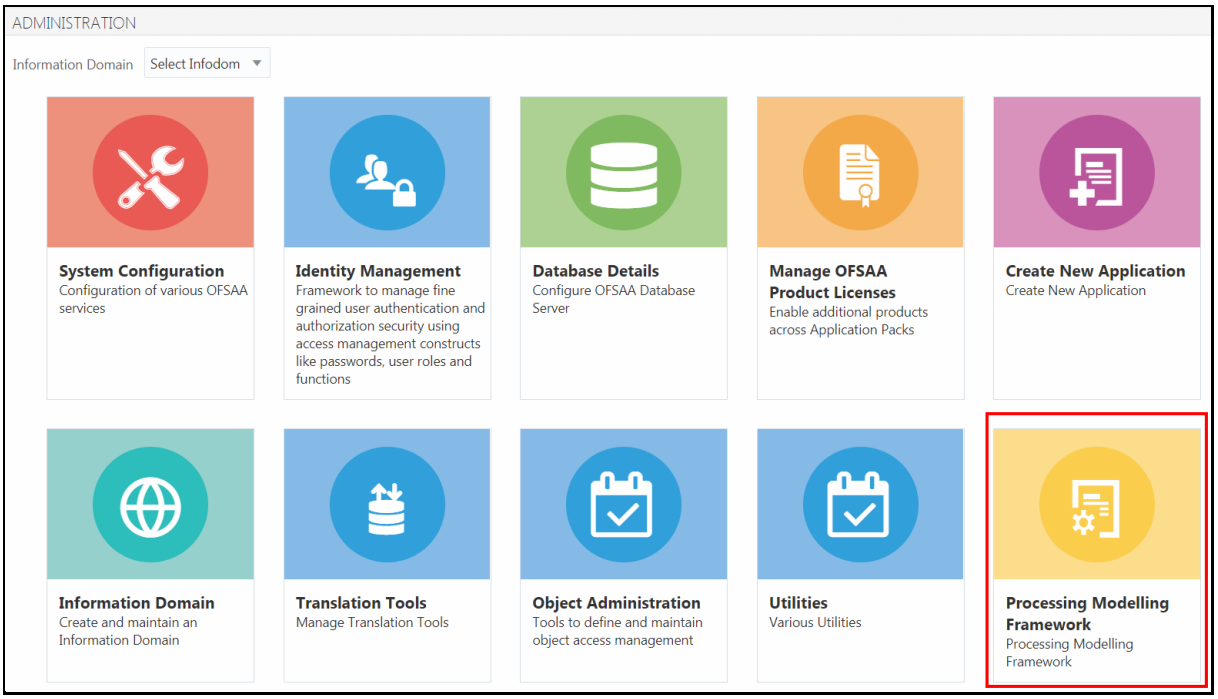
Role Code	Role Name	Functionality
WFACC	Workflow Access	Assign this role to the user to access the Process Modeller menu from the Navigation Tree. **Note:** The mapping of this role does not allow view, edit, and add actions.

Role Code	Role Name	Functionality
WFMACC	Workflow Monitor Access	Assign this role to the user to access the Process Monitor window. **Note:** The mapping of this role does not allow view, edit, and add actions.
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.

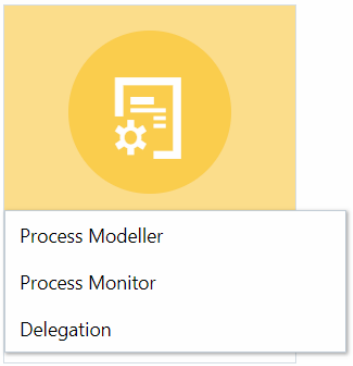
NOTE For administrators, ensure that they are mapped to all the roles described in the preceding table to allow them to perform all types of operations in PMF.

1.3 Accessing Process Modelling Framework

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

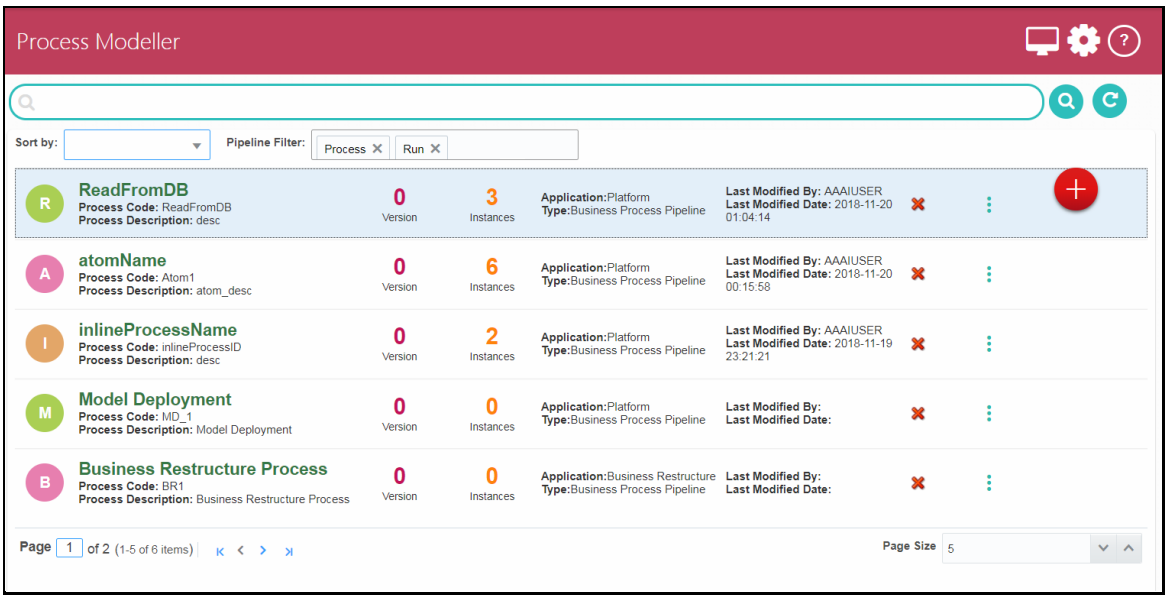


- Select the **Information Domain** from the drop-down list.
- Click **Process Modelling Framework** tile to display the sub-menu.



- Click **Process Modeller** to launch the Process Modeller.
- Click **Process Monitor** to monitor currently running processes.
- Click **Delegation** to launch the Delegation framework.

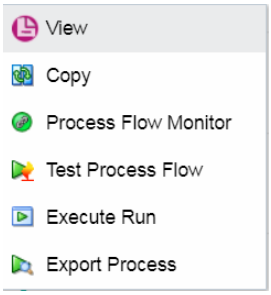
2 Process Modeller



The *Process Modeller* window displays the existing Business Process Pipelines and Run Pipelines with the details such as Process ID, Process Name, Process Description, Version, Instance, Application, and Last Modified 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 Process Flow.
- Click  to delete a Pipeline.
- Click to view the following sub-menu:



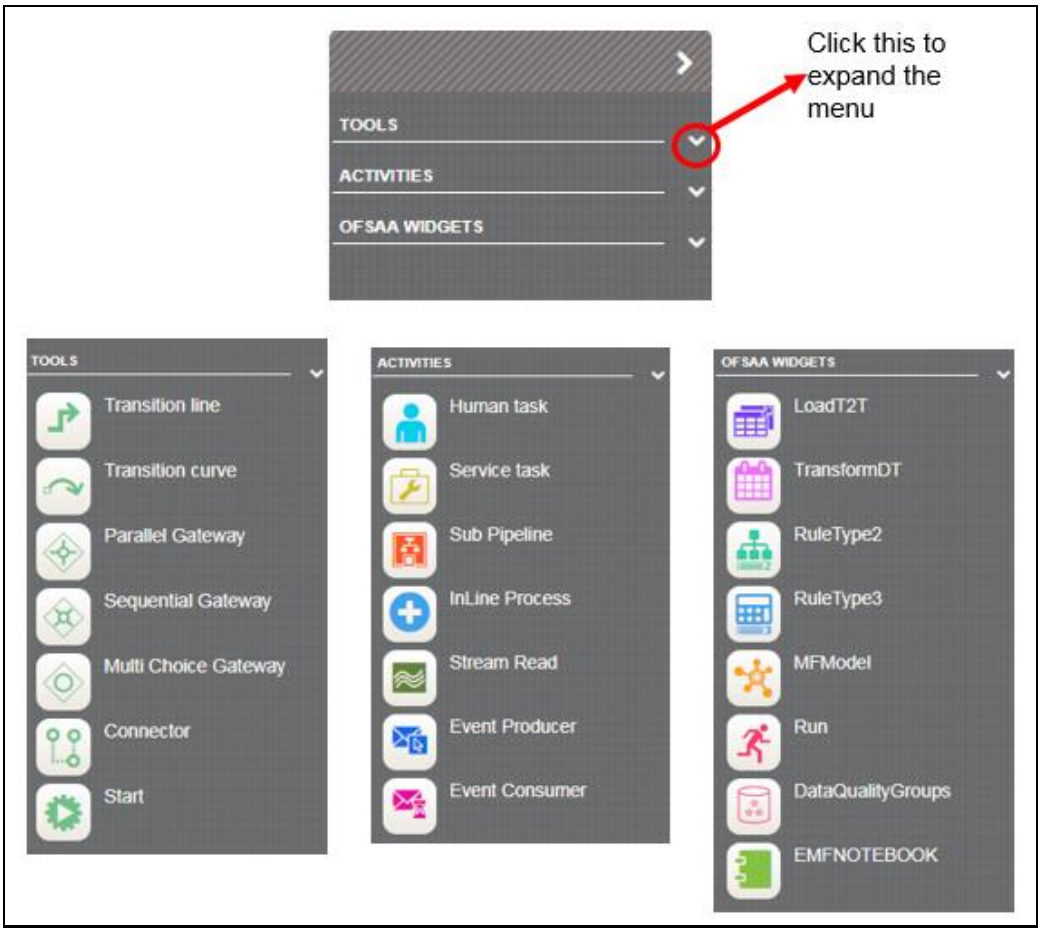
  - Click **View** to see the process flow.
  - Click **Copy** to create a new Pipeline with the same process flow.
  - Click **Process Flow Monitor** to monitor the Pipeline.
  - Click **Test Process Flow** to test/check whether the Business Pipeline you designed works as expected.
  - Click **Execute Run** to execute a Run Pipeline.

  - Click **Export Process** to export a Process.

- 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 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.



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. Gateway can be Sequential, Parallel, or Multi Choice.

- **Parallel Gateway**: A Parallel gateway is used when you want to have multiple transitions/flows which should be executed in parallel.
- **Sequential Gateway**: A Sequential gateway is used when you want to have multiple transitions/flows which 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.

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, 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 is required to get introduced to some artifacts of the Process Modelling Framework.

4.1 Application Package

Application Package is a concept used to group Pipelines, Application Rules, and Data Fields which 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.

V_APP_PACKAGE_ID	V_APP_PACKAGE_DESC	V_LANGIN	V_DEFINITION_PAGE_URL	V_IS_EMAIL_REQUIREI
0	Global			N
10	Business Restructure		Restructure/manage_grid.jsp?userId={ASSIGNEEUSERS}&loca	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 which 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

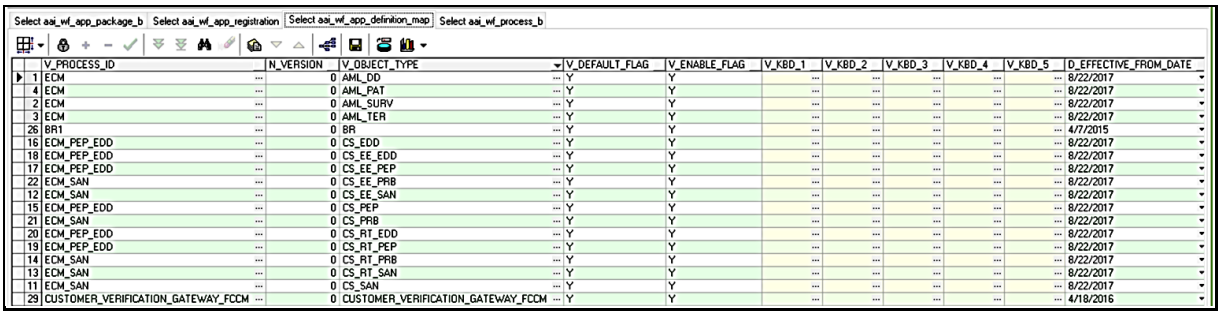
The AAI_WF_APP_REGISTRATION table stores information like the Object Type, Fact Table, Primary Key Column Name, Object Name, Object Type Name and so on for an Application Package.

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	OFS_NGECM	FR_EE	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
2	OFS_NGECM	AML_PAT	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
3	OFS_NGECM	AML_SURV	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
4	OFS_NGECM	FR_AC	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
5	OFS_NGECM	AML_TER	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
6	OFS_NGECM	AML_DD	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
7	OFS_NGECM	FR_ON	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
8	OFS_NGECM	KYC_FIRM	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
9	OFS_NGECM	KYC_IND	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
10	OFS_NGECM	KYC_CORP	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
11	OFS_NGECM	CS_SAN	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
12	OFS_NGECM	CS_EE_SAN	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
13	OFS_NGECM	CS_RT_SAN	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
14	OFS_NGECM	CS_RT_PRB	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
15	OFS_NGECM	CS_PEP	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
16	OFS_NGECM	CS_EDD	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
17	OFS_NGECM	CS_EE_PEP	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
18	OFS_NGECM	CS_EE_EDD	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
19	OFS_NGECM	CS_RT_PEP	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
20	OFS_NGECM	CS_RT_EDD	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
21	OFS_NGECM	CS_PRB	kdd_cases	CASE_INTRL_ID	CASE_INTRL_ID	Case Management
22	OFS_NGECM	CS_EE_PRB	kdd_cases	CASE_INTRL_ID	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.



Select aai_wf_app_package_b | Select aai_wf_app_registration | Select aai_wf_app_definition_map | Select aai_wf_process_b

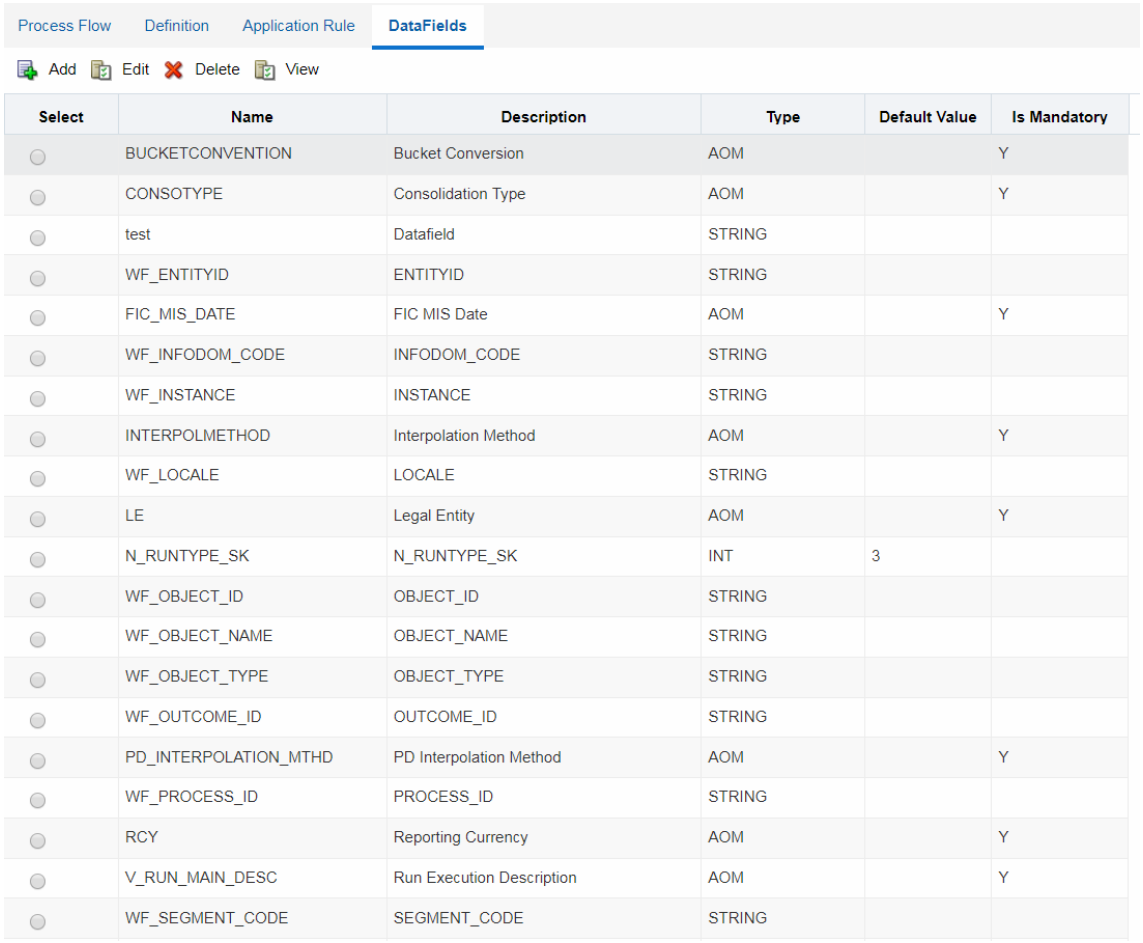
	V_PROCESS_ID	N_VERSION	V_OBJECT_TYPE	V_DEFAULT_FLAG	V_ENABLE_FLAG	V_KBD_1	V_KBD_2	V_KBD_3	V_KBD_4	V_KBD_5	D_EFFECTIVE_FROM_DATE
1	ECM	0	AML_DD	Y	Y						8/22/2017
4	ECM	0	AML_PAT	Y	Y						8/22/2017
2	ECM	0	AML_SURV	Y	Y						8/22/2017
3	ECM	0	AML_TER	Y	Y						8/22/2017
26	BR1	0	BR	Y	Y						4/7/2015
16	ECM_PEP_EDD	0	CS_EDD	Y	Y						8/22/2017
18	ECM_PEP_EDD	0	CS_EE_EDD	Y	Y						8/22/2017
17	ECM_PEP_EDD	0	CS_EE_PEP	Y	Y						8/22/2017
22	ECM_SAN	0	CS_EE_PRB	Y	Y						8/22/2017
12	ECM_SAN	0	CS_EE_SAN	Y	Y						8/22/2017
15	ECM_PEP_EDD	0	CS_PEP	Y	Y						8/22/2017
21	ECM_SAN	0	CS_PRB	Y	Y						8/22/2017
20	ECM_PEP_EDD	0	CS_RT_EDD	Y	Y						8/22/2017
19	ECM_PEP_EDD	0	CS_RT_PEP	Y	Y						8/22/2017
14	ECM_SAN	0	CS_RT_PRB	Y	Y						8/22/2017
13	ECM_SAN	0	CS_RT_SAN	Y	Y						8/22/2017
11	ECM_SAN	0	CS_SAN	Y	Y						8/22/2017
29	CUSTOMER_VERIFICATION_GATEWAY_FCCM	0	CUSTOMER_VERIFICATION_GATEWAY_FCCM	Y	Y						4/18/2016

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.

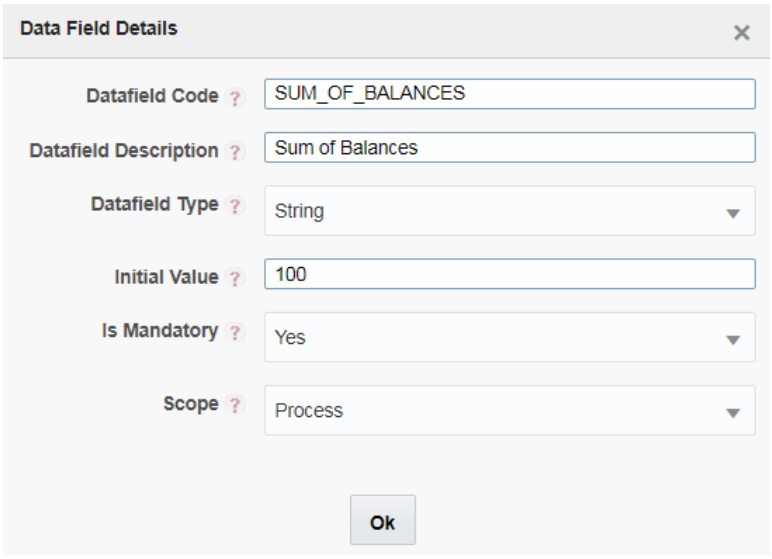
4.3.1 Adding a Data Field

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.
1. Select the *DataFields* tab.



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		

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



3. Enter the details as given in the table:

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: • 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.

4. 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, will be 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 these are categorized into three types.

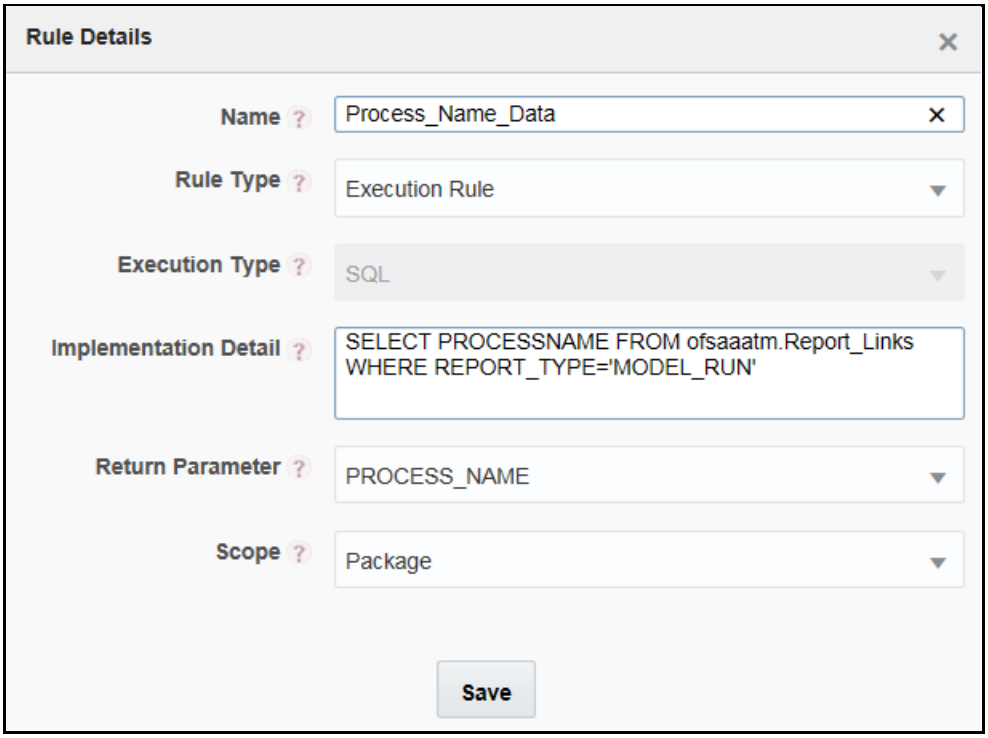
- **Execution Rule**: These are Business Logic executed as Task by an Activity.

- **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`

4.4.1 SQL Application Rule:

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

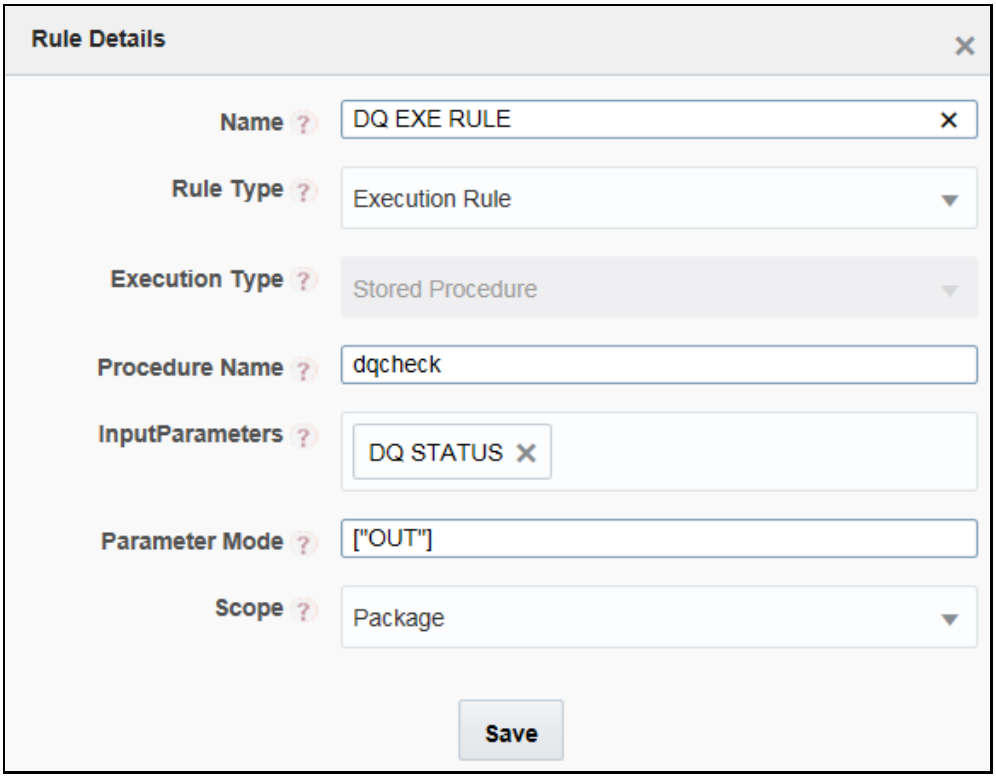


Field Name	Description
Name	Enter a unique name for the SQL Application Rule.
Rule Type	Select the **Rule Type** from the drop-down list. The SQL Application Rule can be used as Decision Rule, Execution Rule or Selection Rule based on your requirement.
Execution Type	Displays the Application Execution Type as SQL.

Field Name	Description
Implementation Detail	• **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	Select the Data Field which will receive the return parameter of the SQL Rule, from the drop-down list. • 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.

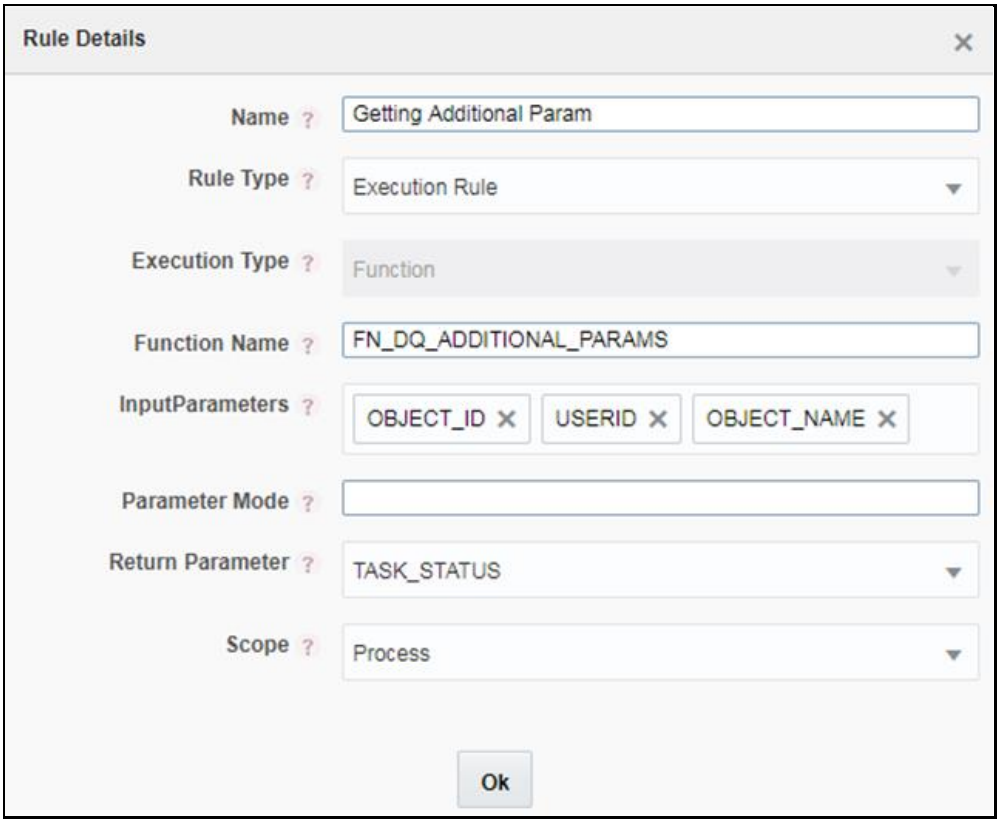


Field Name	Description
Name	Enter a unique name for the SQL Application Rule.
Rule Type	Select the **Rule Type** from the drop-down list. The Stored Procedure Application Rule can be used as a Decision Rule, Execution Rule or Selection Rule based on your requirement.
Execution Type	Displays the Application Execution Type as Stored Procedure.
Procedure Name	Enter the Stored Procedure Name.
Input Parameters	Select the list of Data Fields which will be passed as input parameters, from the drop-down list.
Parameter Mode	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"].During Execution of Stored Procedure, • 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.

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.

4.4.3 For Function Application Rule:

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

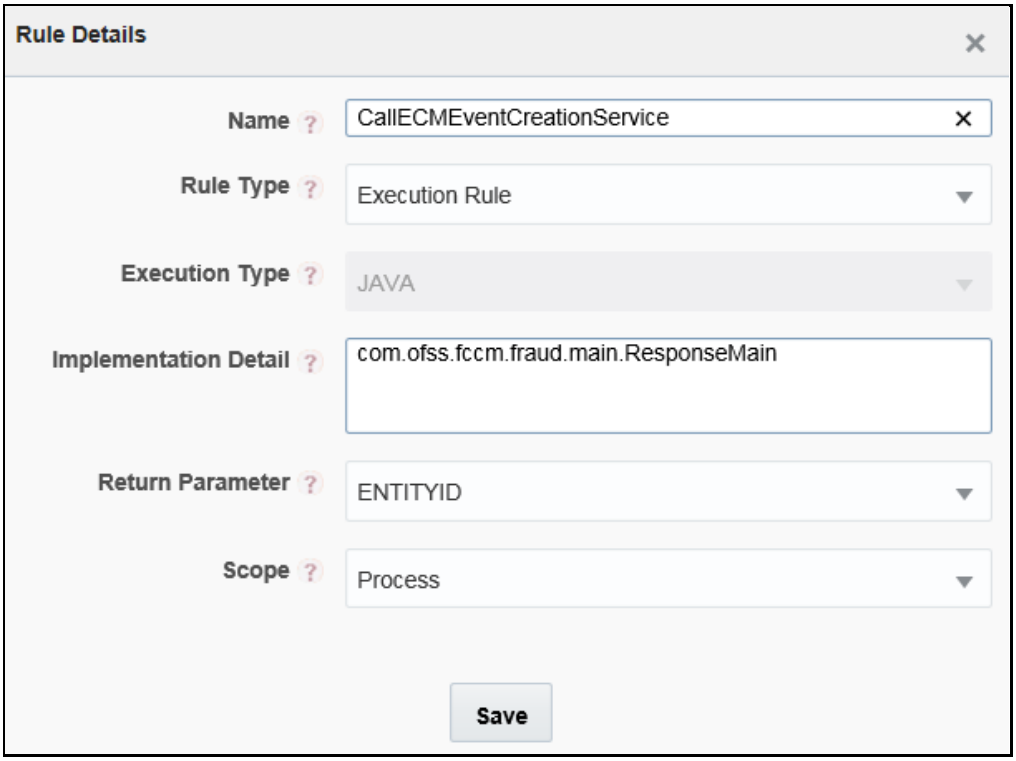


Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the rule type from the drop-down list. The available rule types are Decision Rule, Execution Rule and Selection Rule.
Execution Type	Displays the Application Execution Type as Function.
Function Name	Enter the Function Name.
Input Parameters	Select the list of Data Fields which will be passed as input parameters, from the drop-down list.

Field Name	Description
Parameter Mode	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"]. During Execution of Function, • 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	Select the Data Field which will receive the return parameter of the Java function, from the drop-down list. • 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 For Java Application Rule:

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



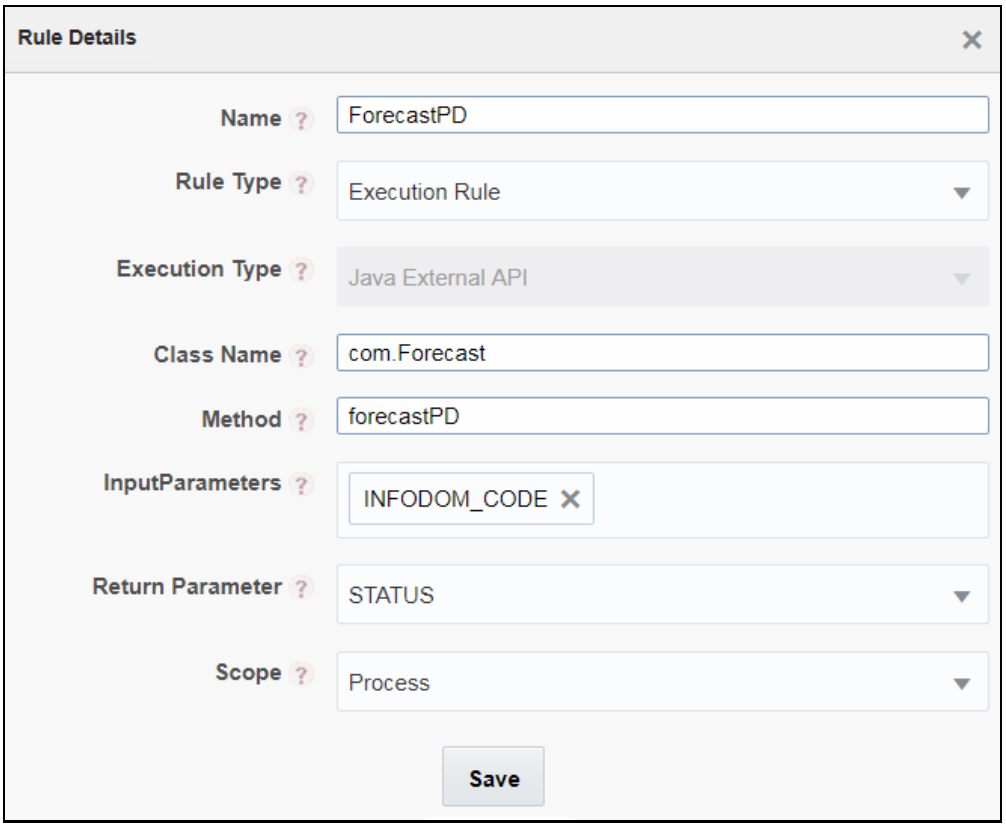
Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	This Rule Execution type supports only ExecutionRule type.
Execution Type	Displays the Application Execution Type as JAVA.
Implementation Detail	Enter the complete java class name which implements the Interface : com.ofs.aai.service.wf.external.base.Activity. The implementation class has to override the method with the Business Logic. executeTask(List<Data Field> Data Fields) 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.
Return Parameter	Select the Data Field which will receive the return parameter of the Java function, from the drop-down list. • 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”.

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 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 For 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.



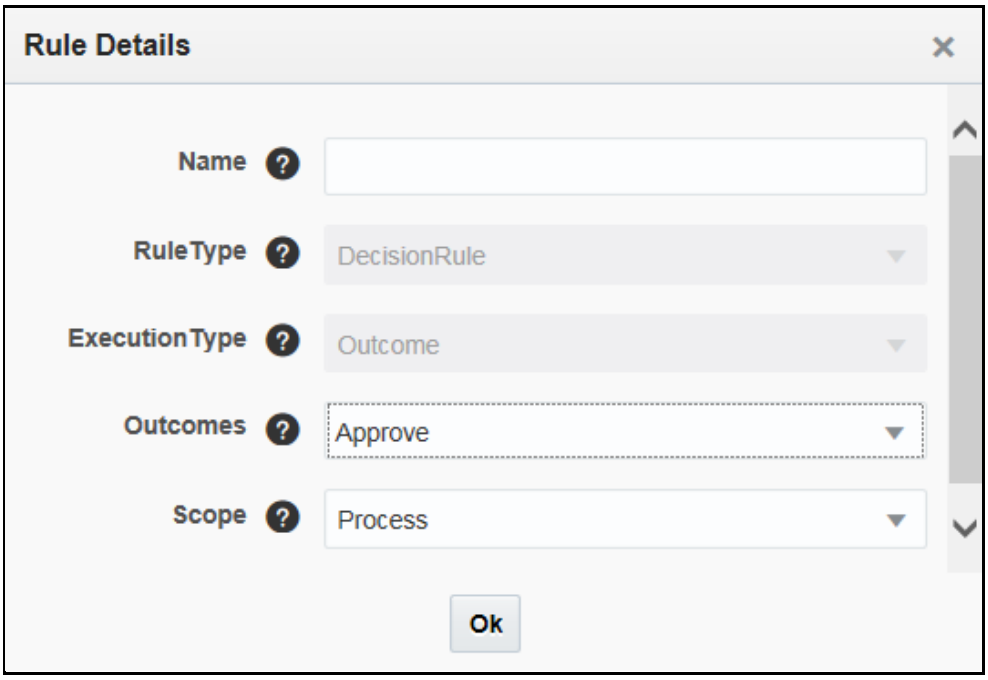
Field Name	Description
Name	Enter a unique name for the Application Rule.

Field Name	Description
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 which implements the Business Logic.
Method	Enter the method which 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 which will store the Return Parameter of the method, from the drop-down list.
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 API's for using as Java External APIs, see Appendix B.

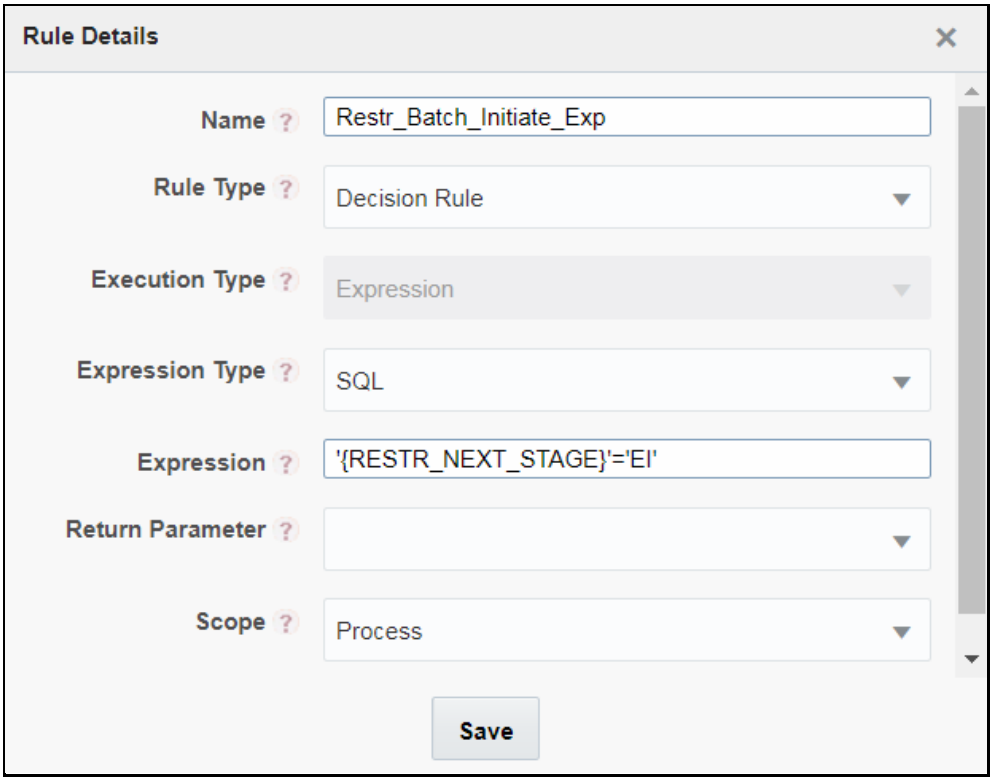
4.4.6 For Outcome Rules Application Rule



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 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'.

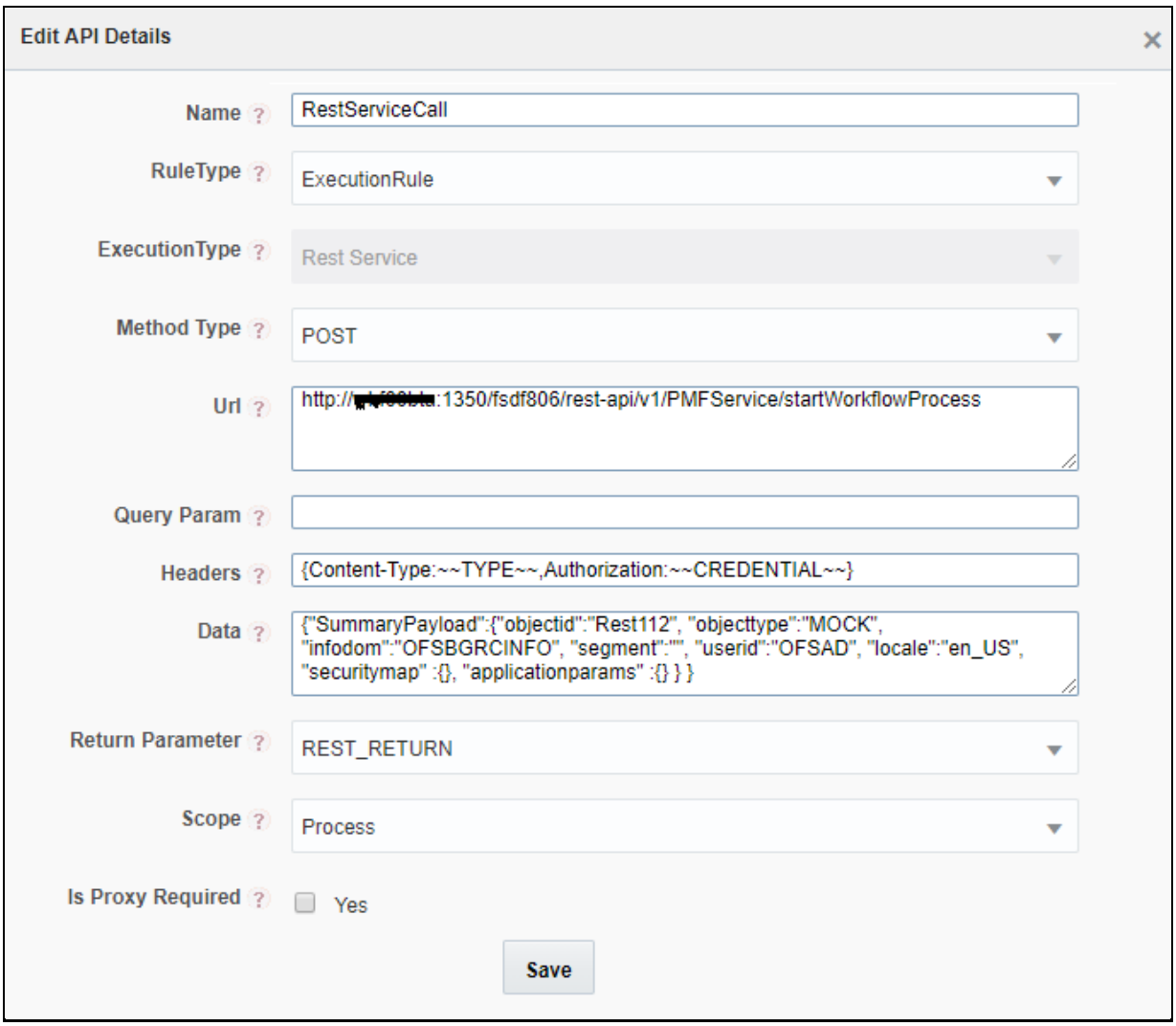


Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the **Rule Type** from the drop-down list. The Expression Application Rule can be used as Decision Rule, Execution Rule or Selection Rule based on your requirement.
Execution Type	Displays the Application Execution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

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.

4.4.8 Rest Service Application Rule

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



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

Field Name	Description
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).
Authorization Type	Select the authorization type from the drop-down list. The options are: • **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	This field is displayed only if you have selected **Basic Auth** as **Authorization Type**. • 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 section. • In case this field is left blank, logged in user credentials will be taken as authorization header. **Note**: If Authorization is given in the Header explicitly, then it will take preference than the value given in the **Authorization** field.
Query Param	Enter the **Query Parameters** that needs to be passed to the rest API. For example, http://example.com/foo?bar
Headers	Enter any headers that need to be passed to the rest API. For example, `"content-type": "application/json"` To pass the header values dynamically, use the following format: {Content-Type:~~TYPE~~,Authorization:~~CREDENTIAL~~} Where TYPE and CREDENTIAL are data fields.
Data	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. 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}`

Field Name	Description
Return Parameter	Select the Data Field which will receive the return parameter of the Expression, from the drop-down list. • For ExecutionRule type, the business logic is implemented in the method and the parameter value returned from the Application Rule is saved in the mapped Data Field. • In case of SelectionRule type, the Application Rule should be a String value. • In case of 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.
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.

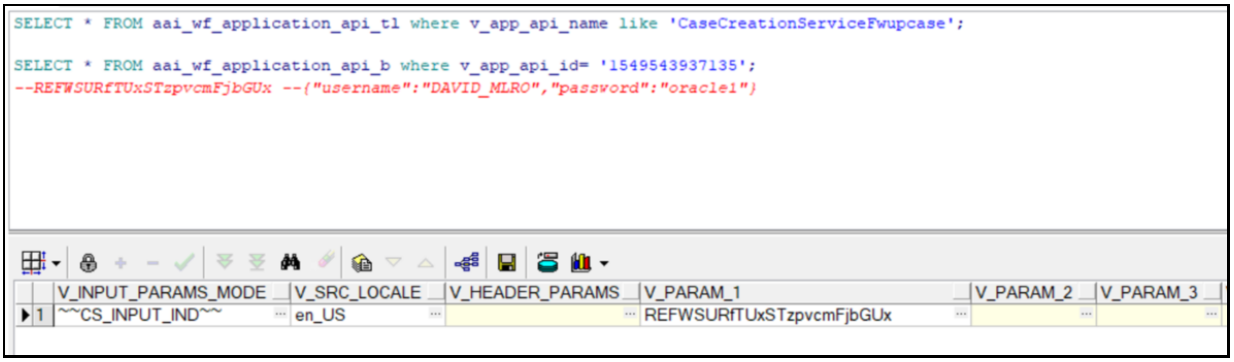
Add the following entries in the `AAI_WF_GLOBAL_SETTINGS` table:

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

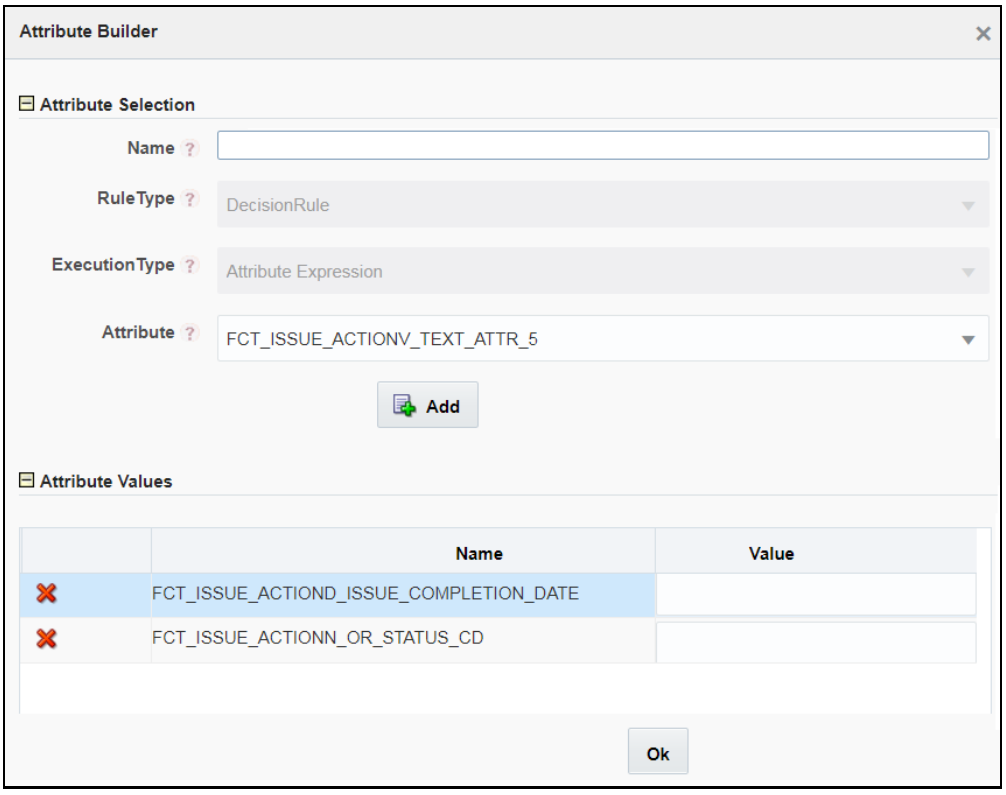
4.4.8.2 Authentication of Rest Service

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 will be converted to "REFWSURfTUxSTzpvcmFjbGUx".

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:



4.4.9 For Attribute Expression Application Rule



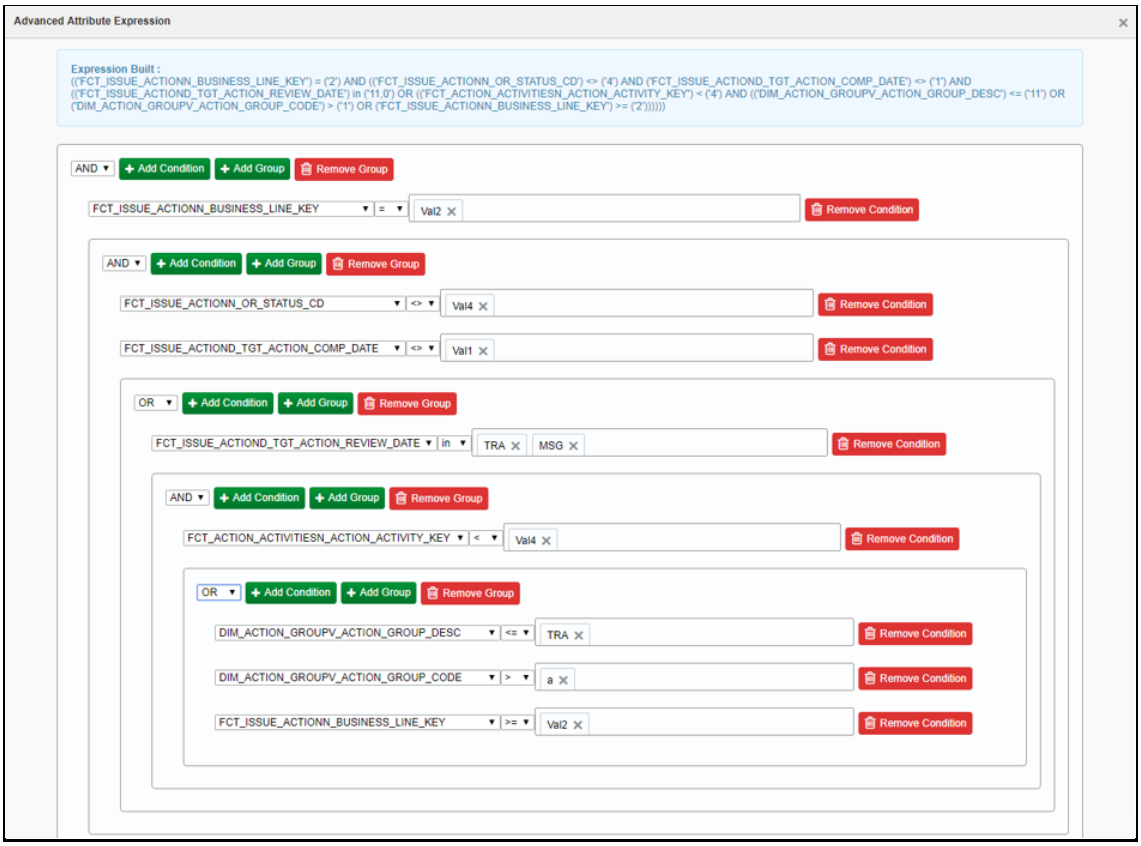
Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Displays the rule type as DecisionRule. This Rule Execution type supports only DecisionRule type.

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

4.4.10 For 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 will return True or False value after evaluation. This is used as a DecisionRule in transitions.

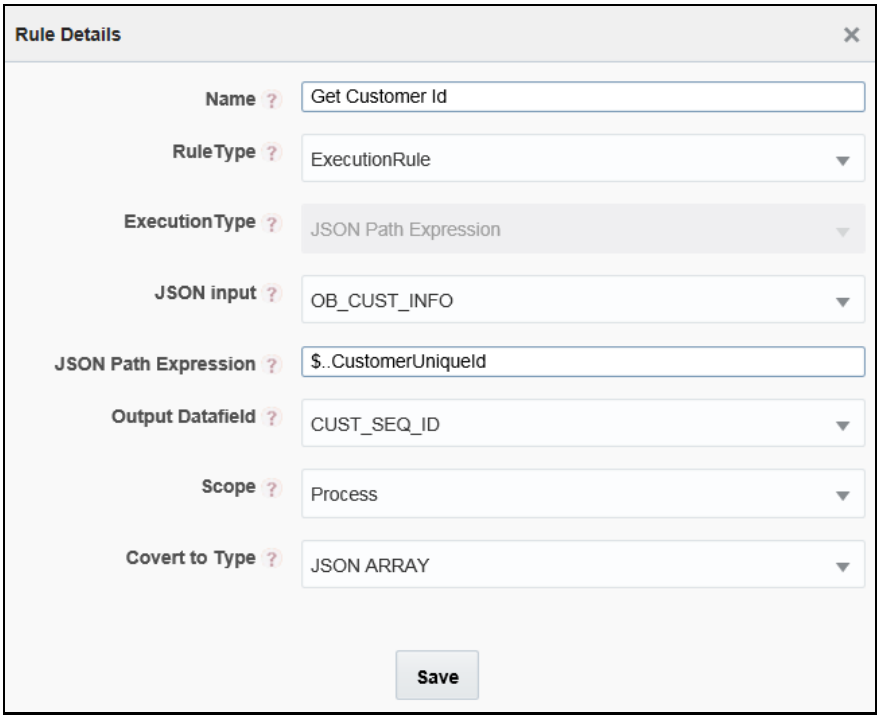
You can define this application rule with multiple conditions and nested groups.



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	When you click **Add Condition**, a new row gets added. To define a condition, select the attribute, the operator and the value from the drop-down lists. Multiple values can be selected for each attribute. • Attribute- The drop-down list displays the attributes configured for the selected application and component. 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. Click **Remove Condition** to delete an already added condition.
Add Group	Click **Add Group** if you want to have nested conditions. For each group, select the required logical operator as AND or OR. Click **Remove Group** to delete a group of conditions.

4.4.11 For JSON Path Expression Application Rule

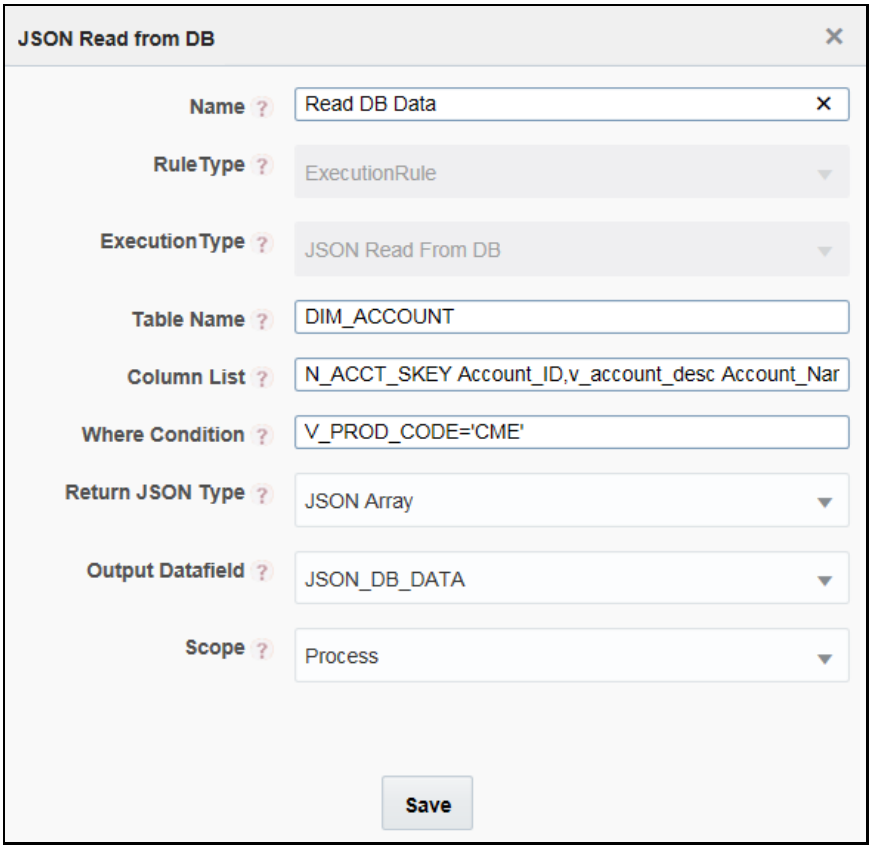
This Application rule is used to extract data from the JSON Path Expression, which gets returned from a Rest API or Web Service call, and you can store it into a Data Field for further processing.



Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Select the rule type from the drop-down list. The available rule types are Decision Rule and Execution Rule. For Decision Rule, the output of JSON Path Expression will be compared with RHS expression and rule will return either true or false accordingly. For execution rule, JSON Path Expression is evaluated and the output is returned to the DataField selected as Output DataField.
Execution Type	Displays the Application Execution Type as JSON Path Expression.
JSON Input	Select the Data Field in which the output of Web Service is stored from the drop-down list. You should select a DataField which has JSON as its value.
JSON Path Expression	Enter the JSON path expression. 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 <=.
RHS Expression	This field is displayed only if Rule Type is selected as Decision Rule. Enter the expression to which you want to compare the JSON path expression.
Output DataField	This field is displayed only if Rule Type is selected as Execution Rule. Select the DataField to which you want to return the value of JSON Path Expression, from the drop-down list.
Scope	Select the scope of the Application Rule from the drop-down list. The options are: • Process- Select Process if you want to use the Application Rule only in the current process. • Package- Select Package if you want to use the Application Rule across all the processes in the Application package.
Convert To Type	Select JSON ARRAY to store the output in Array format or select String to store as a string, from the drop-down list.

4.4.12 JSON Read From DB Application Rule

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

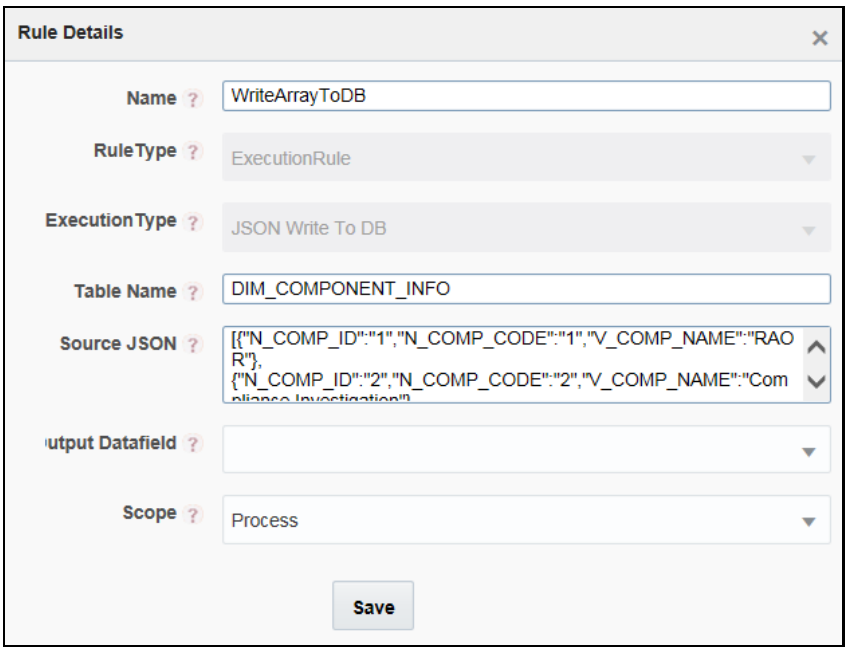


Field Name	Description
Name	Enter a unique name for the Application Rule.
Rule Type	Only the Execution Rule type is supported.
Execution Type	Displays the Execution Type as JSON Read From DB.
Table Name	Enter the table name from which you want to read the data.
Column List	Enter the column names of the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

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

4.4.13 JSON Write To DB Application Rule

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



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.

Field Name	Description
Output DataField	This field is not applicable.
Scope	Select the scope of the Application Rule from the drop-down list. The options are: • Process- Select Process if you want to use the Application Rule only in the current process. • Package- Select Package if you want to use the Application Rule across all the processes in the Application package.

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

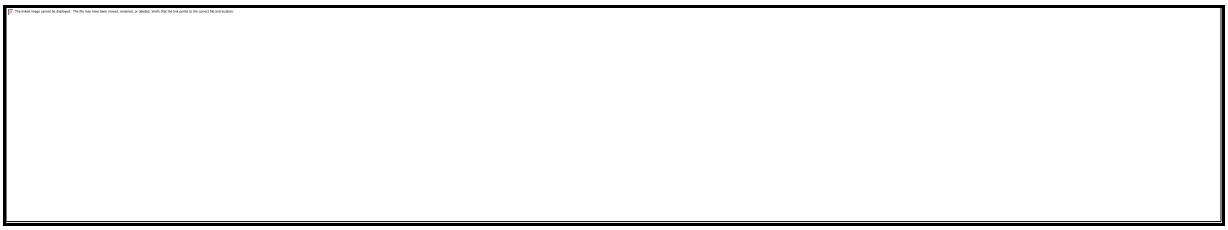
In this table, make entries for each attribute as given in the following table:

Column Name	Description
APP_COMP_ATTR_MAP_ID	Enter a unique ID for the attribute. 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.
V_ATTR_CODE	Name of the attribute.
N_ATTR_TYPE_ID	The ID of the attribute type. The values of the attributes are fetched based on the attribute type. 1001- Static 1002- SQL Query 1003- JavaAPI 1004- Hierarchy 1005- Multi Select Hierarchy 103- Date field 102- Text box field For more information, see Attribute Types.

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

Example for Run Pipeline

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



4.5.2 Attribute Types

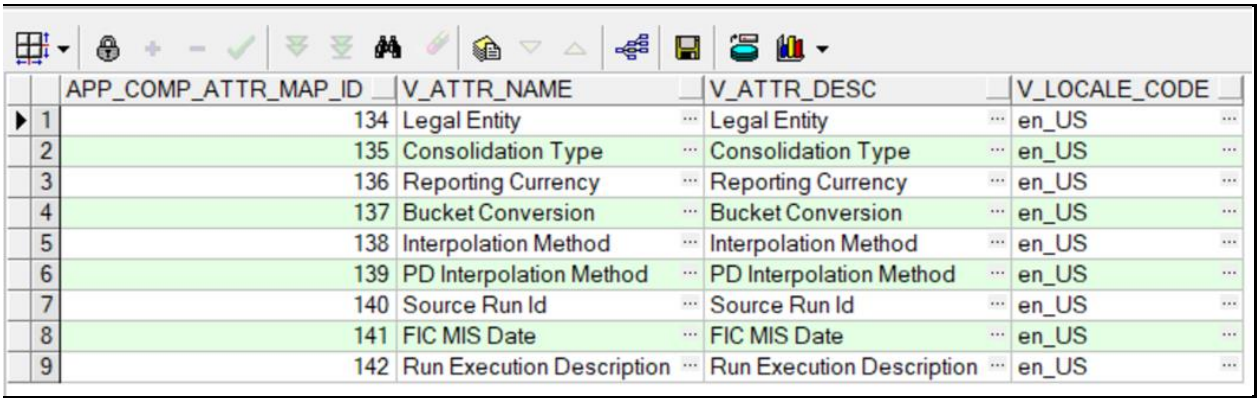
The values of attributes are fetched based on the attribute types. Following are the attribute types with their IDs:

- **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.

4.5.3 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:

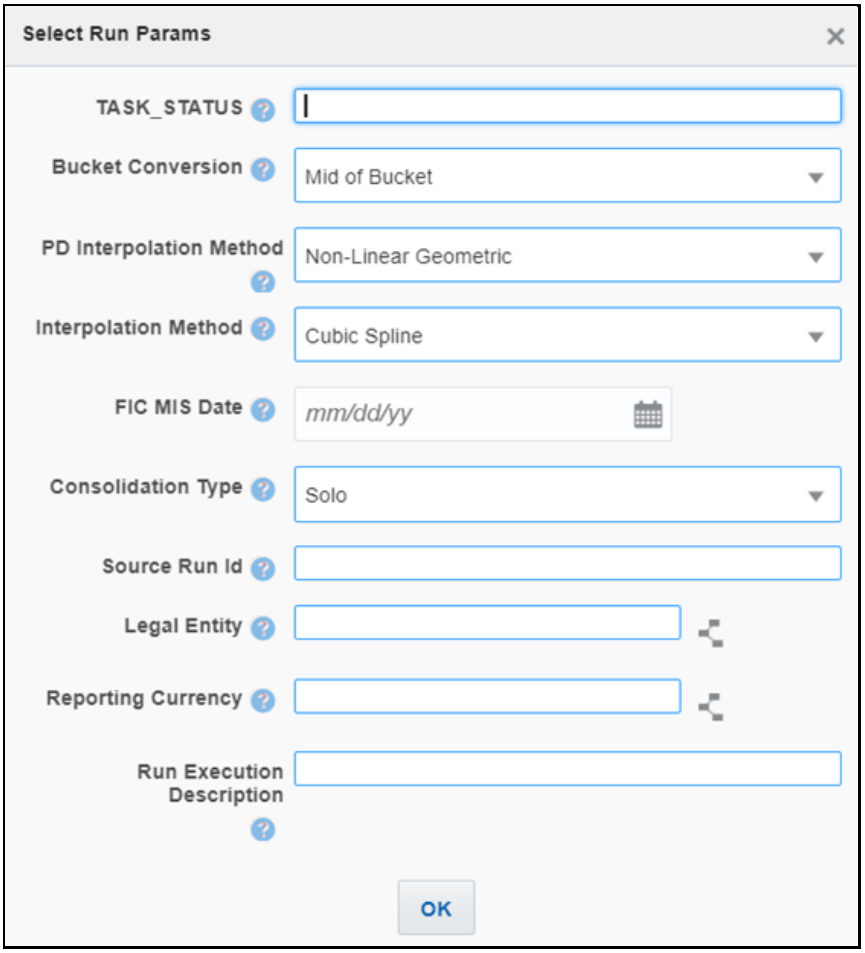


	APP_COMP_ATTR_MAP_ID	V_ATTR_NAME	V_ATTR_DESC	V_LOCALE_CODE
1	134	Legal Entity	Legal Entity	en_US
2	135	Consolidation Type	Consolidation Type	en_US
3	136	Reporting Currency	Reporting Currency	en_US
4	137	Bucket Conversion	Bucket Conversion	en_US
5	138	Interpolation Method	Interpolation Method	en_US
6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

4.5.4 Usage of AOM Attributes in Run Pipeline

While executing Run pipeline, the Select Run Params window displays the AOM fields which 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 will be displayed as shown:


Select Run Params
TASK_STATUS
Bucket Conversion
Mid of Bucket
PD Interpolation Method
Non-Linear Geometric
Interpolation Method
Cubic Spline
FIC MIS Date
mm/dd/yy
Consolidation Type
Solo
Source Run Id
Legal Entity
Reporting Currency
Run Execution Description
OK

5 Designing a Pipeline

Business pipelines are defined in OFSAA to design and execute the sequence of tasks which 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:

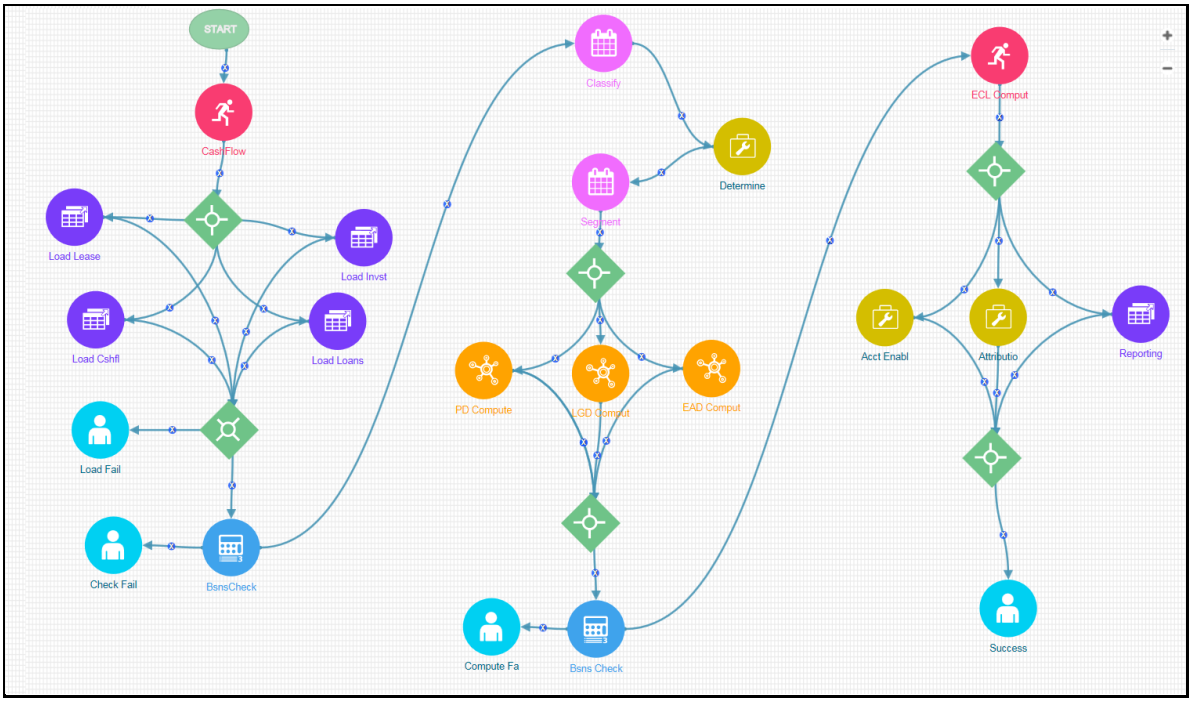
a) Orchestrate a Business pipeline

b) Orchestrate a Run pipeline using PMF modeling

5.1 Orchestration of 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.

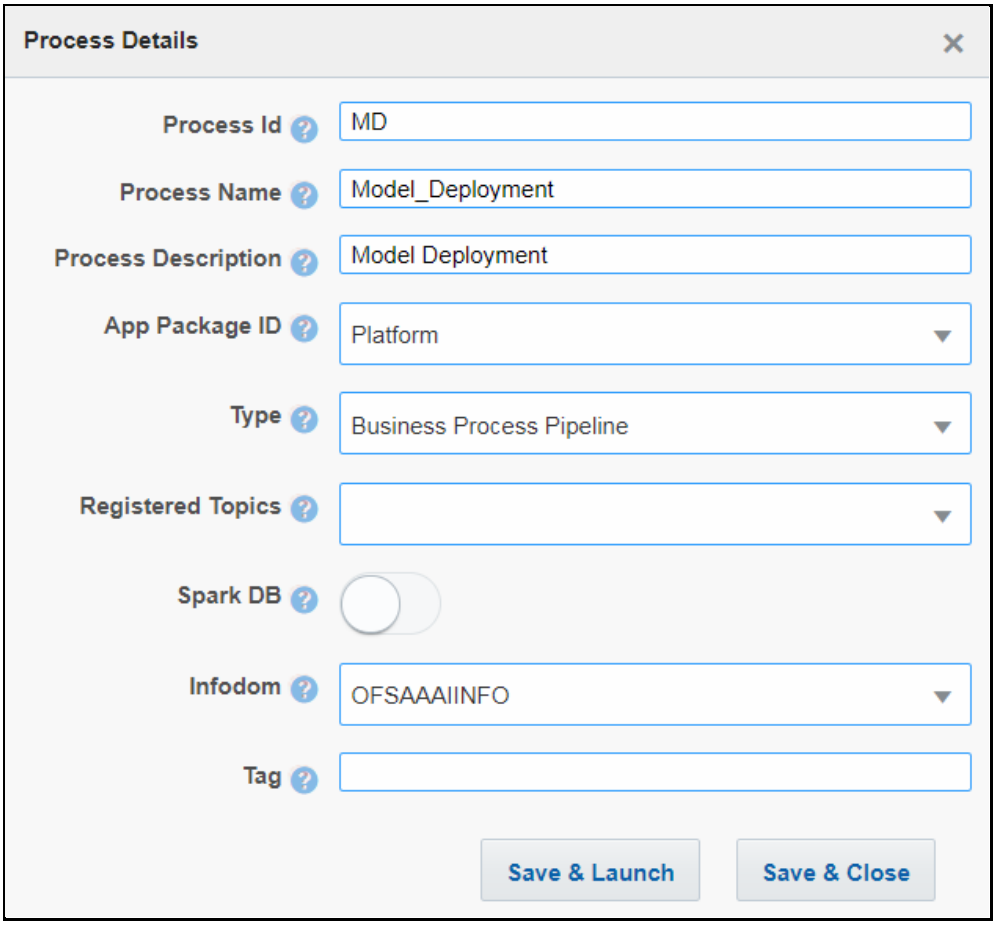
5.1.1 An Example of a Business Pipeline



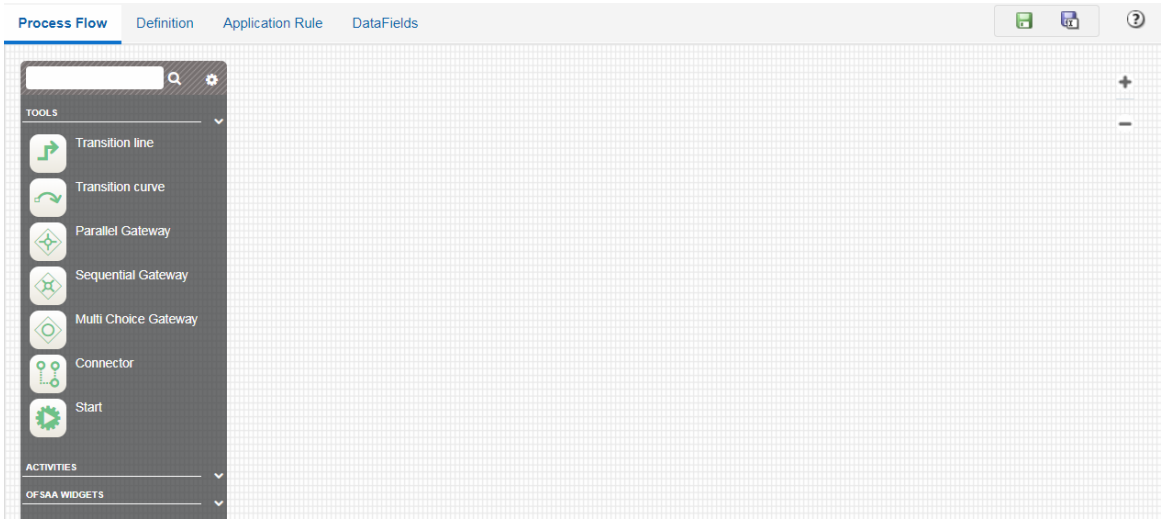
In the example shown, we use various OFSAA widgets like Run, LoadT2T, RuleType3, TransformDT, MFModel, and activities like Human Tasks and Service tasks, which are related to each other through transitions. For executing tasks in parallel, we used Parallel Gateways.

5.1.2 Creating a Business Pipeline

Click  in the *Process Modeller* Summary window.



5. Enter a unique Process ID, Process Name and a description.
6. Select the appropriate app package in which you want to create the process. For more information, see the Application Package section.
7. Select **Process Modelling** from the **Type** drop-down list.
8. 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 mapped to the applications configured in your OFSAA instance.
9. Click **Save & Launch.** The Process Flow canvas is displayed.



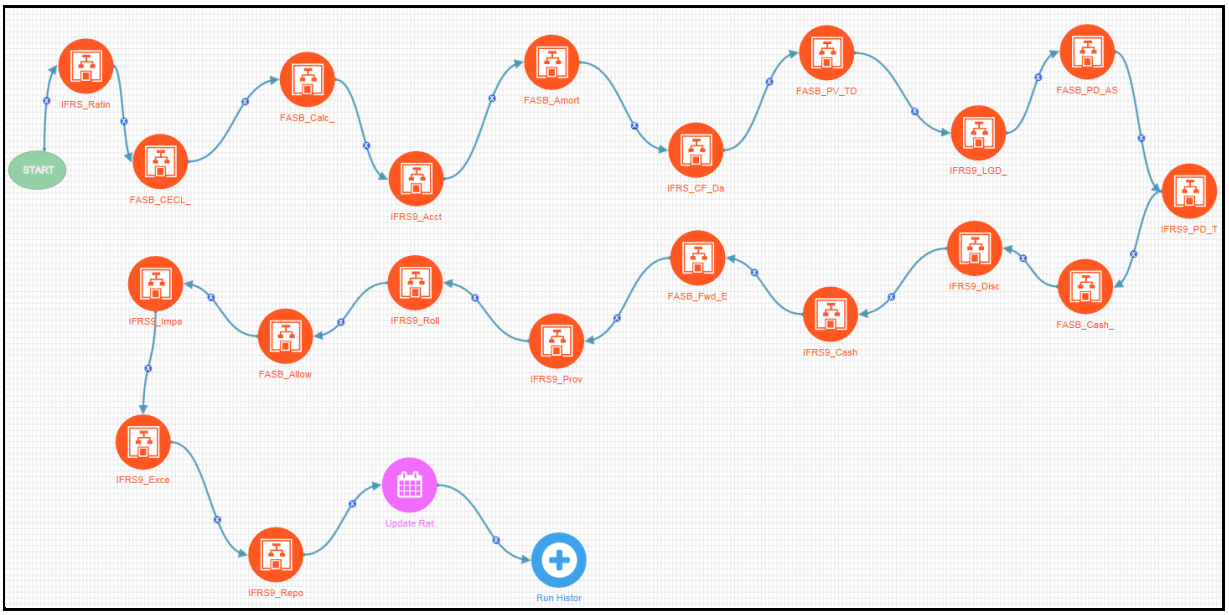
10. Click  **Start** from the *Tools* pane in the floating toolbar and click on the drawing canvas where you want to place it. This Start activity indicates the beginning of the Process.
11. Design your Process with various components available in the *Process Flow* tab. For more information on each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

NOTE

5.2 Run Pipeline

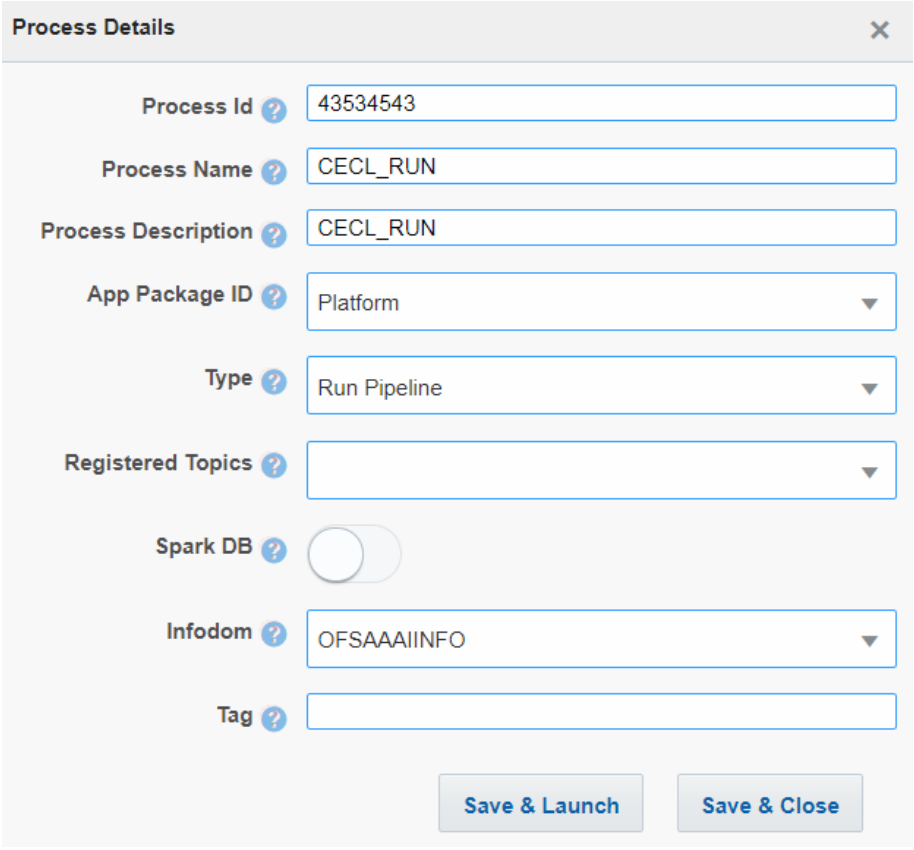
Run Process is used to create a Run definition in Rule Run Framework (RRF) using PMF Process. Visual representation of the Run is enabled through PMF by the construction of a Run Pipeline. Various OFSAA widgets that enable the construction of a Run Pipeline are available in the Component toolbar.

5.2.1 An Example of Run Pipeline



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.



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** 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 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.

Design your Run Pipeline using OFSAA Widgets

You can construct a Run pipeline using only OFSAA Widgets. For details, see the Configuring OFSAA Tasks in Your Process Flow section.

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.

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

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

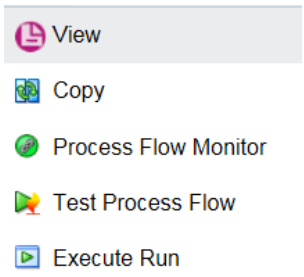
5.2.3 Executing Run Pipeline

You can execute a Run Pipeline from UI or using a command line utility called wfExecExternal.sh.

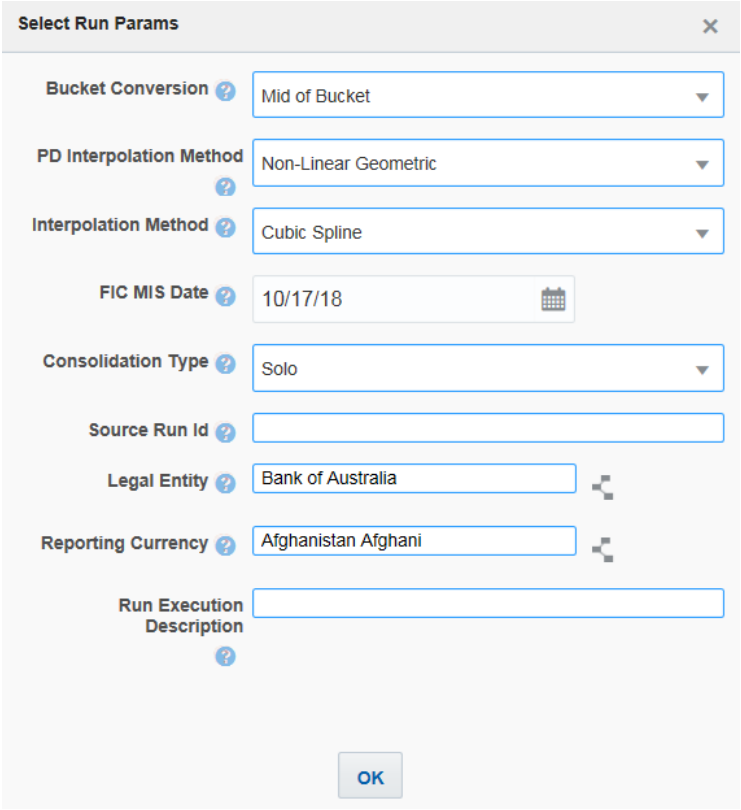
From UI

To execute Run Pipeline from UI:

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



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



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.

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` -Specify a unique Object ID. This is a mandatory parameter.
   - `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 1542221676429 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
```

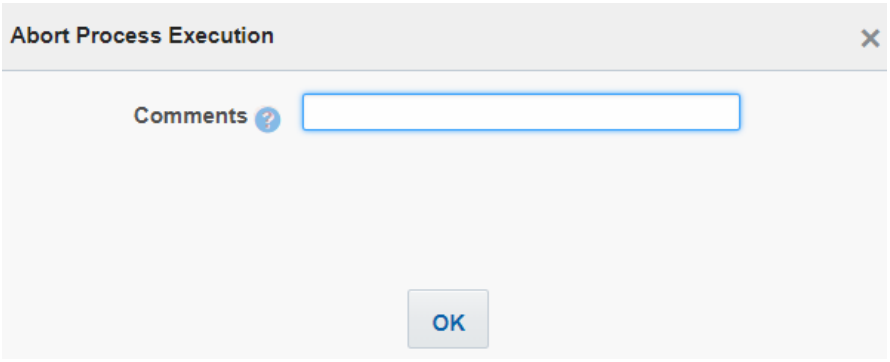
> **TIP** Null should be passed if you do not want to pass value for a parameter.

5.2.4 Abort Run Pipeline

The Abort feature facilitates you to abort a Run Pipeline which 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**.



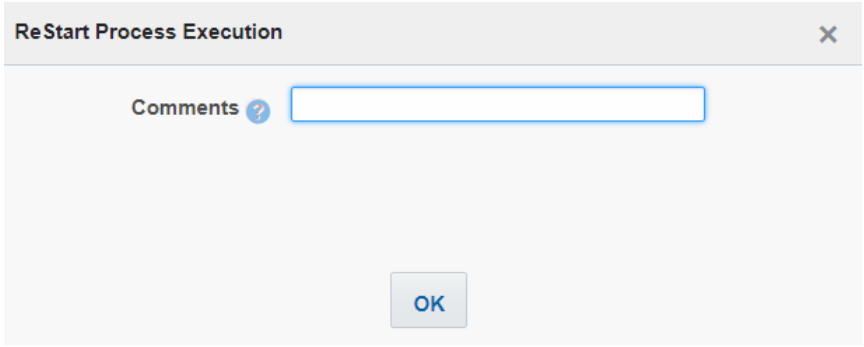
2. Enter comments if any, for aborting the Run Pipeline execution.

5.2.5 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**.



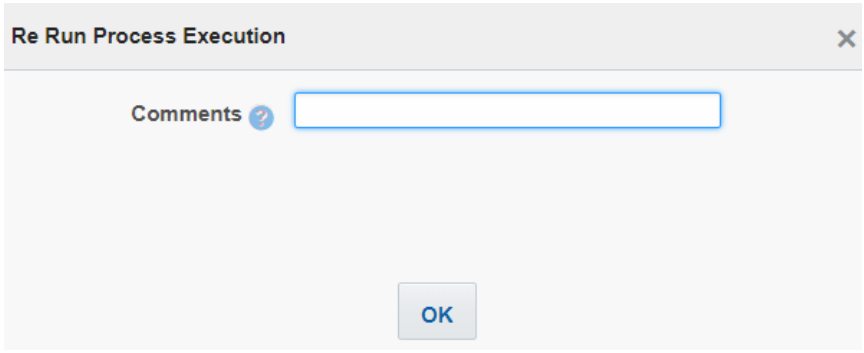
2. Enter comments if any, for restarting the Run Pipeline execution.

5.2.6 Re-run Run Pipeline

You can re-run a Run Pipeline which has previously been executed, irrespective of the previous execution state.

To re-rum a Run Pipeline

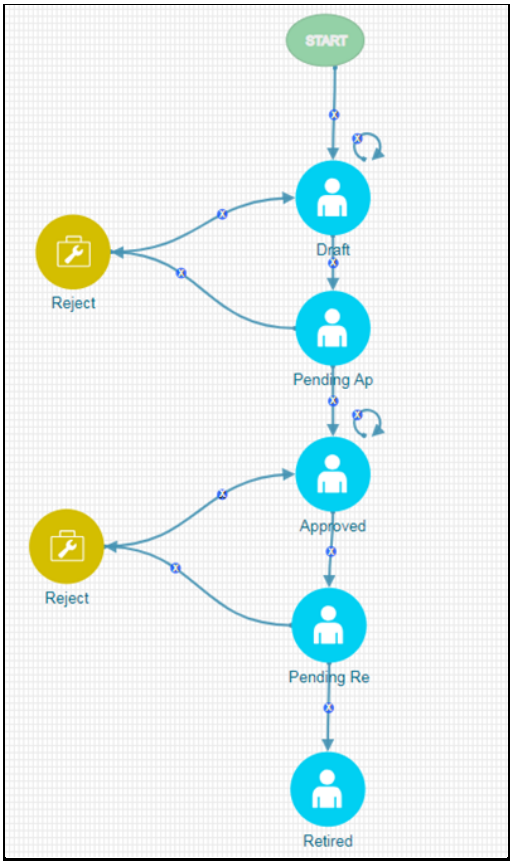
1. From the *Process Monitor* window, click ⋮ corresponding to the Run Pipeline you want to re-run and click **Re-Run**.



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

6 Human Tasks

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

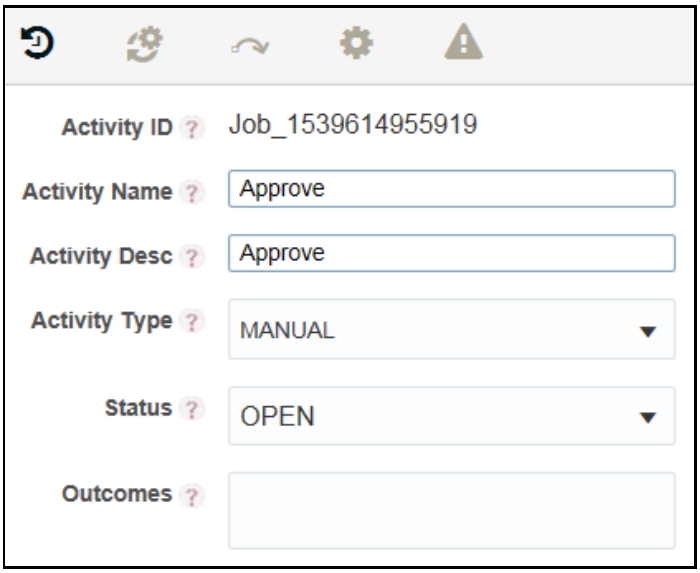


6.1 How to Use Human Task

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

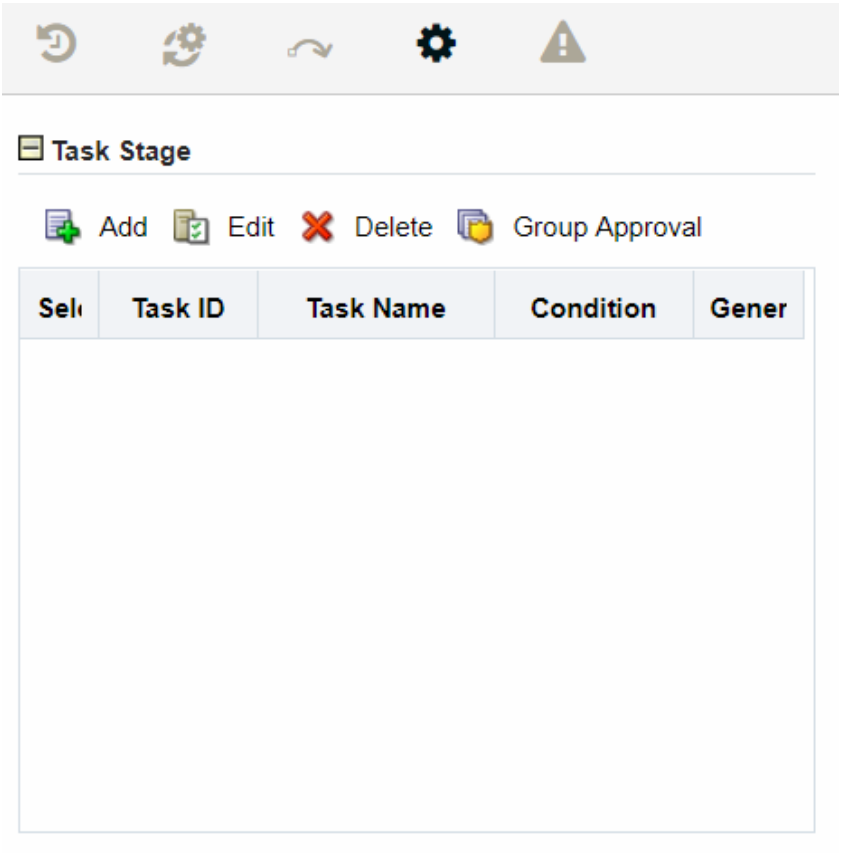


1. **Activity ID** is auto populated.
2. Enter Activity Name and Description.
3. Select the **Status** of the activity from the drop-down list. The list displays the seeded values in the AAI_WF_STATUS_B table.

6.1.2 Action Tab for Creating Tasks/Notification

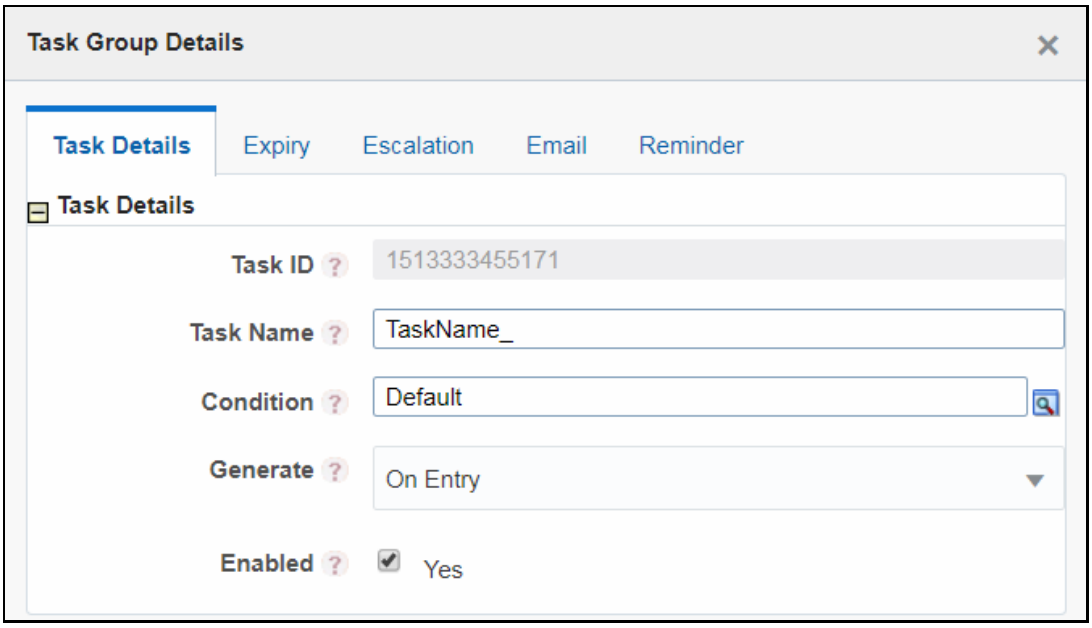
Action/Task is used to inform the assigned user about an action to be completed in the current stage of Workflow. You can add multiple tasks for an activity. A task can be assigned to a user, user group, users with a particular role, or users selected by an application rule. The task is sent to the Application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

1. Click  to define Actions.



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

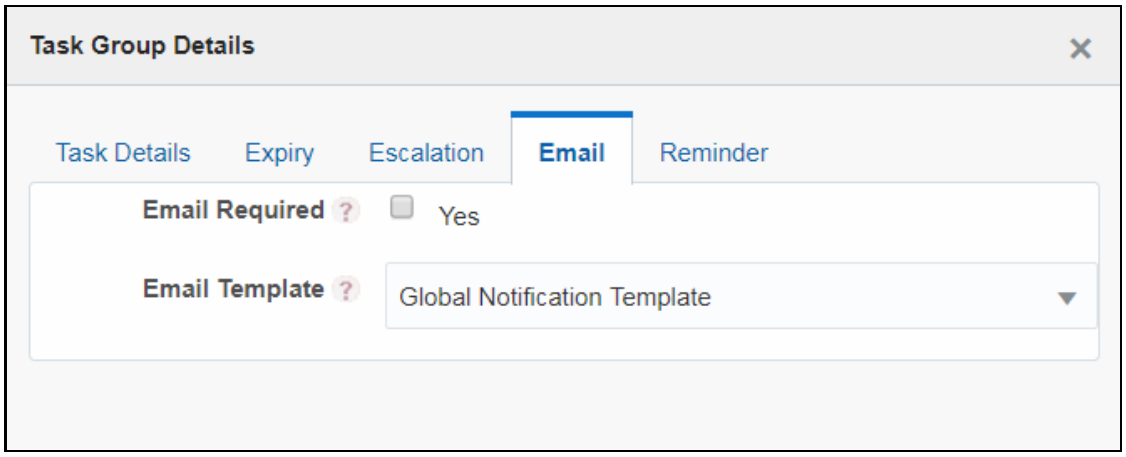
6.1.2.1 Defining Task Details



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 will be 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

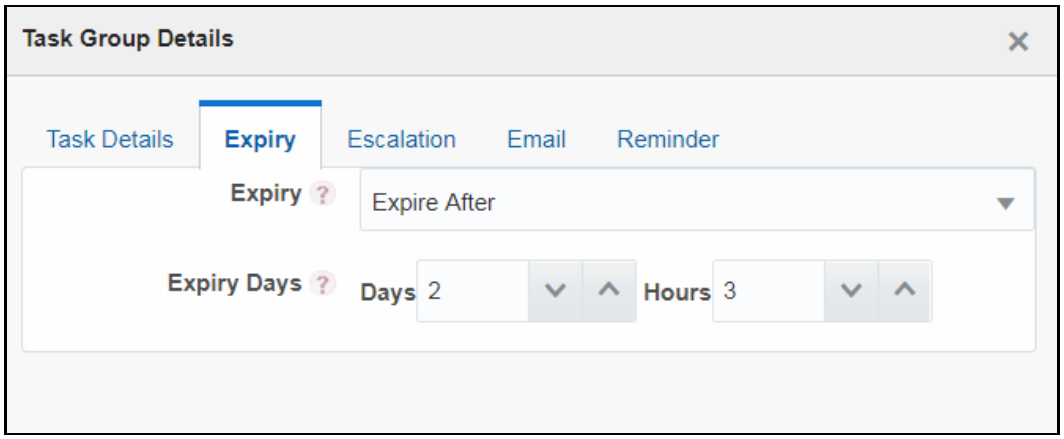


You should have configured Email settings before you set an email notification for your tasks. For more details on how to configure email settings, see Configuring Email for Human Tasks section.

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.

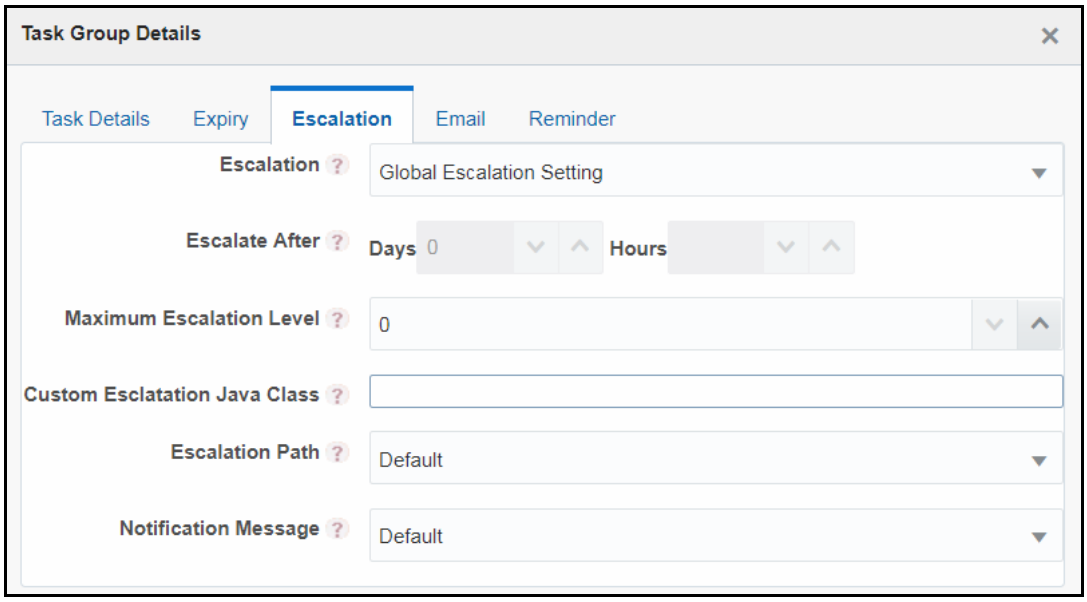
6.2 Additional Functionalities for Human Tasks

6.2.1 Setting Task Expiry



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

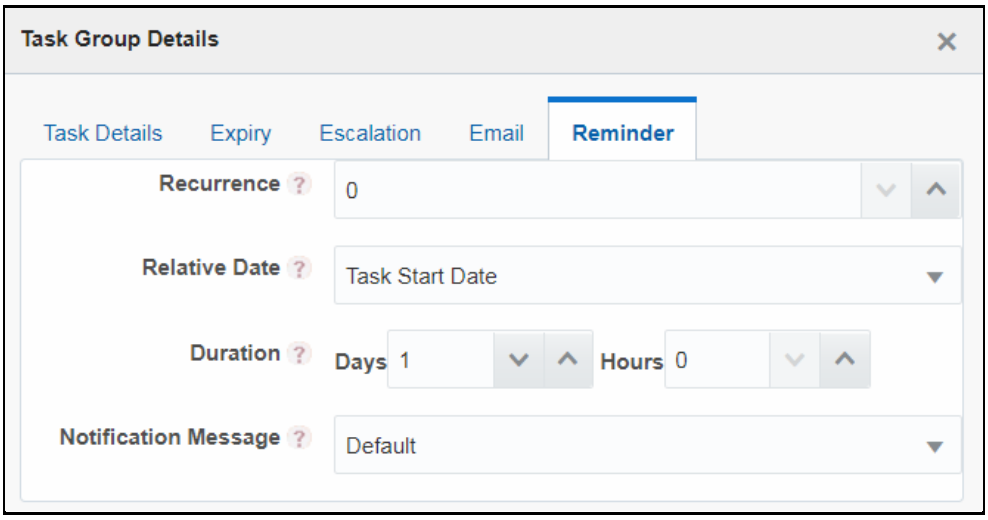


1. Select the **Escalation** criteria:
   - Global Escalation Setting- Select this option to set the task escalation based on a global setting. Global setting can be set from the backend.
   - Never Escalate - Select this option if escalation is not required for the task.
   - Escalate After- Select 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 which 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



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

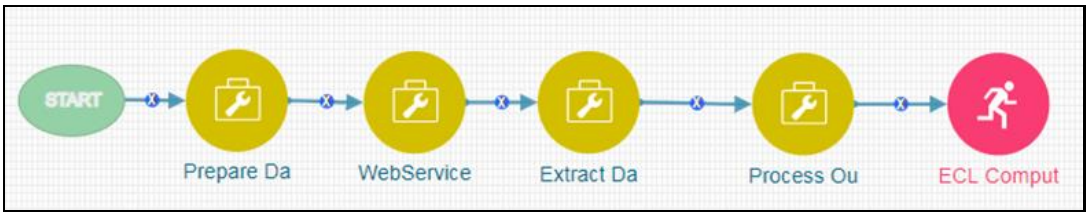
1. Enter the number of times you want to set the reminder from the **Recurrence** field.
2. Select **Task Start Date** if you want to send a reminder after the defined number of days and/or hours, from the start date of the task. Select **Task Expiration Date** if you want to send a reminder before the 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 which 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 typically used to invoke External Model Service through Rest API, External Java APIs, Stored Procedures, and Functions.

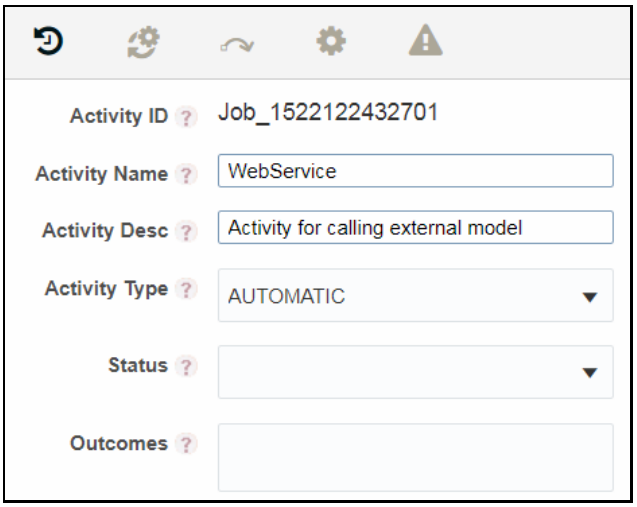
A more detailed explanation of invoking external model service is available in How to invoke External Model through Web Service section.



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.

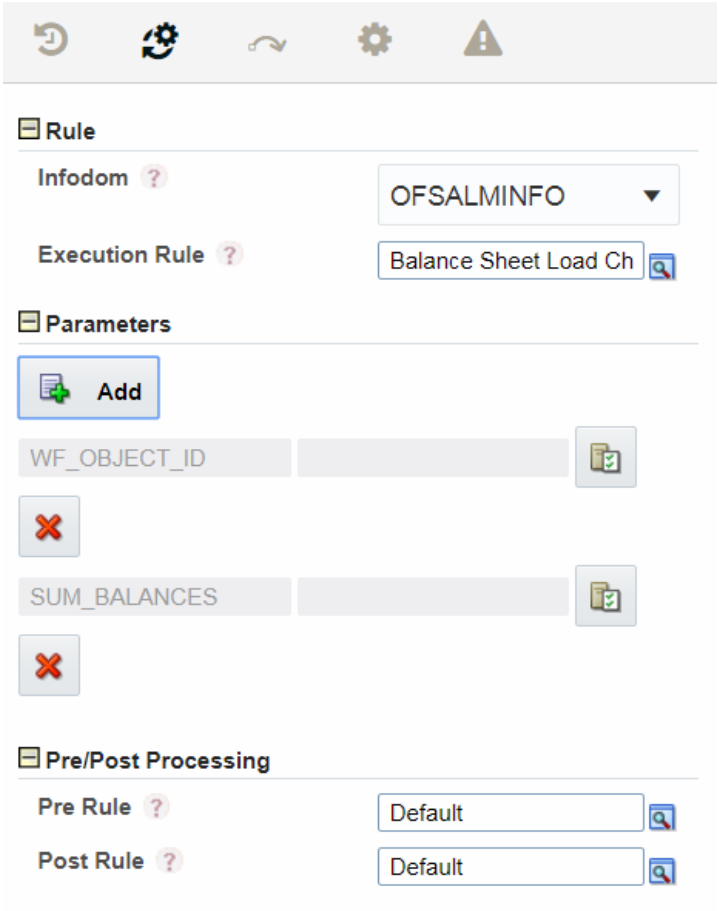
7.1.1 Activity Tab



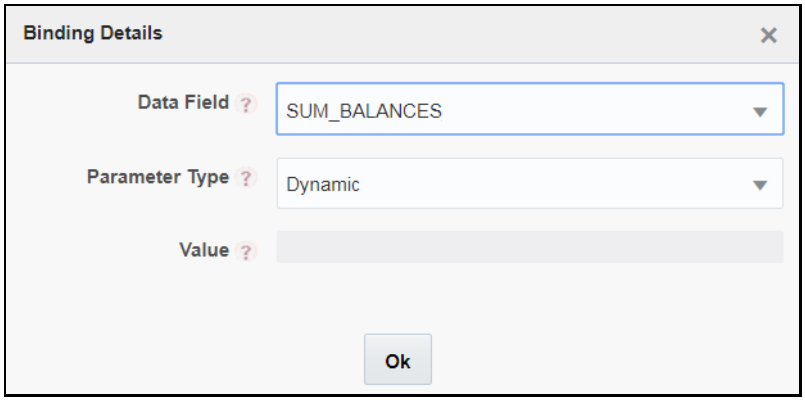
3. **Activity ID** is auto populated.
4. Enter Activity Name and Activity Description.

7.1.2 Implementation Tab

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



6. 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.
7. Select the **Execution Rule** which 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**.
8. 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.



- 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.

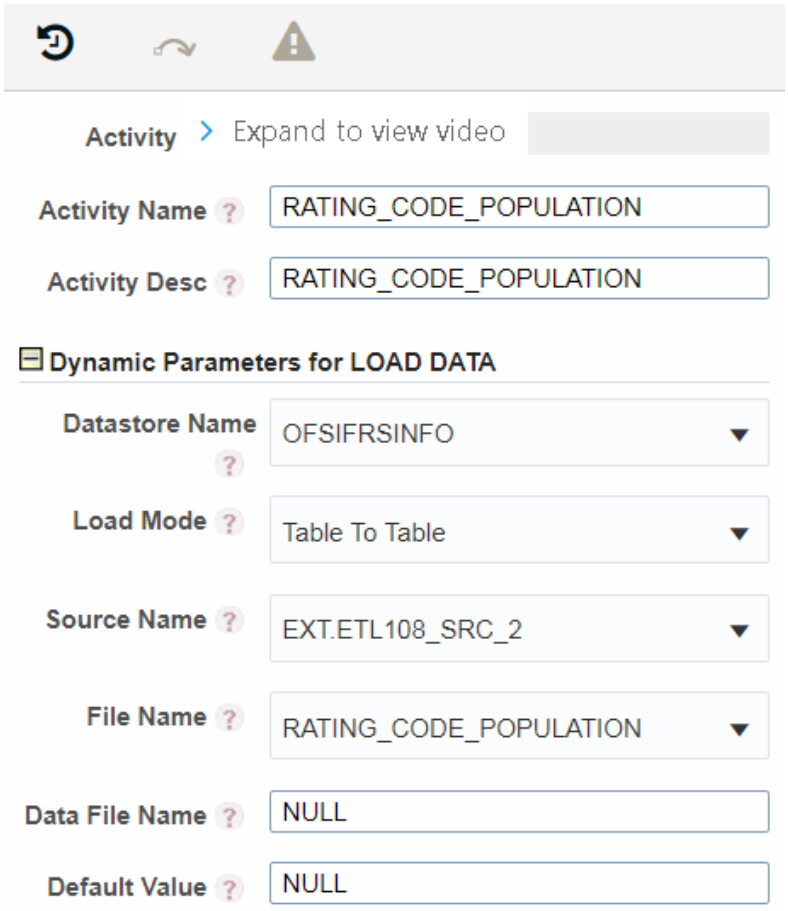
9. Select an application rule which you want to execute before executing the Execution Rule.
   - Click corresponding to **Pre Rule** and select the required application rule.
10. Select an application rule which you want to execute after executing the Execution Rule.
    - Click corresponding to **Post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

8 Configuring OFSAA Tasks in Your Process Flow

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

8.1 How to Use an OFSAA Widget

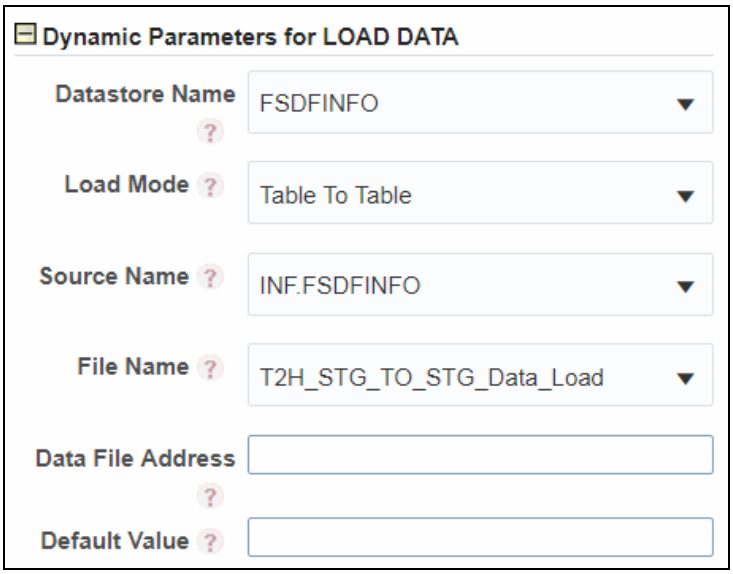
1. From the *Process Flow* tab, click the required widget under the **OFSAA Widget** toolbar in the left pane and click the canvas. The available OFSAA Widgets are LoadT2T, TransformDT, RuleType2, RuleType3, MFModel, Run, and DataQualityGroups.
2. Double-click the widget. On the Right Pane, the *Activity*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



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.

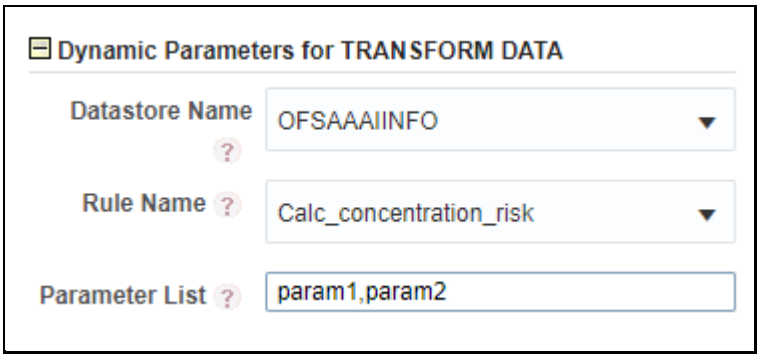
8.1.1 Dynamic Parameters for OFSAA Widgets

8.1.1.1 LoadT2T



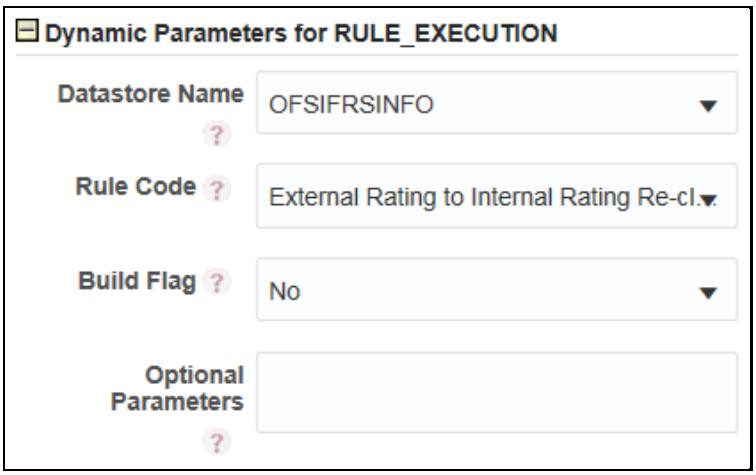
Field Name	Description
Datastore Name	Select the Information Domain in which the Data Mapping you want to execute is present, from the drop-down list.
Load Mode	Table to Table should be selected for Data Mapping definitions such as T2T, T2H, H2T, H2H, and L2H definitions.
Source Name	Select the required 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.

8.1.1.2 TransformDT



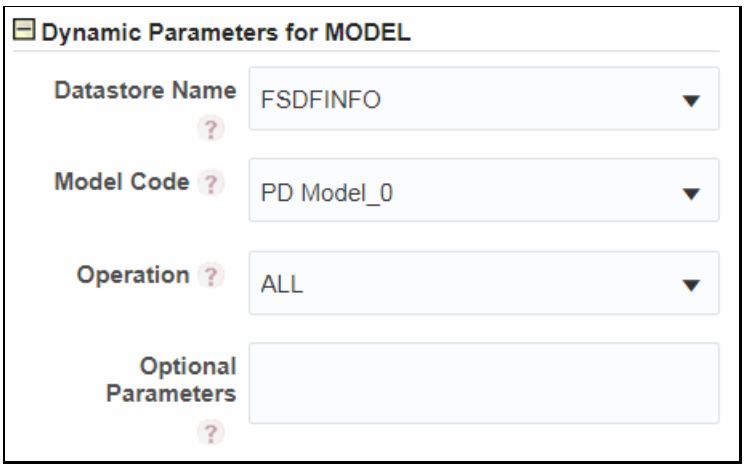
Field Name	Description
Datastore Name	Select the datastore name in which the PLC you want to execute is present from the drop-down list.
Rule Name	Select the Post Load Changes (DT) definition you want to execute from the drop-down list. The list displays the Post Load Changes definitions in the selected 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.

8.1.1.3 RuleType2 or RuleType3



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.

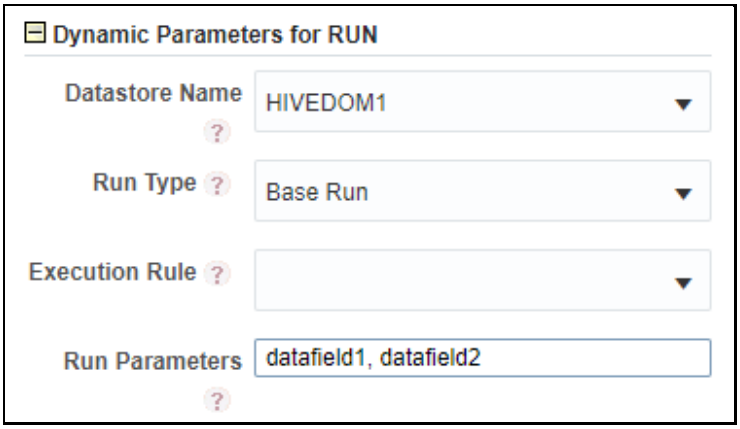
8.1.1.4 MFModel



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.

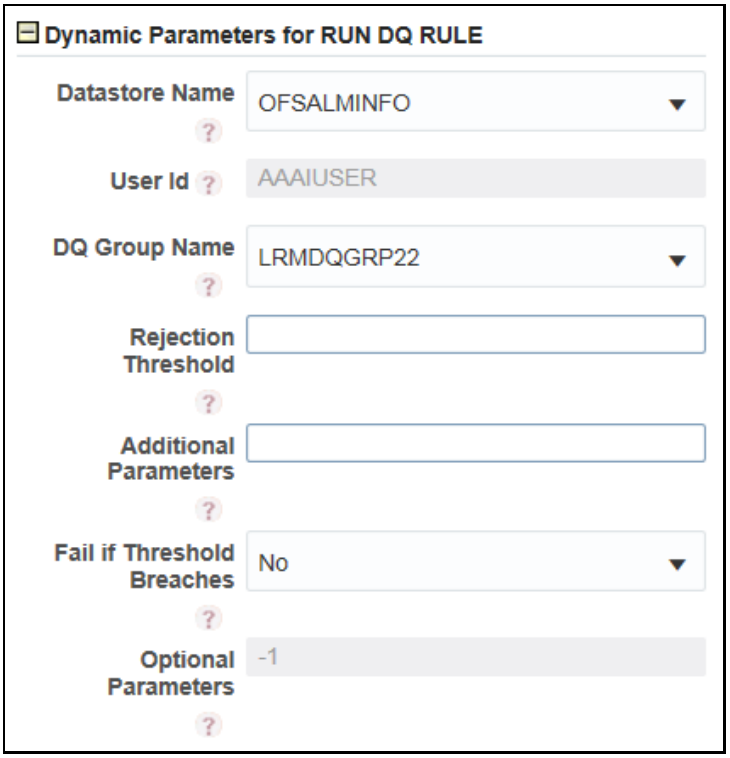
Field Name	Description
Model Code	Display the codes of the EMF Models defined under the selected Infodom.
Operation	The All definition for the Operation field conveys the process of extracting the data from the flat files and applying the run regression on the data extracted. For Batches that are being built for the first time the data will be extracted from the flat files and the run regression will be 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.

8.1.1.5 Run 



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	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.

8.1.1.6 DataQualityGroups



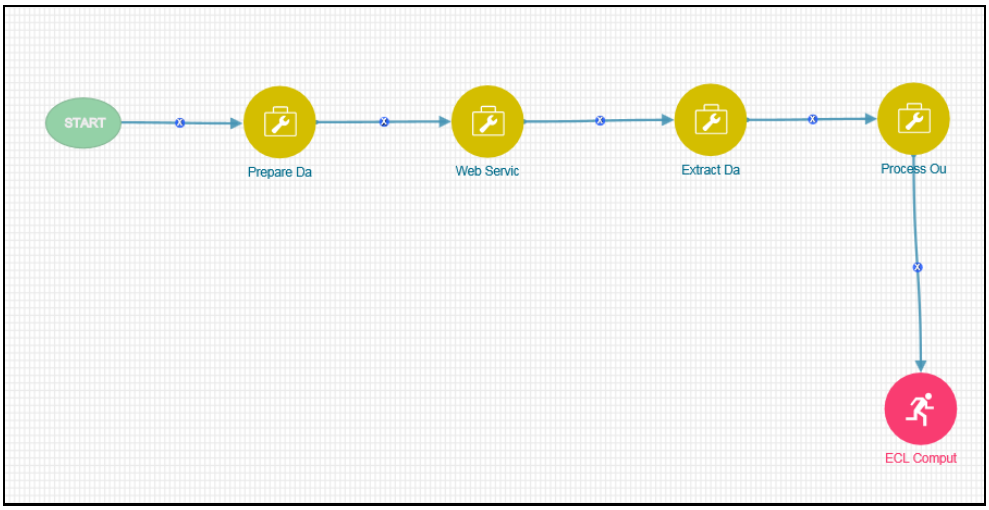
Property	Description
DQ Group Name	Refers to the Data Quality Groups consisting of associated Data Quality Rule definition(s). Select the required DQ Group from the drop-down list.
Rejection Threshold	Specify the percentage of the Rejection Threshold (%) limit in numeric value. This refers to the maximum percentage of records that can be rejected in a job. If the percentage of failed records exceeds the Rejection Threshold, the job will fail. 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 will 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.

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



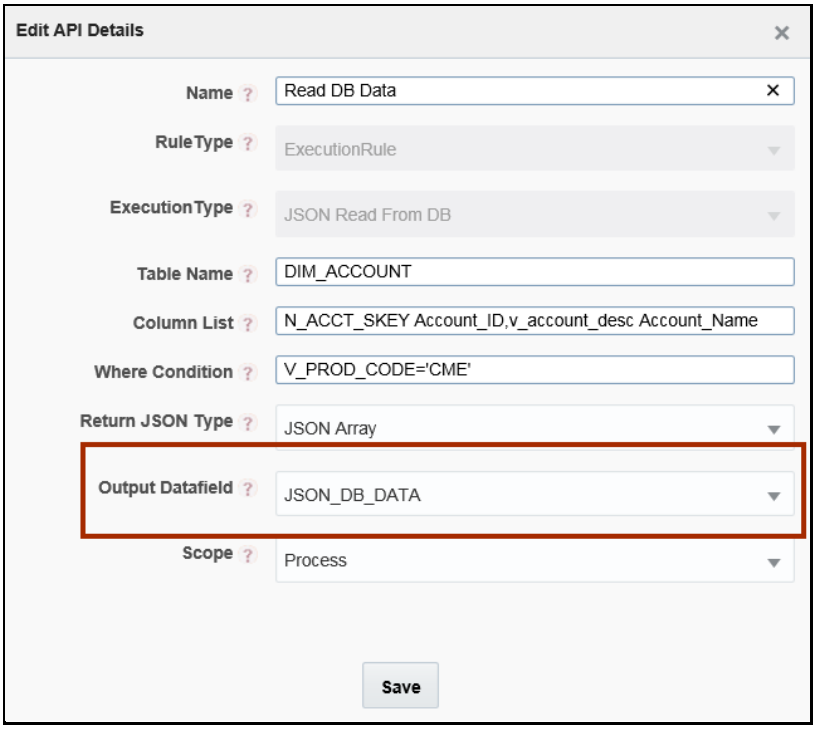
A typical External Web Service invocation will have three steps:

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.

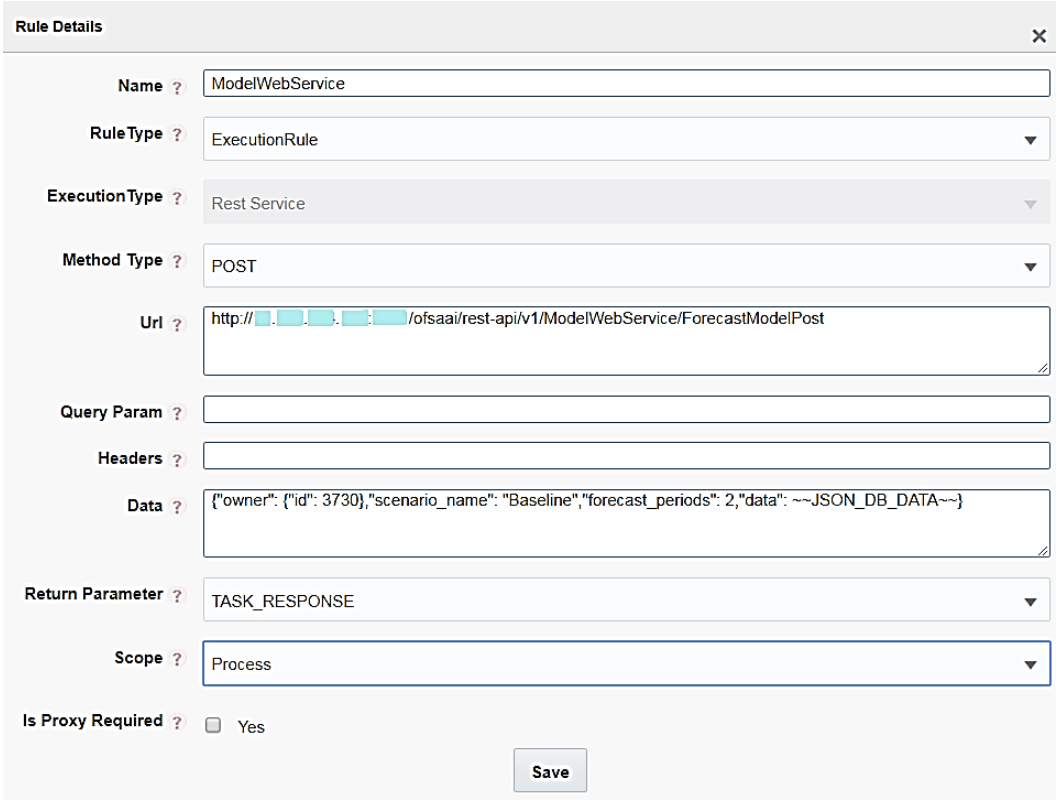


For more information, see JSON Read From DB Application Rule section.

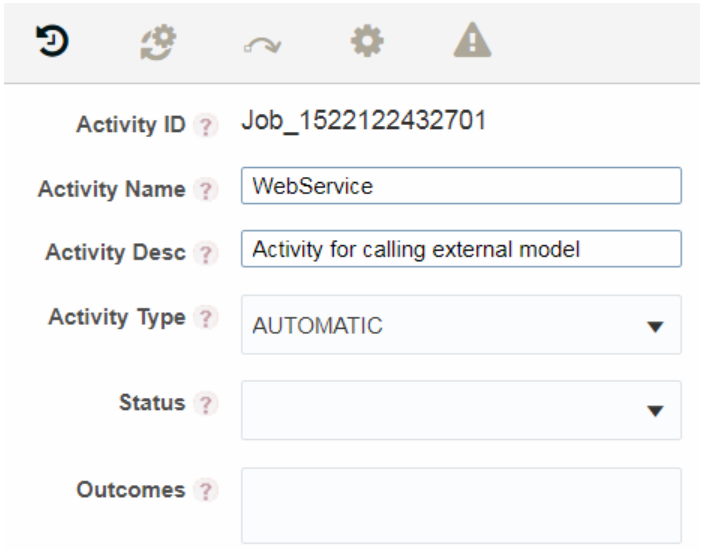
9.1.2 Webservice Invocation

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`.

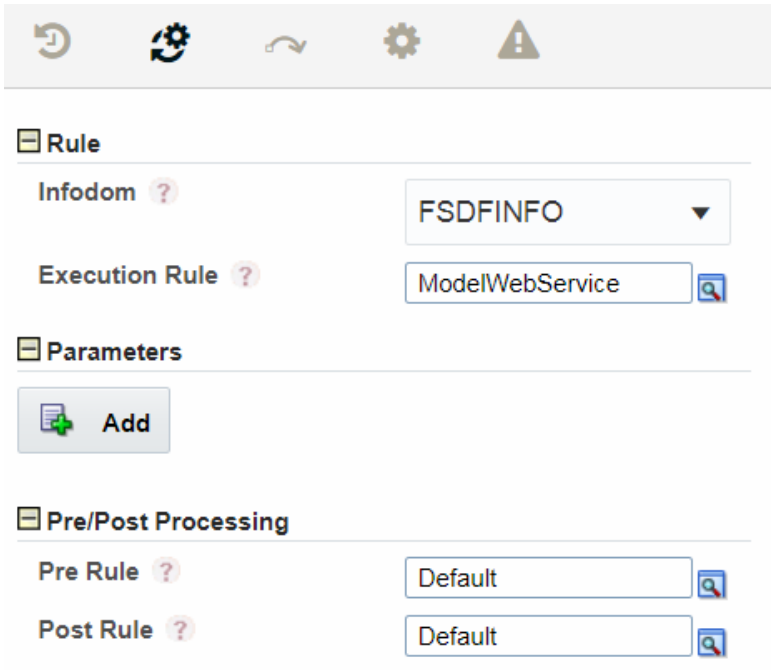
1. Launch the Process for which you want to add an external model.
2. Go to the *Application Rule* tab and click **Add** >**Rest Service**.



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.



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**.

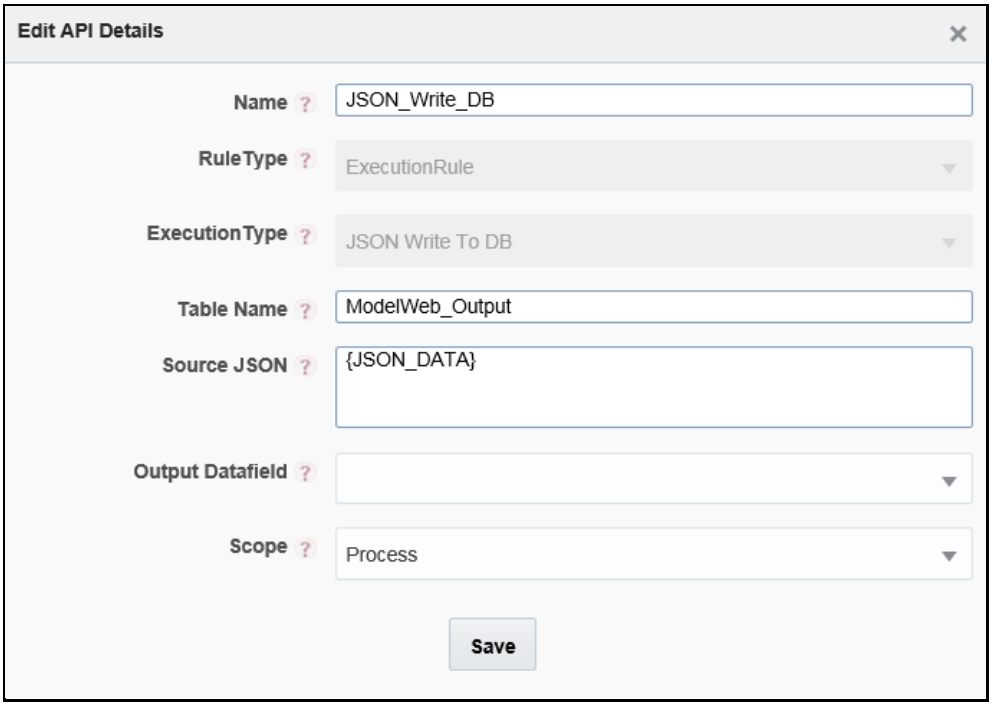


9.1.3 Data Extraction

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:



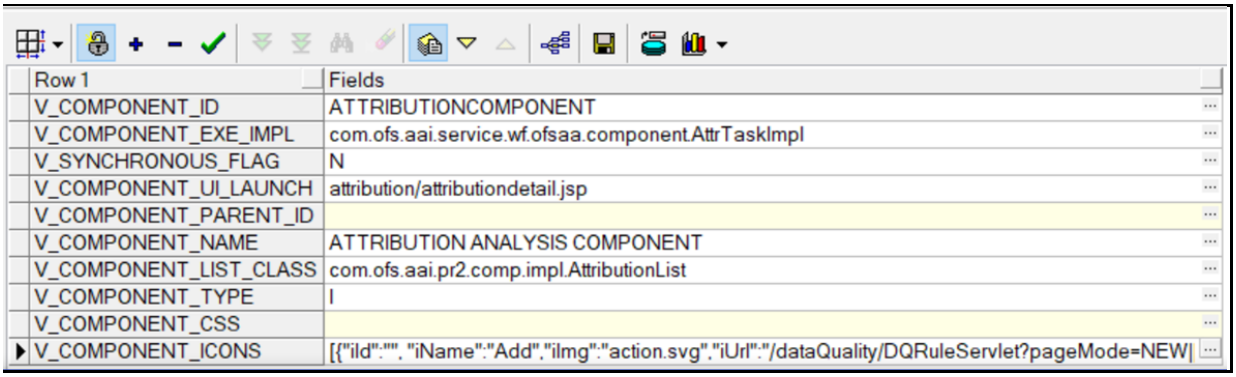
For more information, see JSON Write To DB Application Rule section.

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

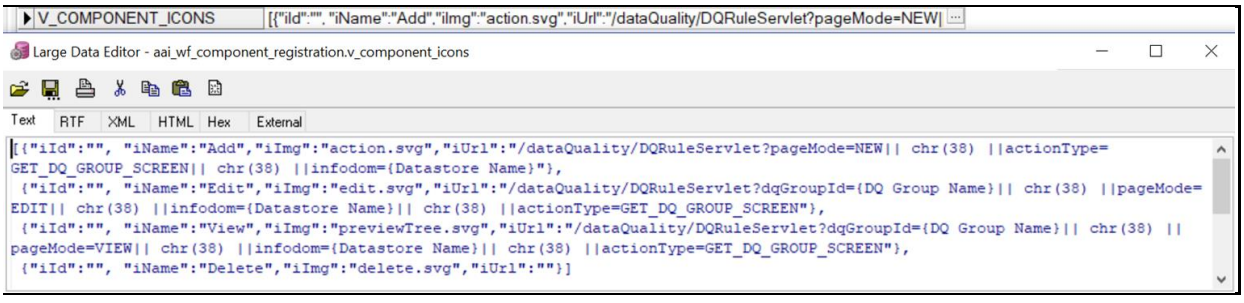
An entry into the AAI_WF_COMPONENT_REGISTRATION table will create a new custom component in the PMF OFSAA widgets.



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	[{"ild":"", "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 which needs to be opened when the component is double clicked in the canvas.
- V_COMPONENT_NAME – Specify a name for the component, which will be displayed for the custom component in the OFSAA Widgets.
- V_COMPONENT_ICONS - All the icons and menu are configured in this field.

For example,



10.2 AAI_WF_COMPONENT_PARAMETERS Table

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

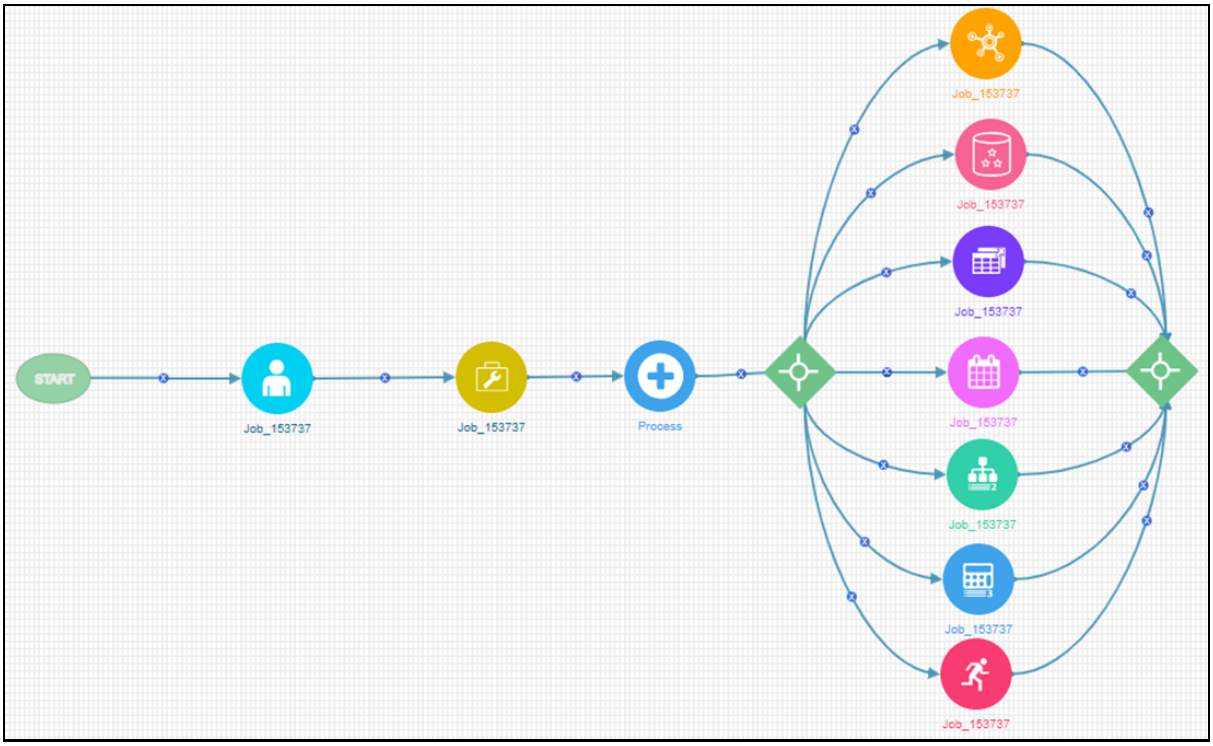
	V_COMPONENT_ID	V_PARAMETER_NAME	I_PARAMETER_ORDER	I_DISPLAY_ORDER	V_PARAMETER_CODE	TYPE_OF_DISPLAY
1	LOAD DATA	IP Address	1	3	SMSLB.IP_ADDR	2
2	LOAD DATA	Datastore Type	2	1	ICC.FE.LBL_DATA_STORE_TYPE	2
3	LOAD DATA	Datastore Name	3	2	ICC.FE.LBL_DATA_STORE	2
4	LOAD DATA	Load Mode	4	4	ICC.FE.LBL_LOADMODE	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	ETL.PROP_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.

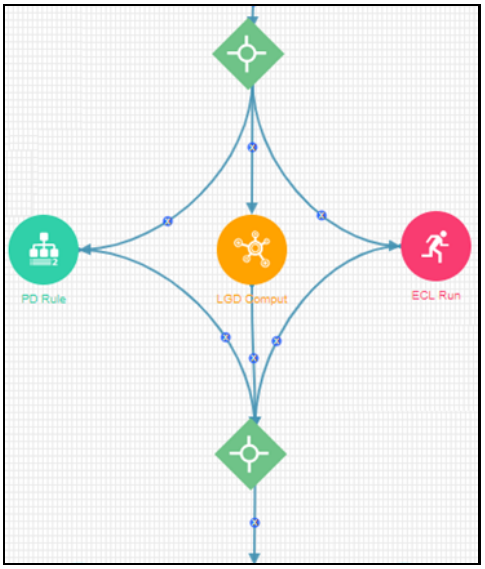
11 Executing Parallel Tasks

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



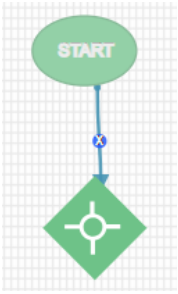
In the shown example, after  is executed, Parallel Gateway  gets invoked. All the OFSAA components which are placed between Parallel Gateways  are executed simultaneously. It waits till all components are executed, and then moves to the next activity in the Process Flow.

11.1 How to Use Parallel Gateways



This section explains how to design 3 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).

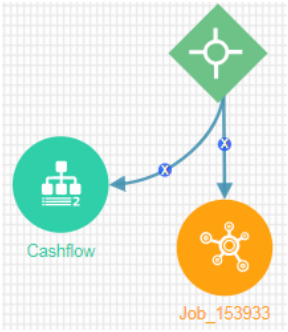
6. Click  **RuleType2** from the *OFSAA Widgets* pane, and then click the drawing canvas where you want to place it.

7. Click **Transition Curve** from the *Tools* pane and click Parallel Gateway and then click Job_153933.

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.

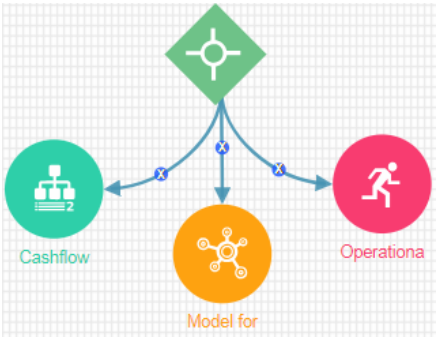
10. Click **Transition Curve** from the *Tools* pane and click Parallel Gateway and then click Job_153933.



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.

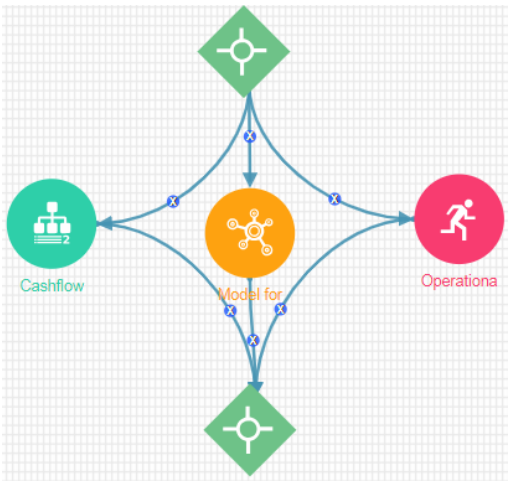
13. Click **Transition Curve** from the *Tools* pane and click Parallel Gateway and then click .



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.

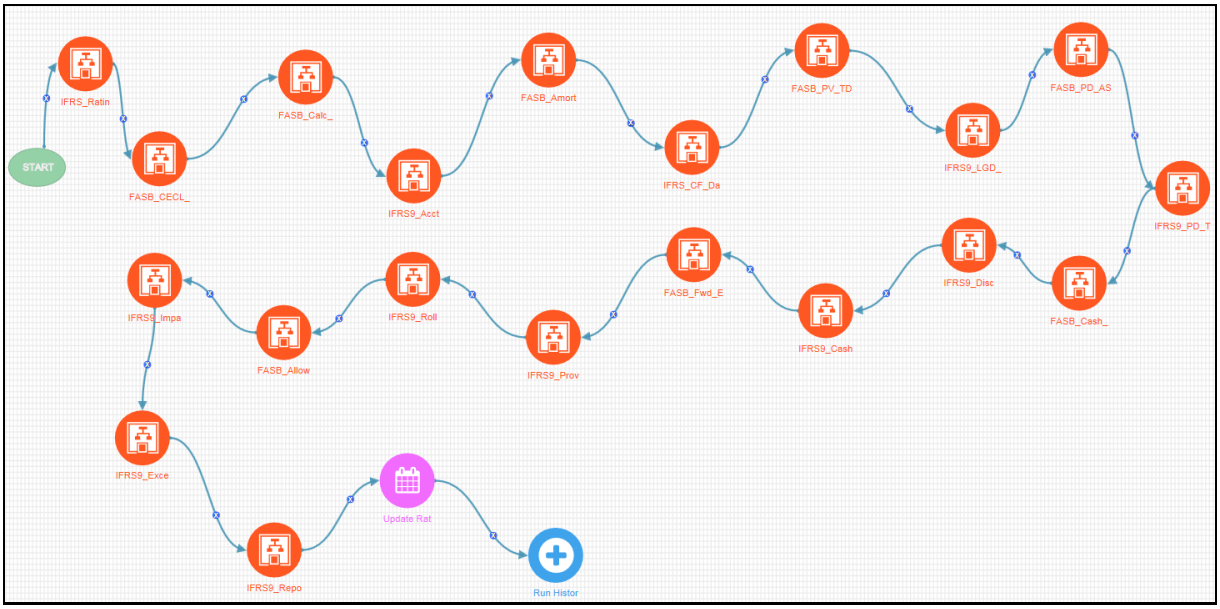
16. Draw transitions from RuleType2, MFModel and Run widgets to the merging  Parallel Gateway as shown:



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

12 Calling another Pipeline from Your Parent Pipeline

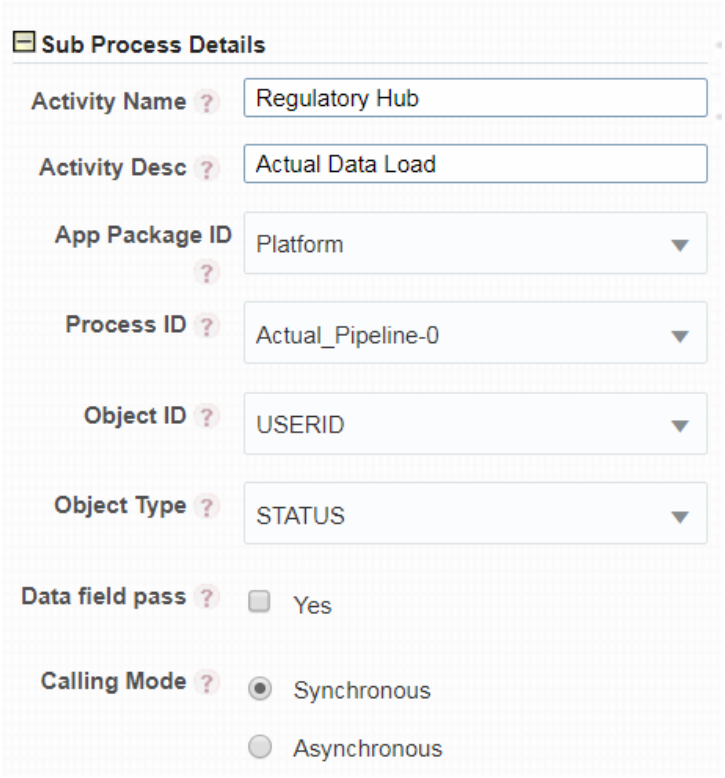
Reusability is important while designing your pipeline. Sub Pipeline is the mechanism in the PMF to call another pipeline from your parent pipeline.



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 . The *Sub Process Details* window is displayed.



3. Enter the details as given in the table:

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 which 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.
Calling Mode	• 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. And the parent workflow proceeds to the next activity.

13 Configuring Email for Human Tasks

Following table entries should be made for setting Email notification:

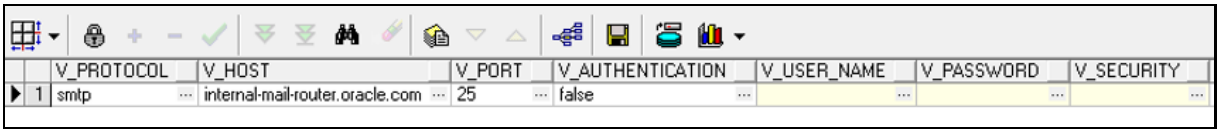
13.1 AAI_EMAIL_CONFIG Table

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 will be 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.



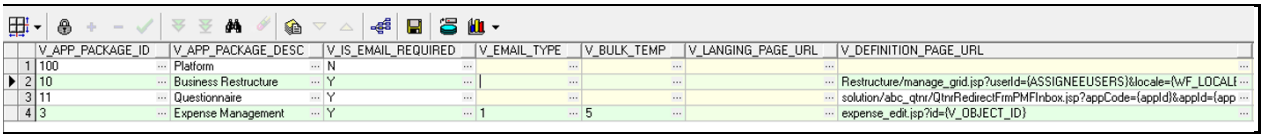
	V_PROTOCOL	V_HOST	V_PORT	V_AUTHENTICATION	V_USER_NAME	V_PASSWORD	V_SECURITY
1	smtp	internal-mail-router.oracle.com	25	false			

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 will be sent with all the pending notifications from the last trigger, as a PDF attachment. Once 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.

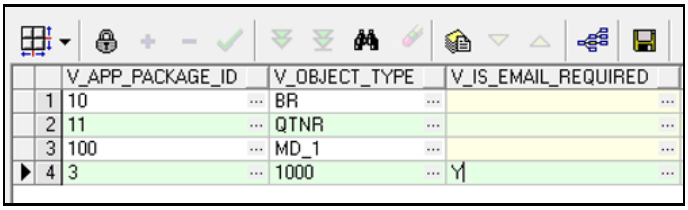


	V_APP_PACKAGE_ID	V_APP_PACKAGE_DESC	V_IS_EMAIL_REQUIRED	V_EMAIL_TYPE	V_BULK_TEMP	V_LANGING_PAGE_URL	V_DEFINITION_PAGE_URL
1	100	Platform	N				
2	10	Business Restructure	Y				Restructure/manage_grid.jsp?userId={ASSIGNEEUSERS}&locale={WF_LOCALE
3	11	Questionnaire	Y				solution/abc_qtnr/QtnrRedirectFrmPMFInbox.jsp?appCode={appId}&appId={app
4	3	Expense Management	Y	1	5		expense_edit.jsp?id={V_OBJECT_ID}

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.



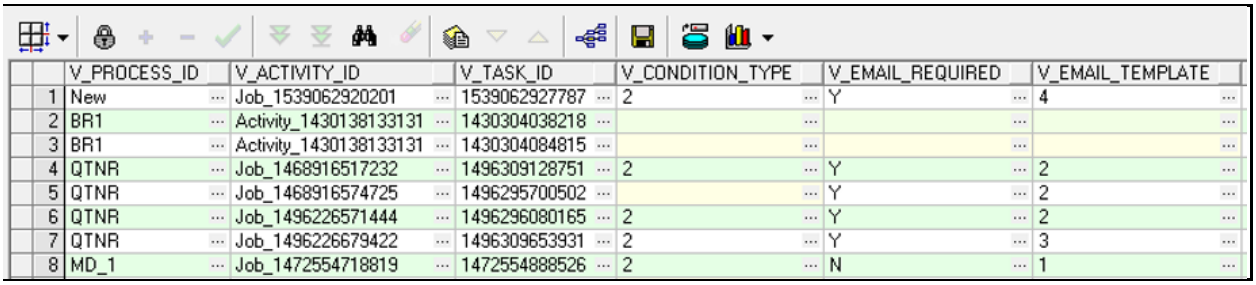
	V_APP_PACKAGE_ID	V_OBJECT_TYPE	V_IS_EMAIL_REQUIRED
1	10	BR	
2	11	QTNR	
3	100	MD_1	
4	3	1000	Y

13.4 AAI_WF_ACTIVITY_TASK_BTable

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.



	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.

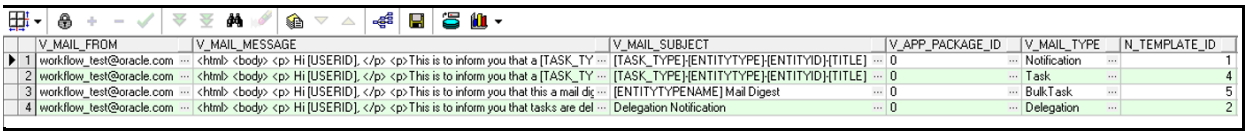
V_USER_ID	N_EMAIL_NOTIF_REQ
USER1	1
USER2	2

- 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 will be sent with all the pending notifications from the last trigger, as a PDF attachment. Once 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 which needs to be sent.



	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> Hi [USERID], This is to inform you that a [TASK_TY	[TASK_TYPE]-[ENTITYTYPE]-[ENTITYID]-[TITLE]	0	Notification	1
2	workflow_test@oracle.com	<html> <body> Hi [USERID], This is to inform you that a [TASK_TY	[TASK_TYPE]-[ENTITYTYPE]-[ENTITYID]-[TITLE]	0	Task	4
3	workflow_test@oracle.com	<html> <body> Hi [USERID], This is to inform you that this a mail dig	[ENTITYTYPENAME] Mail Digest	0	BulkTask	5
4	workflow_test@oracle.com	<html> <body> Hi [USERID], This is to inform you that tasks are del	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.

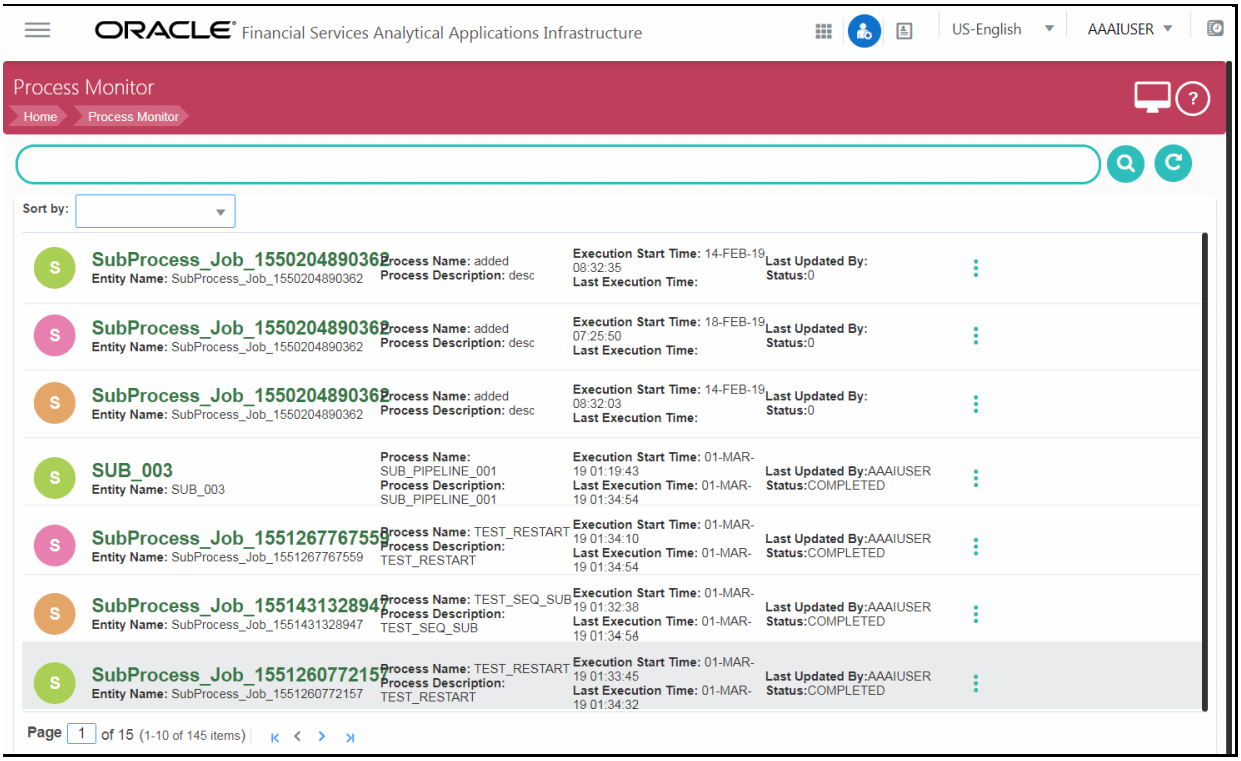
V_USR_ID	V_EMAIL
USER1	user1@oracle.com
USER2	user2@oracle.com

13.9 AAI_MAIL_AUDIT_TRAIL Table

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

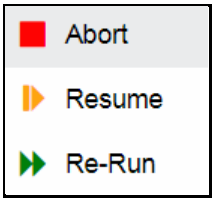
14 Process Monitor

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 should be mapped to the function role WFMACC (Workflow Monitor Access) to access the Process Monitor window.



This window displays all the Workflows which 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:



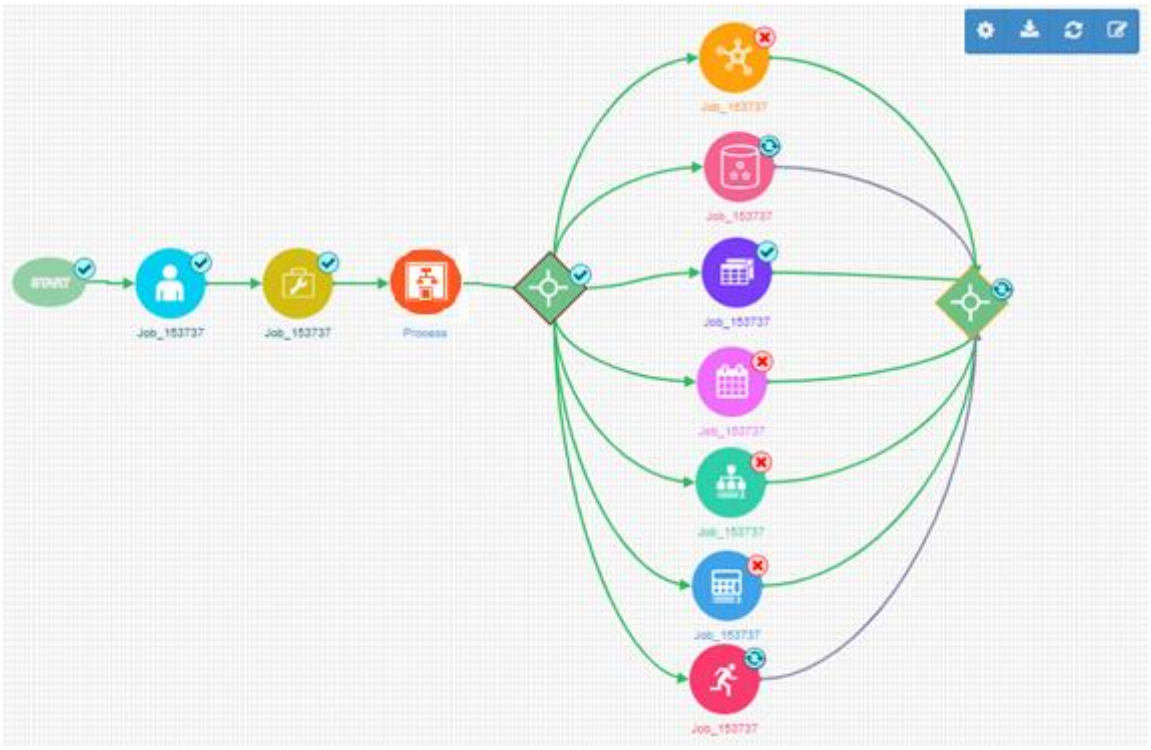
- Abort- to abort an ongoing Run Pipeline.
- Resume- to resume a Run Pipeline
- Re-run- to execute a Run Pipeline again irrespective of the previous execution status.

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.



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

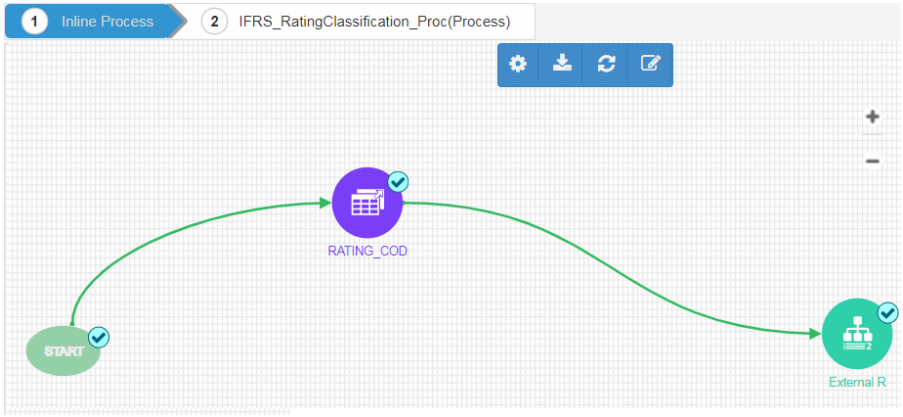
- This indicates that the execution of activity is successful.

- 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.

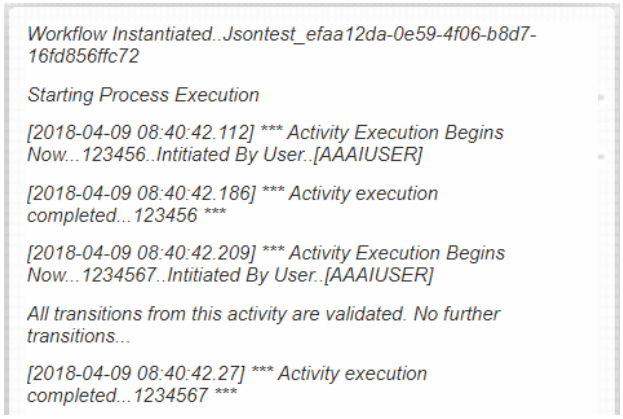
- This indicates that the activity is yet to be executed.

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



This figure shows the status of the tasks inside a Sub Pipeline Process.

2. Click to refresh the pane. Click to view the *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.

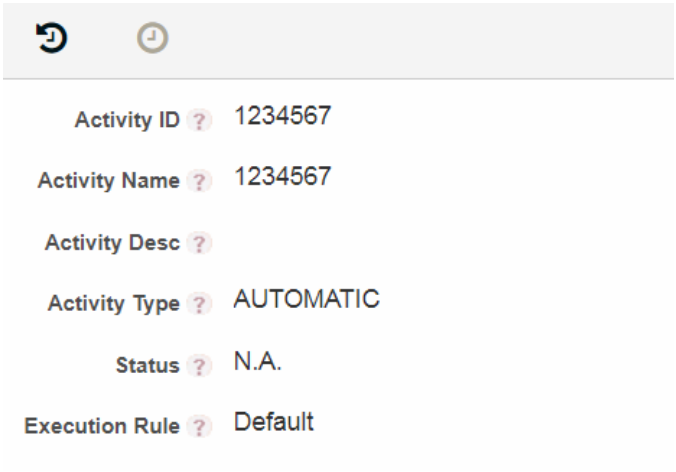
You can use and to zoom in and out the *Process Monitor* window.

14.2 Viewing Activity Logs

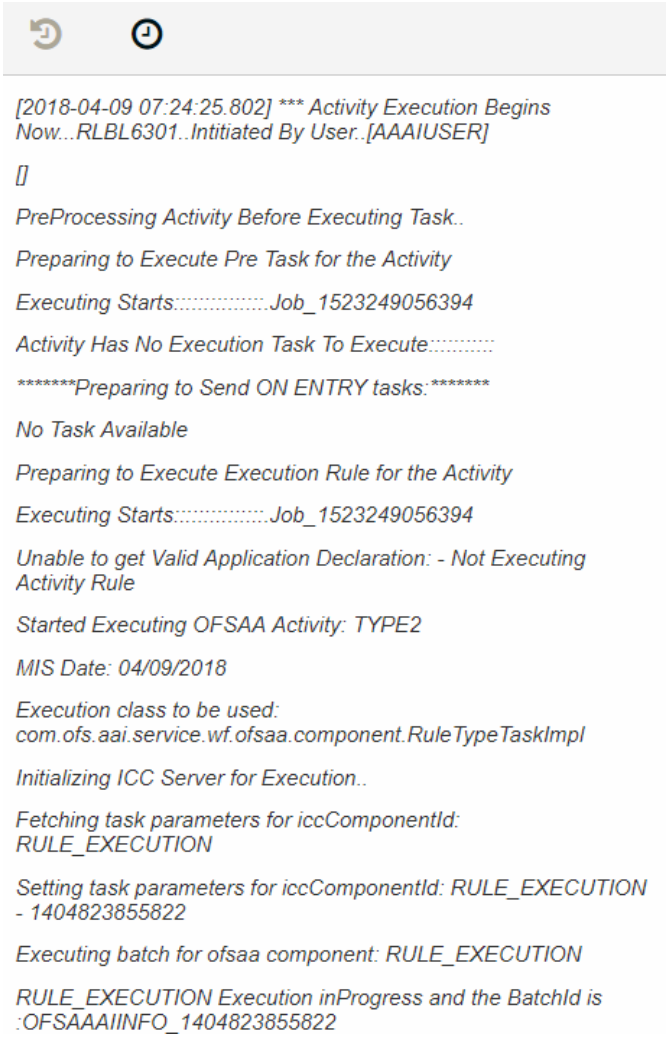
This feature allows you to view logs of each activity from the Process Monitor window.

To view Activity logs

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



2. Click to view the Activity Logs.



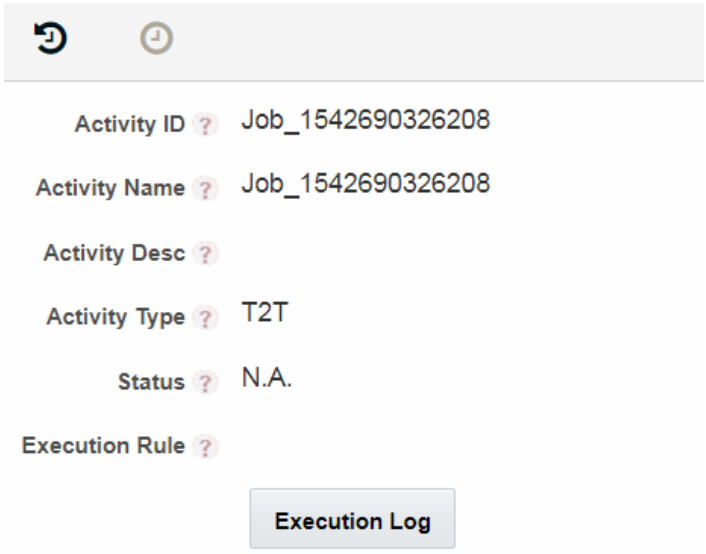
The log shows all the execution stages of the selected Activity, if it is already executed. Otherwise, it will be blank.

14.3 Viewing Execution Log for OFSAA Tasks

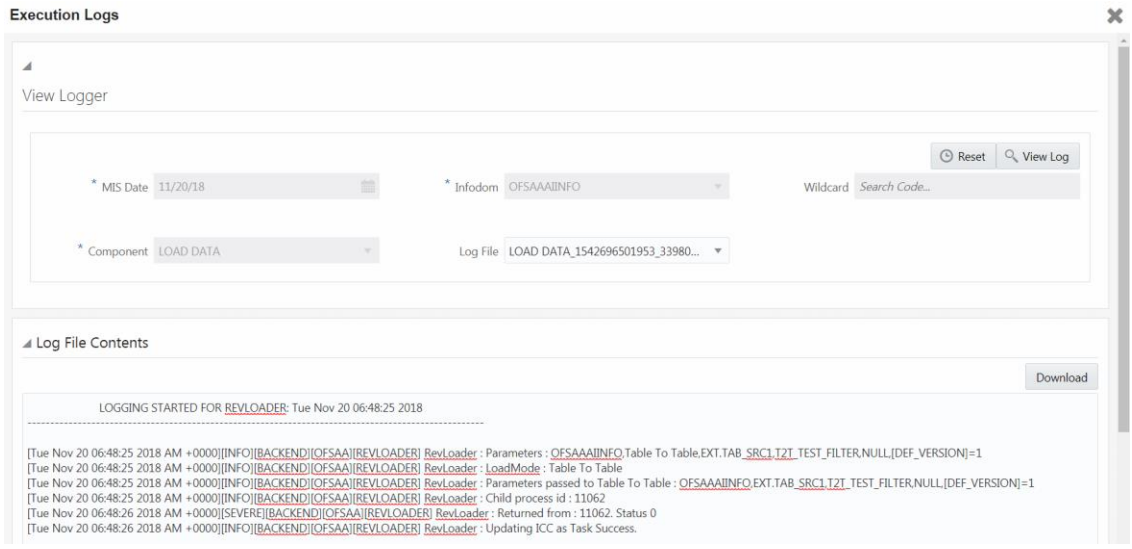
This option allows you to view the execution logs for OFSAA widgets. This option is not available for Service tasks or human tasks.

To view Execution logs

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



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

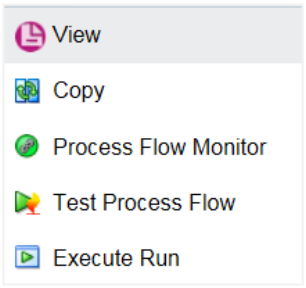
```
URL: <contextPath>/PMFService/startWorkflowProcess

Method: POST

Consumes("application/json")

Produces("text/plain")

Sample Input Params:

"{\"SummaryPayload\":{\"objectid\":\"123\",\"objecttype\":\"QTNR\",\"in
fodom\":\"OFSAAAIINFO\",\"segment\":\"OFSAAAIINFO\",\"userid\":\"AAAIUS
ER\",\"locale\":\"en_US\",\"securitymap\"

:{},\"applicationparams\"

:{\"testparam\":\"value1\",\"testparam2\":\"value2\"}}}";
```

15.2 Within PMF Summary Screen UI

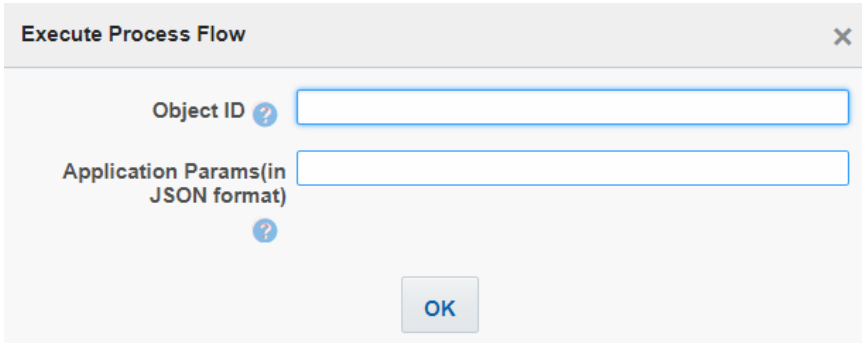
15.2.1 Using Test Workflow

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.



3. Enter an **Object ID** to identify the Process flow. This will be displayed as Entity ID in the *Process Monitor* window.
4. Enter the **Application Parameters** which 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

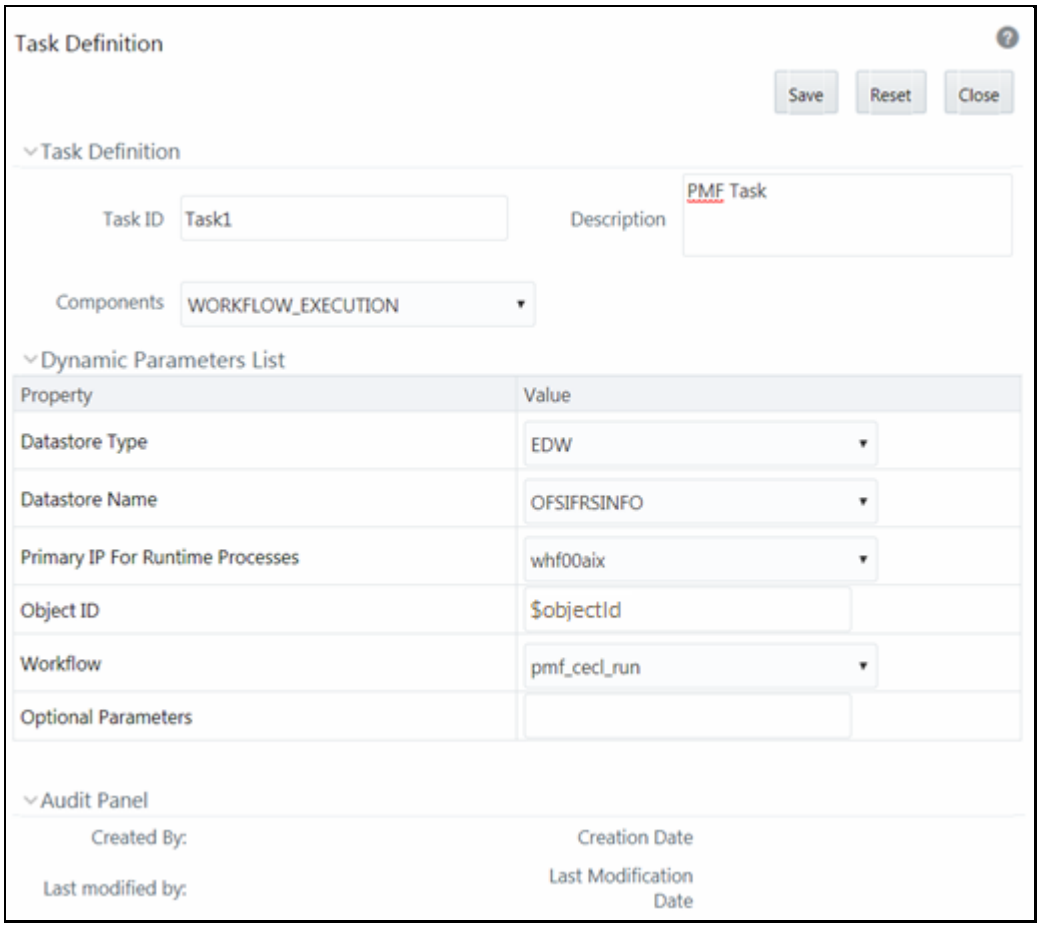
See the Executing Run Pipeline section on how to invoke a Run pipeline from the PMF screen.

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.



   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

16.1 Configuring Group Approval for Human Tasks

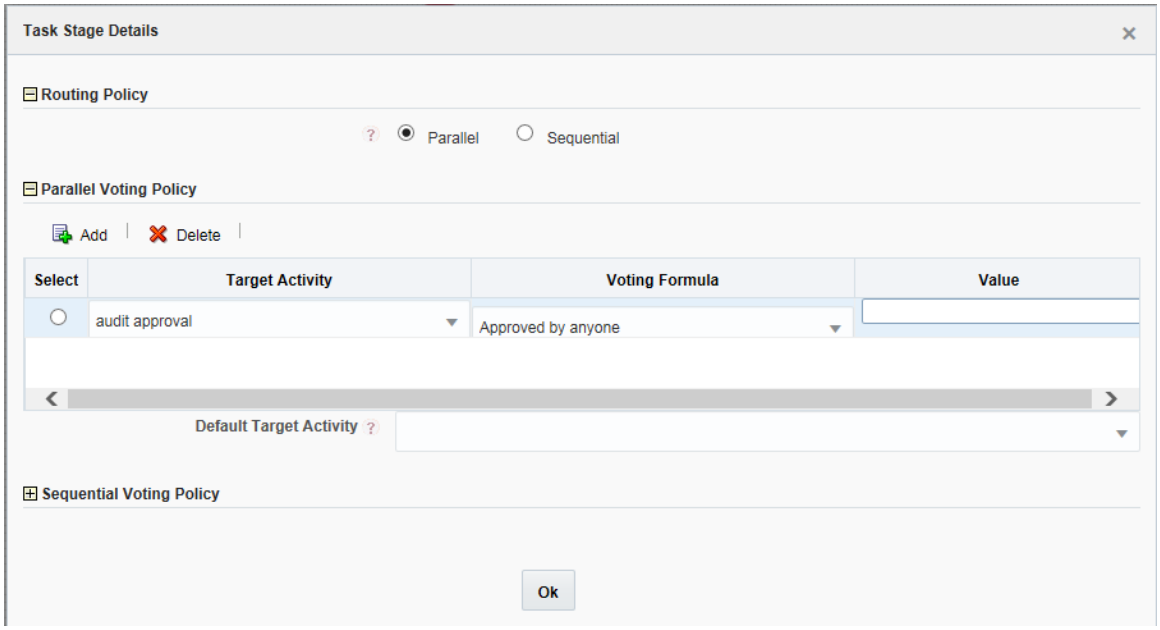
Group approval/Group Consensus can be used to decide whether the flow has to move to a particular activity based on the response of a single member, majority of members of the group, or all members of the group. If more than one group is present, then you can design to move the flow to a particular activity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



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.

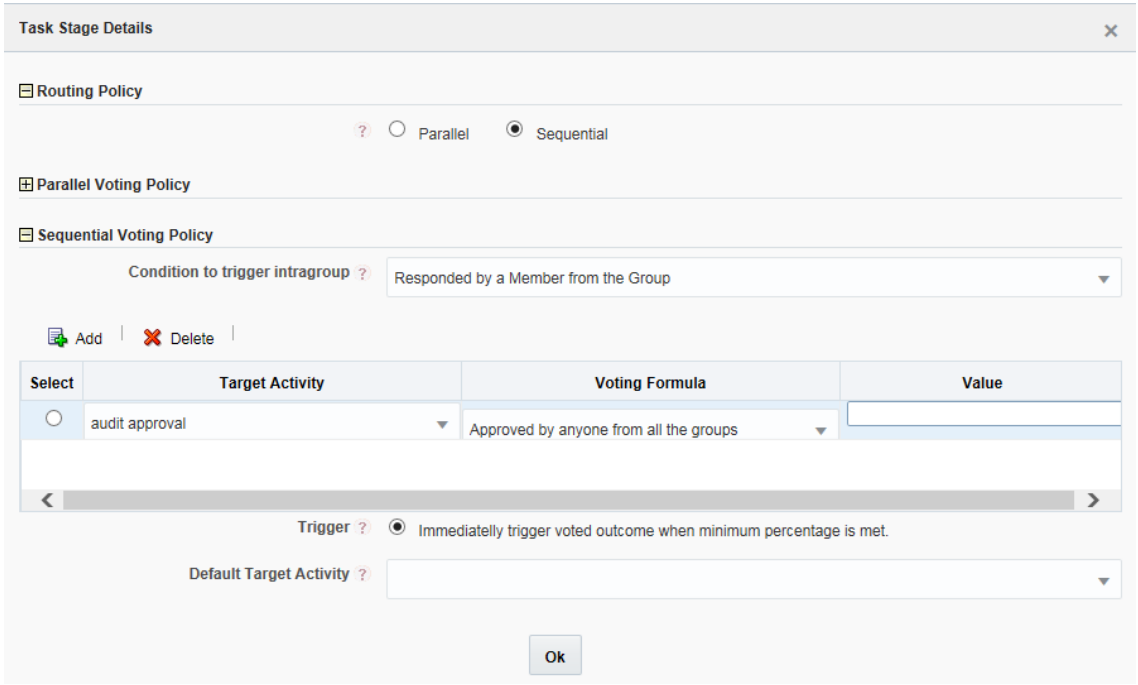
7. Select the required **Target Activity** from the drop-down list.
8. Select the required option from the **Voting Formula** drop-down list. The options are:
   - **Approved by anyone**- If any one of the users from any task group chooses the selected **Target Activity**, the flow moves to the selected 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 which will be 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.



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

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
`[?(<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.

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")])
```

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'])]
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

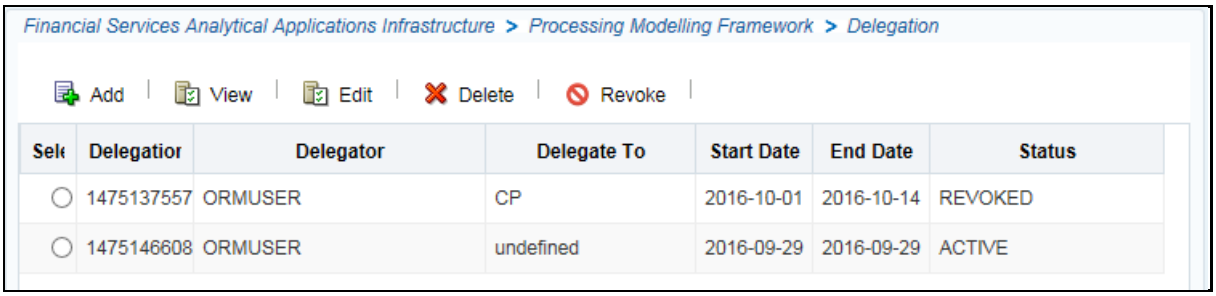
16.2.4 Path Examples

```
{
    "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.

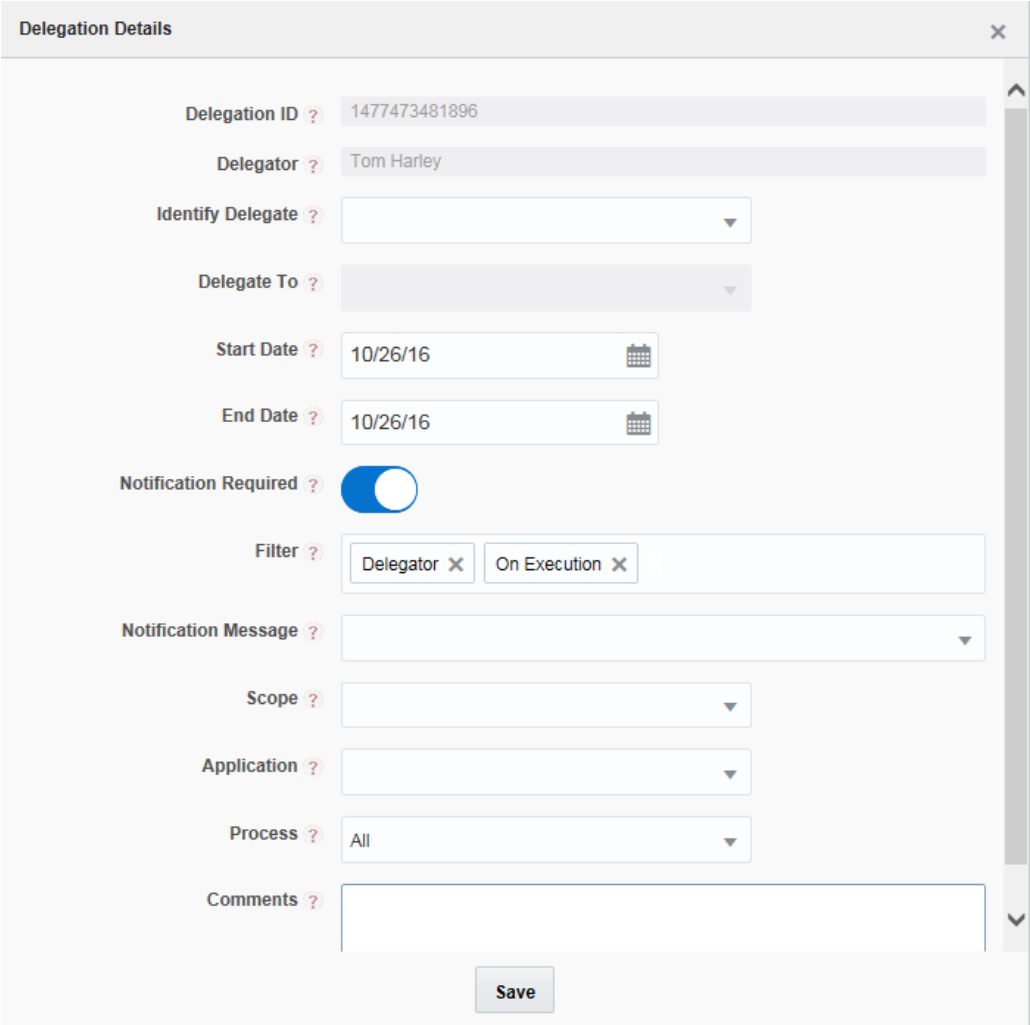


This window displays all the delegations which are defined by the logged-in user with details such as Delegation ID, Delegator, Delegate To, Start Date, End Date, and Status. You can add a new Delegation, view, modify, delete and revoke a delegation.

16.3.1 Adding a Delegate

To add a delegate

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



2. Enter the details as tabulated:

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	Select the required option from the drop-down list. The options are: • 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	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.
Start Date and End Date	Specify the duration for which you want to delegate your tasks by selecting the Start Date and End Date from the calendar.
Notification Required	Turn ON the toggle button if you want to send a notification to the delegate or delegator.
Filter	This field is enabled only if the Notification Required toggle button is turned ON. Select to whom you want to send the notification. You can set to send notification to Delegator and Delegate. Select when you want to send the notification. The options are On Defining and On Execution.
Notification Message	This field is enabled only if the Notification Required toggle button is turned ON. Select the notification message you want to send to the delegate or delegator.
Scope	Select the scope of the delegation from the drop-down lists. The options are: • 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	This field is enabled only if Application or Process is selected as Scope. Select the required Application from the drop-down list. All your tasks related to the selected application will be delegated to the selected user.
Process	This field is enabled only if Process is selected as Scope. 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 will be delegated to the selected user.
Comments	Enter if you want to add any comments for the delegation.

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

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(infodom,false);
```

For example:

```
public boolean testMethod(String attr1) {
    Connection atomicConn = null;
    PreparedStatementDecorator prepStatement = null;
    ResultSet rs = null;
    try {
      atomicConn = ConnectionAdapter.getDBConnections(infodom,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(prepStatement);
   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);
```