

Oracle Financial Services

Analytical Applications Infrastructure

Extension Pack



Release 8.1.2.0.0

F83878-01

February 2023

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Oracle Financial Services Analytical Applications Infrastructure Extension Pack, Release 8.1.2.0.0

F83878-01

Copyright © 2000, 2023, Oracle and/or its affiliates.

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

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

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

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

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

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

Contents

1 Preface

1.1	About this Manual	1-1
1.2	Audience	1-1
1.3	Recommended Skills	1-1
1.4	Recommended Environment	1-2
1.5	Prerequisites	1-2

2 Overview of OFS AAIEP

2.1	What is New in this Release of OFS AAIEP	2-1
2.1.1	New Features in Release 8.0.8.0.0	2-1
2.1.2	Deprecated Features	2-2
2.1.3	Desupported Features	2-2

3 Categorization of Pipelines

3.1	User Role Mapping and Access Rights	3-2
-----	-------------------------------------	-----

4 Access OFS AAIEP

5 Distributed Processing Capabilities

5.1	Distributed Activation Manager (AM) Based Processing	5-1
5.1.1	Set Up Secondary AM Server	5-1
5.1.2	Configure the OFSAA Instance through Load Balancer to Distribute Batch Tasks on Multiple AM Nodes	5-4
5.1.3	Execute Batches on Multiple AM Nodes	5-4

6 Content Management Integration

6.1	Configurations for Document Upload to Multiple Libraries	6-2
-----	--	-----

7 Event-Based Orchestration of Process Flow

7.1	Producer Activity	7-2
7.2	Consumer Activity	7-2
7.3	Seed Data in the Event and Event Subtype Tables	7-2
7.4	Create an Event Process Flow in the Process Modeller	7-3
7.5	Configure Event Producer	7-5
7.5.1	Configure Activity	7-5
7.5.2	Configure Transition	7-6
7.5.3	Configure Notifications	7-7
7.6	Configure Event Consumer	7-8
7.6.1	Configure Activity	7-9
7.6.2	Configure Consumer	7-10
7.6.3	Configure Implementation	7-10
7.6.4	Configure Notifications	7-11
7.7	Use Case: Event-Based Framework Execution in Real-time Transaction Monitoring	7-13

A Appendix A - Distributed Activation Manager Deployment

B Appendix B: Set Up Event Framework Metadata

B.1	Prerequisites	B-1
B.2	Set Up Topics	B-1
B.2.1	Create a Topic in the Kafka Server	B-1
B.2.2	Add Topic Metadata Details	B-2
B.2.3	Activate Consumer Group	B-2
B.2.4	Deactivate Consumer Group	B-3
B.2.5	Decrease Consumers	B-3
B.2.6	Increase Consumers	B-4
B.2.7	Replay Messages	B-4
B.2.8	Save Consumer Properties	B-5
B.2.9	Save Producer Properties	B-5
B.2.10	Start Consumers	B-6
B.2.11	Stop Consumers	B-6

Index

List of Figures

7-1	The Event-Based Process Flow Workflow	7-1
7-2	The Process Details Drawer Window	7-4
7-3	A Simple Representation of Event Process Flow on the PMF Canvas	7-5
7-4	The Producer Activity Tab	7-6
7-5	The Transition Tab	7-7
7-6	The Notifications Tab	7-8
7-7	The Consumer Activity Tab	7-9
7-8	The Consumer Tab	7-10
7-9	The Implementation Tab	7-11
7-10	The Notifications Tab	7-12
7-11	Use Case Flow Chart for Transaction Monitoring	7-14
7-12	Process Monitoring Window Showing Execution Path	7-15
7-13	Process Monitoring Window Showing Lineage of Data Flow	7-16
A-1	Illustration of Distributed Activation Manager Deployment	A-1

List of Tables

2-1	New features in the OFSAAAI Application Pack Release 8.0.8.0.0	2-2
3-1	User Role Mapping for PMF	3-2
7-1	Argument Description Table for Event Subtype Tables	7-3
B-1	Argument Description Table for Topic Metadata	B-2
B-2	Argument Description Table for Activate Consumer Group	B-2
B-3	Argument Description Table for Deactivate Consumer Group	B-3
B-4	Argument Description Table for Decrease Consumers	B-3
B-5	Argument Description Table for Increase Consumers	B-4
B-6	Argument Description Table for Replay Messages	B-4
B-7	Argument Description Table for Save Consumer Properties	B-5
B-8	Argument Description Table for Save Producer Properties	B-6
B-9	Argument Description Table for Start Consumers	B-6
B-10	Argument Description Table for Stop Consumers	B-7

1

Preface

The Oracle Financial Services Analytical Applications Infrastructure Extension Pack (OFS AAIEP) offers advanced features and functionalities which can be installed on top of Oracle Financial Services Analytical Applications Infrastructure (OFS AAI).

The information contained in this document is intended to give you an exposure and an understanding of the features in OFS AAIEP.

- [About this Manual](#)
- [Audience](#)
- [Recommended Skills](#)
- [Recommended Environment](#)
- [Prerequisites](#)

1.1 About this Manual

This guide explains the functionalities of OFS AAIEP in a procedural approach. The OFS AAIEP works on top of the Oracle Financial Services Analytical Applications Infrastructure (OFS AAI) with additional licenses enabled.

The information in this document pertains to the use of the OFS AAIEP with separate sections that for features such as Distributed Processing Capabilities, Analytic Pipeline and Process Models, Content Management Interoperability Services, and Event Based Orchestration.

1.2 Audience

This guide is intended for:

- Business Analysts who are instrumental in solution designing and creation of statistical models using Historical Data.
- System Administrators (SA) who are instrumental in maintaining and executing batches, making the Infrastructure Application secure and operational, and configuring the users and security of Infrastructure.

1.3 Recommended Skills

- System Administrators needs to be aware of the database concepts and the underlying database structure of the Infrastructure Application from an operational perspective. System Administrators also need to be technically sound in configuring the databases for data extraction procedures.
- Business Analysts must have an in-depth knowledge of the underlying data sources that store organization's data, the ETL concept of Data Warehousing and associated terminologies along with the statistical techniques for Model Designing and Execution.

1.4 Recommended Environment

For best viewing of Infrastructure Pages, set the window resolution to a minimum resolution of 1024 x 768 pixels.

For a list of compatible browsers, see the [Oracle Financial Services Analytical Applications 8.0.8.0.0 Technology Matrix](#).

1.5 Prerequisites

- Successful installation of Infrastructure and related software.
- Application of Additional Licenses for OFS AAIEP
- Good understanding of business needs and administration responsibilities.
- In-depth working knowledge of business statistics.

2

Overview of OFS AAIEP

You can define and orchestrate Business Process Pipelines only if the Oracle Financial Services Analytical Applications Infrastructure Extension Pack (OFSAAIE) is enabled in the OFS AAI setup.

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

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

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](#)
- [What is New in this Release of OFS AAIEP](#)

2.1 What is New in this Release of OFS AAIEP

This section lists new features, enhancements, and other updates in the OFS AAIEP for Release 8.0.8.0.0.

- [New Features in Release 8.0.8.0.0](#)
- [Deprecated Features](#)
- [Desupported Features](#)

2.1.1 New Features in Release 8.0.8.0.0

This section lists the new features described in this User Guide.

Table 2-1 New features in the OFSAAI Application Pack Release 8.0.8.0.0

Feature	Description
Distributed Processing Capabilities	
Analytic Pipeline and Process Models	
Content Management Interoperability Services	
Event-Based Orchestration	

For more details, see the [Oracle Financial Services Analytical Applications Infrastructure Extension Pack v8.0.8.x Readme](#).

2.1.2 Deprecated Features

There are no Deprecated Features in this release.

2.1.3 Desupported Features

There are no Desupported Features in this release.

3

Categorization of Pipelines

The Categorization of Pipelines creates a clear segregation of the nature of tasks being orchestrated.

For details about PMF, see the [Oracle Financial Services Analytical Applications Infrastructure 8.0.8.0.0 Process Modelling Framework Orchestration Guide](#).

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

1. Workflow Pipeline
2. Run Pipeline
3. Sub-Run Pipeline
4. Business Process Pipeline

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

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

1. **Run Pipeline:** If any orchestrated pipeline consists of OFSAA Tasks and Service Calls that run within the OFSAA Context, it is categorized as Run Pipeline and it allows to stitch or orchestrate other processes of the type "subrunprocess". The Sub-run Process is further explained, or the Run can be mapped within a Run, which allows you to configure an array of tasks and orchestrate them based on the decisions attached.
 2. **Workflow:** This represents a state-machine pipeline that allows you to orchestrate manual and automated system tasks, which helps you to stitch a state-change machine by performing manual or system tasks. If there is a Sub-pipeline, then you can stitch a Workflow pipeline. This process also allows you to have service tasks that make REST calls to systems outside of OFSAA.
 3. **Subrunprocess:** This is a process similar to the Run process where a Sub-run Process Pipeline allows you to orchestrate the tasks in OFSAA as well as the Services and APIs that run within the OFSAA Context. You can also map other processes of the type "subrunprocess" within a Sub-run Process.
 4. **Business Process:** This process allows you to access a full range of orchestration where the Business Process Pipeline can interleave a Run Pipeline to a Workflow Pipeline or can orchestrate a Sub-Run process or Workflow process. It enables the Widget to have a full range of tasks in OFSAA as well as workflow tasks. This process also allows you to have service tasks that make REST calls to systems outside of OFSAA.
- [User Role Mapping and Access Rights](#)

3.1 User Role Mapping and Access Rights

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

The following user role mapping is mandatory for PMF:

Table 3-1 User Role Mapping 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.
WFKAFKA	Kafka Producer Consumer	Assign this role to the users who are required to perform the Event-based Orchestration of Process Flow. NOTE: The mapping of this Role displays the following icons in the PMF Canvas: <ul style="list-style-type: none"> · Event Producer · Event Consumer The messages sent by Banks to Topics to trigger the PMF Workflow is consumed by the Event Consumer. The messages from the PMF Workflow to the Banks (Topics) is sent by the Event Producer.
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.

Table 3-1 (Cont.) User Role Mapping for PMF

Role Code	Role Name	Functionality
WFACCNEXE	Workflow Execute On Read	Assign this Role to Users who have to Execute, Re-run, or Abort a Process. PREREQUISITE: Ensure that the Users are assigned the Workflow Access (WFACC) Role before you assign this Role.

 **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.

4

Access OFS AAIEP

The OFS AAIEP Features are available through the OFSAA Login.

For how to login, see the [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

Additionally, you may want to see the the [Oracle Financial Services Analytical Applications Infrastructure 8.0.8.0.0 Process Modelling Framework Orchestration Guide](#) for PMF user assistance.

5

Distributed Processing Capabilities

The OFS AAIEP provides Distributed Processing capabilities (includes parallel processing), which can help you run the Application on many processors concurrently.

- [Distributed Activation Manager \(AM\) Based Processing](#)

5.1 Distributed Activation Manager (AM) Based Processing

The Distributed AM-based Processing Feature allows you to configure AM Engines to run on multiple OFSAA nodes and then you can configure ICC Batch Tasks to be distributed across AM Engines on multiple nodes to enable distributed or parallel task executions. Distributed AM-based Processing is achieved in OFSAA by using the following mechanisms.

1. Configuring OFSAA Processing Tier through Load Balancer where Batch Tasks are distributed across multiple AM Nodes
2. Manually configuring Batch Tasks to run on specific AM Nodes

For the mechanisms mentioned previously, you must configure the Secondary AM Server.

For details, see the [Set Up Secondary AM Server](#) Section.



Note:

For an illustration of the Distributed Activation Manager Deployment, see the A - Distributed Activation Manager Deployment Appendix.

- [Set Up Secondary AM Server](#)
- [Configure the OFSAA Instance through Load Balancer to Distribute Batch Tasks on Multiple AM Nodes](#)
- [Execute Batches on Multiple AM Nodes](#)

5.1.1 Set Up Secondary AM Server

The following is the prerequisite to Set up the Secondary AM Server.

1. For information on the hardware and software requirements to set up of the Secondary AM Server, see the Hardware and Software Requirements Section in the [Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack Installation Guide](#).
2. Execute the following SQL Script on the Configuration Schema by replacing <NEWAMIPADDRESS> with the IP Address or Hostname of the new AM Node that you want to set up. Replace <EXISTINGAMIPADDRESS> with the IP Address or IP Address of the existing AM Server.

```

INSERT INTO ficssystemmaster

(WEBIPADDRESS, APPIPADDRESS, DBIPADDRESS, ETLAPPHOME, NOOFCPUS, VMEMORY, P
MEMORY, CACHE, NOOFTHEADS, IOTRANSFER, MAXTRANSPERSEC, DISKSDBSTRIPING, D
ISKSFILESTRIPING, MAXFILESIZE, MAXFTPFILESIZE, MAXFILENAMELEN, OSDATABLO
CKSIZE, DATASETTYPE, STAGEPATH, DBFTPSHARE, DBFTPUSERID, DBFTPPASSWD, DBFT
PPOINT, DBFTPDRIVE, APPFTPSHARE, APPFTPPOINT, APPFTPDRIVE, APPFTPUSERID, APP
FTPPASSWD, WEBFTPSHARE, WEBFTPPOINT, WEBFTPUSERID, WEBFTPDRIVE, WEBFTPPASS
WD, OSTYPE, SOCKETSERVERPORT, SEC_SHARE_NAME, SEC_USERID, SEC_PASSWD, F_IS
PRIMARY, N_PRECEDENCE)

SELECT

WEBIPADDRESS, APPIPADDRESS, '<NEWAMIPADDRESS>', ETLAPPHOME, NOOFCPUS, VME
MEMORY, PMEMORY, CACHE, NOOFTHEADS, IOTRANSFER, MAXTRANSPERSEC, DISKSDBSTRIP
ING, DISKSFILESTRIPING, MAXFILESIZE, MAXFTPFILESIZE, MAXFILENAMELEN, OSD
ATABLOCKSIZE, DATASETTYPE, STAGEPATH, DBFTPSHARE, DBFTPUSERID, DBFTPPASSW
D, DBFTPPOINT, DBFTPDRIVE, APPFTPSHARE, APPFTPPOINT, APPFTPDRIVE, APPFTPUSER
ID, APPFTPPASSWD, WEBFTPSHARE, WEBFTPPOINT, WEBFTPUSERID, WEBFTPDRIVE, WEBF
TPPASSWD, OSTYPE, SOCKETSERVERPORT, SEC_SHARE_NAME, SEC_USERID, SEC_PASSW
D, F_ISPRIMARY, N_PRECEDENCE

FROM ficssystemmaster

WHERE DBIPADDRESS='<EXISTINGAMIPADDRESS>'

```

To set the newly added AM Node as the Primary Node, execute the following SQL Script by replacing <NEWAMIPADDRESS> with the IP Address or Hostname of the newly added AM Node:

```

UPDATE FICSSYSTEMMASTER SET F_ISPRIMARY = 'Y', N_PRECEDENCE=200 WHERE
DBIPADDRESS = '<NEWAMIPADDRESS>'

```

3. Update the IS_DISTRIBUTED_AM_SUPPORTED Parameter in the **Configuration** Table.

To Set up the Secondary AM Servers, follow these steps:

1. Copy the following directories to the Secondary AM Server from the Primary OFSAA Server:
 - \$FIC_HOME/conf
 - Entire ficdb and its sub-directories
 - .profile File from \$HOME Directory of the Primary OFSAA Server
2. Perform the following configurations in the Secondary AM Server:
 - Set the FIC_HOME variable to the Directory which is the OFSAA Home (must contain both the ficdb and conf directories) directory.

 **Note:**

Oracle recommends that you set up the OFSAA Secondary AM under the same user as in the Primary Server.
For example, if OFSAA is installed on the Primary Server under `/scratch/ofsausr`, you can setup the secondary OFSAA Instance in `/scratch/ofsausr` user.

3. Set the `FIC_DB_HOME` Variable to the Directory where the `/ficdb` Directory is copied in the Secondary AM Server.

```
AM_HOME=$FIC_HOME/ficdb

export AM_HOME

AM_CONF_FILE=$FIC_DB_HOME/conf/am.conf

export AM_CONF_FILE

FICTEMP=$FIC_DB_HOME/conf

export FICTEMP
```

4. Ensure that the following variables are pointed to the valid Hostname or IP Address on which the Message Server, Router Server, and Router Engines are running.

```
MESSAGE_SERVER_HOST=10.XXX.XXX.XXX

export MESSAGE_SERVER_HOST

MESSAGE_SERVER_PORT=6666

export MESSAGE_SERVER_PORT

FIC_ROUTER_HOST=10.XXX.XXX.XXX

export FIC_ROUTER_HOST

FIC_ROUTER_PORT=7777

export FIC_ROUTER_PORT
```

5. Set the `JARPATH` Variable to the `$FIC_DB_HOME/lib` Directory.
6. Ensure that the `ORACLE_SID` Variable is pointed to the correct Oracle Instance and users can successfully connect to this instance from the Secondary AM Server using SQL Plus.
7. Update the Secondary AM Node details in the `AM.conf` File present in the `$FIC_HOME/ficdb/conf` Directory.

```
<AM_HOST>`<Secondary AM node host name/IP Address>`

<AM_PORT>`<Secondary AM node Port number>`
```

 **Note:**

Do not alter the <ROUTER_NAME> and <ROUTER_PORT> Tag values.

8. Modify the logger XML files such as MFLogger.xml, OFSAALogger.xml, DQLogger.xml, and PR2Logger.xml, which are available in the \$FIC_DB_HOME/conf Directory with the Secondary AM Server \$FIC_HOME Directory path.

5.1.2 Configure the OFSAA Instance through Load Balancer to Distribute Batch Tasks on Multiple AM Nodes

See the **Configuring OFSAA Load Balancer** Section in the [Configuration for High Availability \(HA\) Best Practices Guide](#) for details on how to configure the Load Balancer.

 **Note:**

The Message Server must be running in all the nodes where the AM Servers are configured.

5.1.3 Execute Batches on Multiple AM Nodes

While defining a Task in a Batch from the **Task Definition** Window in the Operations Module, you can choose on which node each task needs to be executed. The **Primary IP for Runtime Processes** drop-down list in the **New Task Definition** Window displays all the registered AM Server Nodes. Select the IP Address of the AM Node where you want the task to be executed.

For more information on how to define a Batch, see the [Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack Installation Guide](#).

 **Note:**

OFS AAIEP supports the crash handling of Backend Servers. For more information, see the **Crash Handling of Backend Servers** Section in the [Oracle Financial Services Analytical Applications Infrastructure User Guide](#).

6

Content Management Integration

The Content Management Interoperability Services (CMIS) is an OASIS standard enabling information sharing between different Content Management Systems. Document management within OFSAA can integrate with CMIS services to support document upload and download to the CMIS repository.



Note:

To use the features explained in this section, additional licenses may apply. For details, contact [My Oracle Support](#).

Perform the following configurations:

1. Set the following parameters in the Configuration Table in the Config Schema to enable CMIS:
 - a. Set the value of `IS_CMIS_ENABLED` Parameter to TRUE. If this is set to FALSE, the document upload will occur on ftpshare.
 - b. Update the value of `CMIS_ATOMPUB_URL` Parameter with the repository URL. Ensure the URL is up and running.
For example: `http://192.0.2.1:7777/service/cmisis`
2. Modify the `INFODOM_cmis.properties` Property File, which is available in the `$FIC_HOME/ficweb/webroot/conf` Directory.
 - a. Rename the file by replacing the INFODOM with actual name of the Infodom. For example if the Infodom name is "OFSAAINFO", rename the file to `OFSAAINFO_cmis.properties`.
 - b. The property file will contain the following entries. Update them as per the CMIS URL.

```
REPOSITORY_ID=5  
  
USER=admin  
  
PASSWORD=password  
  
DEFAULTPATH=/Default  
  
DOC_OBJ_TYPE_ID=cmis:document  
  
FLDR_OBJ_TYPE_ID=cmis:folder
```

3. Redeploy the application onto your configured Web Application Server.

For more information on generating and deploying the EAR/WAR file, see the Post Installation Configuration Section in the [Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack Installation Guide](#).

4. Restart all the OFSAAI Services.
For more information, see the Start Infrastructure Services and Stop Infrastructure Services Sections in the [Oracle Financial Services Advanced Analytical Applications Infrastructure Application Pack Installation Guide](#).
- [Configurations for Document Upload to Multiple Libraries](#)

6.1 Configurations for Document Upload to Multiple Libraries

You can upload documents to multiple libraries instead of a single library, which allows to control within the threshold the number of documents within each library.

Enter the values for the following parameters in the AAI_CMIS_REPO_MASTER Table in the Config Schema as shown in the following:

- V_REPO_ID - Unique value for identification of Library
- V_REPO_URL - Update the value with the repository URL
- V_DEF_PATH - Update the Directory path
- D_START_DATE - Upload start date for the library/directory
- D_END_DATE - Upload end date for the library/directory
- V_INFODOM - Update Infodom name
- V_CMIS_REPO_ID - Update the value with REPOSITORY_ID (note that this value is case-sensitive)



Note:

For old documents in the DOCUMENT_MASTER Table, only the V_DOC_CMIS_ID Column is updated for CMIS Integrated Uploads. After applying the 80710 ML Patch, you must update the V_REPO_ID Column in the DOCUMENT_MASTER Table with the value updated in the AAI_CMIS_REPO_MASTER.V_REPO_ID Column. You must update for all rows that have the value in the V_DOC_CMIS_ID Column of the DOCUMENT_MASTER Table.

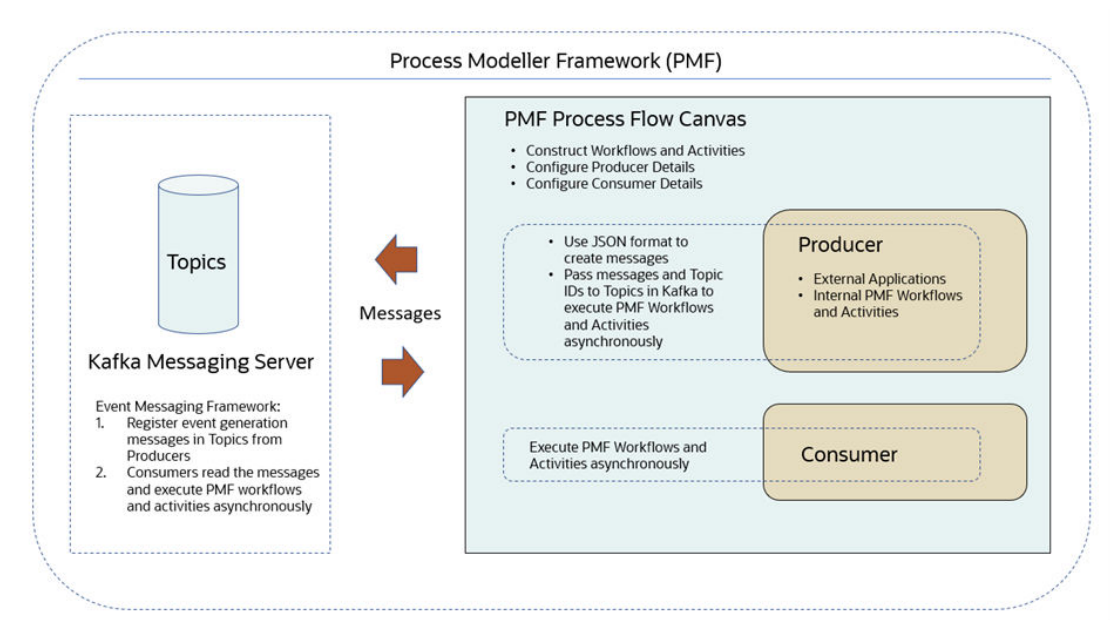
7

Event-Based Orchestration of Process Flow

The Event Framework integrated into PMF is an event-driven messaging framework that uses Apache Kafka (messaging server) to process information asynchronously and in near real-time. Producers that are either external applications or PMF internal processes, send messages (notify about an event occurrence) to Kafka Topic(s). Consumers, that are internal processes within PMF, read the messages from Kafka Topic(s) and execute other events or activities configured in the PMF Canvas. The framework uses the REST transfer protocol to send messages and the JSON data format to construct the messages.

The notification of generation of an event to a Topic can execute a workflow or any individual task asynchronously. Previously, the orchestration of workflows in PMF followed a synchronous pattern and executed the activities in the order of sequence configured. In other words, the execution or completion of activity was required before another activity in the workflow could be executed. However, with the integration of the Event Framework, in addition to the existing sequential workflow, you can now also execute asynchronously not only the workflows constructed in the PMF canvas but also individual activities within the workflows and in any order of the desired configuration.

Figure 7-1 The Event-Based Process Flow Workflow



- [Producer Activity](#)
- [Consumer Activity](#)
- [Seed Data in the Event and Event Subtype Tables](#)
- [Create an Event Process Flow in the Process Modeller](#)

- [Configure Event Producer](#)
- [Configure Event Consumer](#)
- [Use Case: Event-Based Framework Execution in Real-time Transaction Monitoring](#)

7.1 Producer Activity

The Event Framework enables the Producers to create messages as per the JSON definition and invokes the Producer API by passing the message and the Topic ID. This message is recorded in the Event Framework Audit Table for reference purposes.

The Producer and Consumer of messages are mapped to the same Topic (a unique identifier for a data stream in Kafka) and the exchange of data takes place (In PMF, execution of Process Flows and Activities).

See the [JSON Definition for Events](#) Section in the Oracle Financial Services Analytical Applications Infrastructure Process Modelling Framework Orchestration Guide for information about JSON definition formats and examples.

7.2 Consumer Activity

The Event Framework enables the Consumer (internal processes in PMF) to make asynchronous requests to execute activities in a process flow based on the conditions of event occurrences. The messaging service can also be used by external applications to post messages to Topics in Kafka to execute events in individual components or execute the whole process asynchronously. In other words, activities do not have to wait for other activities to be completed (as in a synchronous request) and can be messaged to execute events scheduled for the specific activities.

The Producer and the Consumer of the messages routed through the Kafka messaging server are mapped to the same topic (mapped by Producer ID, Topic ID, and Consumer ID) for the following metadata types:

- Event Type
- Event Sub Type
- Event Source

7.3 Seed Data in the Event and Event Subtype Tables

The Event and Event Subtype Tables require the initial population of data (seeding) as a configuration step.

 **Note:**

Before you can use the Event-based Orchestration of Process Flow, seed the data as discussed in this section. The seeding of data is a prerequisite configuration.

To Seed Data in the Event and Event Subtype tables, perform the following:

- Configure the arguments and Run the following script to add data to the Event Tables in the AAI Config Schema:

```
./SaveEventInfo.sh <EVENT_TYPE> <EVENT_SUB_TYPE> <EVENT_SOURCE>
<EVENT_DESCRIPTION> <LOCALE_CODE>
```

Table 7-1 Argument Description Table for Event Subtype Tables

Argument	Description
EVENT_TYPE	The type of classification at the Parent level for the various tasks. For example: The type of product such as Insurance, loans, and Deposits.
EVENT_SUB_TYPE	The further classification of the Parent level Event Type. For example: For the Loan Event Type, the subtypes such as long-term loans, short-term loans, and loan transfers.
EVENT_SOURCE	The Source of the OFSAA Application for the Event to be added.
EVENT_DESCRIPTION	The description of the Event that is to be added.
LOCALE_CODE	The locale for the UI.

For example:

```
./SaveEventInfo.sh ECMEVENT AML_DD ECM 'ECM_Type_EVENT' en_US
```

 **Note:**

For more information about additional configurations related to Event Framework Metadata, see [Appendix B: Set Up Event Framework Metadata](#).

7.4 Create an Event Process Flow in the Process Modeller

Use the Activities in the PMF Canvas to create an Event Workflow and execute it so that the process of communication of data between the Producers (external applications or PMF internal processes) and consumers (PMF internal processes) is established and the events are triggered from the PMF Canvas. Configure to execute each task independently as events in the process flow.

To create an Event Process Flow, follow these steps:

1. Log in to the OFSAA Application.
2. Click the **Administration** icon from the **Header** to display the administration tasks in the **Tiles** menu.
3. Select the required **Information Domain** from the drop-down list.
4. Select **Process Modelling Framework** to display a submenu.
5. Select **Process Modeller** from the submenu to display the **Process Modeller** window.

- Click the **Add** icon in the window to display the **Process Details** drawer window.

Figure 7-2 The Process Details Drawer Window

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

- Process ID:** 1612169930424 (read-only)
- Process Name:** Event Process Flow Demo
- Process Description:** A demo to show the Event Process Flow
- App Package ID:** Platform (dropdown)
- Type:** Workflow Pipeline (dropdown)
- Registered Topics:** (empty dropdown)
- Spark DB:** No (dropdown)
- Infodom:** OFSAAIINFO (dropdown)
- Tag:** (empty text field)
- Segment:** CAPRPSEG (dropdown)

A green checkmark icon is visible at the bottom right of the drawer window.

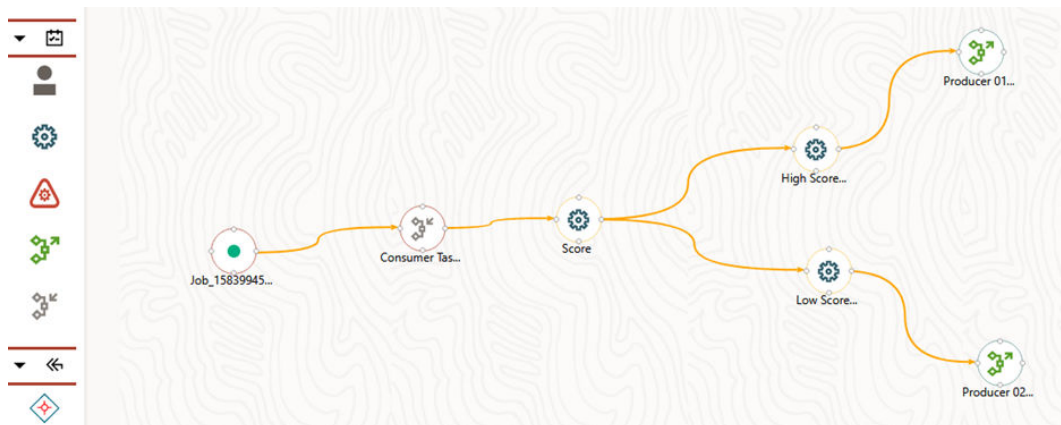
- In the **Process Details** drawer window, enter the required details as follows:
 - Enter a name in the **Process Name** field. The **Process ID** field is created by the application and is read-only.
 - Enter a description for the process in the **Process Description** field.
 - Select **Platform** from the **App Package ID** drop-down.
 - Select **Workflow Pipeline** from the **Type** drop-down.
 - Do not select any value from **Registered Topics** as it does not apply to an event process.
 - Do not select any value from **Spark DB** as it does not apply to an event process.
 - Select the required Infodom from the **Infodom** drop-down.
 - Enter any appropriate tags in the **Tag** field. This field is optional.
 - Select the required application segment from the **Segment** drop-down.
- After entering the details, click the **Accept** icon to save it. The Process Flow is created and is displayed in the PMF Canvas. By default, a job of the type **START** is created on the Canvas.

9. Drag and drop the required activities from the **Activity** menu into the Canvas to construct the Process Flow.
10. Drag and drop **Event Producer** and **Event Consumer** activities into the Canvas.
11. Click the **Save** icon to save the process definition.

The following illustration provides a simple example of the Event Process Flow with Producer and Consumer activities displayed on the Canvas.

For more information about configuring Consumers and Producers, see the [Configure an Event Consumer](#) and [Configure an Event Producer](#) sections.

Figure 7-3 A Simple Representation of Event Process Flow on the PMF Canvas



7.5 Configure Event Producer

To configure an Event Producer, follow these steps:

1. Click an **Event Process Flow** in the **Process Modeller** window to display it in the PMF Canvas.
2. Double-click the **Event Producer** icon, or click the **Edit Activity** icon that displays when you mouse-over the activity, to display the activity configuration details in the **Producer Activity** drawer window.

The window displays three tabs: **Activity**, **Transition**, and **Notifications**. The Activity tab requires mandatory details, whereas the Transition and Notifications tabs are optional.

- [Configure Activity](#)
- [Configure Transition](#)
- [Configure Notifications](#)

7.5.1 Configure Activity

To configure Activity details, follow these steps:

1. Enter a name for the Producer Event in the **Activity Name** field.

Figure 7-4 The Producer Activity Tab

The screenshot shows the 'Producer ECM' configuration interface. It features a dark sidebar on the left with icons for a calendar, back, and notifications. The main content area is titled 'Producer ECM' and contains several input fields and dropdown menus. The fields are: 'Activity Name' (text input, value: 'Producer ECM'), 'Activity Desc' (text input, value: 'EOD Balances for Current'), 'Activity Type' (read-only text, value: 'Event Producer'), 'Status' (dropdown menu), 'Exclude Task' (dropdown menu, value: 'No'), 'Producer Properties' section containing: 'Message Datafield' (dropdown menu, value: 'WF_OUT_MSG'), 'Topics' (text input, value: 'ECM_TF_SWIFT'), 'Event Type' (dropdown menu, value: 'ECMEVENT'), 'Event Sub Type' (dropdown menu, value: 'AML_DD'), 'Source' (dropdown menu, value: 'ECM'), and 'Event Description' (read-only text, value: 'ECM_Type_EVENT'). A green checkmark icon is located at the bottom right of the form.

2. Enter a description for the Producer Event in the **Activity Description** field. By default, the **Activity Type** field displays as Event Producer and is read-only.
3. Select the appropriate status from the **Status** drop-down.
4. Select **Yes** or **No** from the **Exclude Task** drop-down.
5. In the **Producer Properties** section, select the required type of message for the metadata from the **Message Datafield** drop-down.
6. Select the required topics from the **Topics** drop-down.
7. Select the required event type from the **Event Type** drop-down.
8. Select the required event subtype from the **Event Sub Type** drop-down.
9. Select the required source OFSAA application from the **Source** drop-down. The **Event Description** field is populated based on the selection in the Event Sub Type drop-down.
10. After entering the details, click the **Accept** icon to save the details.

7.5.2 Configure Transition

To configure Transition details, follow these steps:

1. Click the **Transition** tab.
2. Click the **Add** icon to display the fields required for transition details.
3. Select the required job to be connected to from the **Connected To** drop-down.

Figure 7-5 The Transition Tab

Add New Transition from Node JOB_16121854413363

Connected To

Transition Name
JOB_16121854413363_JOB_16121854413363

Order
1

Decision Rule
Outcome Approve

Stroke
Normal

4. Enter a name for the transition in the **Transition Name** field.
5. Enter the order of execution in the **Order** field. For example, 1 marks the order as to be executed first.
6. Click the **Search** icon from the **Decision Rule** drop-down and select the required application rule. Click the the **Accept** icon to save the details.
7. Select the stroke format from the **Stroke** drop-down.
8. After entering the details, click the **Accept** icon to save the entered details.

7.5.3 Configure Notifications

To configure details in Notifications, follow these steps:

1. Click the **Notifications** tab.
2. Click the **Add** icon to display the fields required for the Notifications.
3. Enter a name for the notification in the **Notification Name** field. The **Notification ID** field is auto-populated based on the ID from the application.

Figure 7-6 The Notifications Tab

Activity Details > Notification Group

Notification ID
JOB_1612186040060

Notification name
ECM

Condition
Default

Generate
On Entry

Enabled Yes No

Email

Email Required Yes No

Email Template
Global Task Template

✓

4. Click the **Search** icon from the **Condition** drop-down and select the required application rule. Click the **Accept** icon to save the entered details.
5. Select the condition to generate the notification from the **Generate** drop-down.
6. Select **Yes** in the **Enabled** field to activate the notification.
7. In the **Email** section, select **Yes** in the **Email Required** field to select to receive the notification through email. Select **No** to disable this option.
8. Select the required email template from the **Email Template** field.
9. Click the **Accept** icon to save the entered details.

7.6 Configure Event Consumer

To configure an Event Consumer, follow these steps:

1. Click an Event Process Flow in the **Process Modeller** window to display it in the PMF Canvas.
2. Double-click the **Event Consumer** activity icon, or click the **Edit Activity** icon that displays when you mouse-over the activity, to display the activity configuration details in the **Producer Activity** drawer window.

The window displays four tabs: **Activity**, **Consumer**, **Implementation**, and **Notifications**. The details in the Notification tab are optional.

- [Configure Activity](#)
- [Configure Consumer](#)
- [Configure Implementation](#)
- [Configure Notifications](#)

7.6.1 Configure Activity

To configure details in the Activity tab, follow these steps:

1. Enter a name for the **Consumer Event** in the **Activity Name** field.

Figure 7-7 The Consumer Activity Tab

The screenshot shows a configuration window for a consumer activity. The title bar at the top is red and contains the text 'JOB_16121854413374'. On the left side, there is a dark vertical sidebar with several icons. The main content area is white and contains the following fields:

- Activity Name:** A text input field containing 'JOB_16121854413374'.
- Activity Desc:** A text input field containing 'Consumer Activity'.
- Activity Type:** A read-only text field containing 'Event Consumer'.
- Status:** A dropdown menu with a downward arrow.
- Exclude Task:** A dropdown menu with 'No' selected and a downward arrow.

A green checkmark icon is located at the bottom right of the configuration area.

2. Enter a description for the **Consumer Event** in the **Activity Description** field. By default, the **Activity Type** field displays as Event Consumer and is read-only.
3. Select the appropriate status from the **Status** drop-down.
4. Select **Yes** or **No** from the **Exclude Task** drop-down.

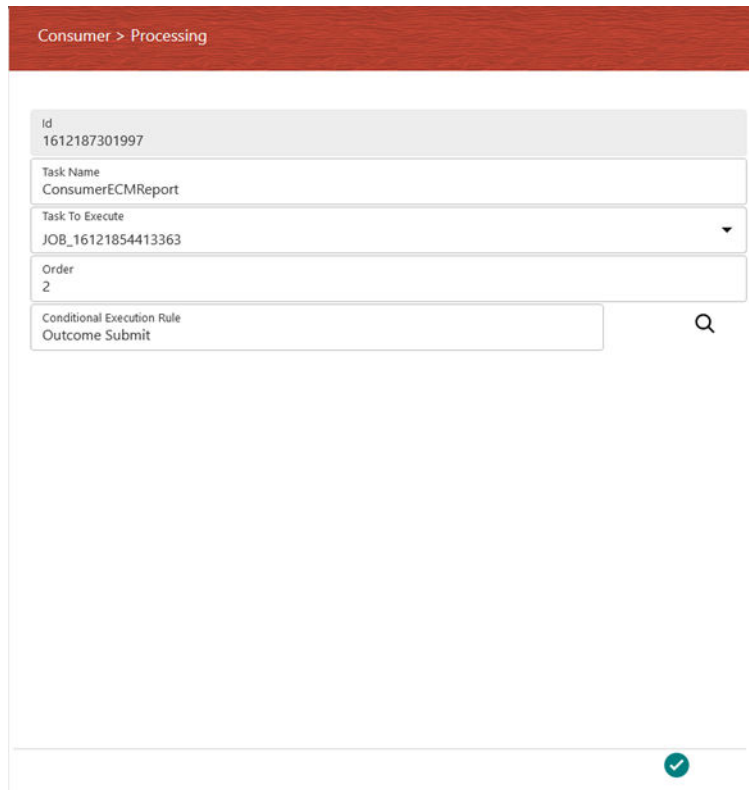
5. After entering the details, click the **Accept** icon to save the entered details.

7.6.2 Configure Consumer

To configure details in the Consumer tab, follow these steps:

1. Enter a name for the consumer task in the **Task Name** field. By default, the **ID** field displays the unique identifier for the consumer task.

Figure 7-8 The Consumer Tab



The screenshot shows a web interface for configuring a consumer task. At the top, there is a red header bar with the text "Consumer > Processing". Below this, the form contains several fields: "Id" with the value "1612187301997", "Task Name" with the value "ConsumerECMReport", "Task To Execute" with a dropdown menu showing "JOB_16121854413363", "Order" with the value "2", and "Conditional Execution Rule" with the value "Outcome Submit" and a search icon. A green checkmark icon is visible at the bottom right of the form area.

2. Select the appropriate jobs from the **Tasks To Execute** drop-down.
3. Enter the order of execution in the **Order** field. For example, 1 marks the order as to be executed first.
4. Click the **Search** icon from the **Conditional Execution Rule** drop-down and select the required application rule. Click the **Accept** icon to save the entered details.

7.6.3 Configure Implementation

To configure details in the Implementation tab, follow these steps:

1. Click the **Implementation** tab to select the rules required for the execution.

Figure 7-9 The Implementation Tab

JOB_16121854413374

Rule

Infodom
OFSAAIINFO

Execution Rule
None

Parameters

<u>/F_TASK_RESPONSE</u>	Datafield WF_TASK_RESPONSE Type DYNAMIC	Value	

Pre/Post Processing

Pre Rule
None

Post Rule
None

2. Select the required infodom from the **Infodom** drop-down.
3. Click the **Add** icon to display the popup with fields required for the **Parameters** section.
4. Select the required job to be connected to from the **Connected To** drop-down.
5. Select the required data field from the **Data Fields** drop-down.
6. Select the required parameter type from the **Parameter Type** drop-down. The options are **Dynamic** and **Static**. If you select Static, then enter the value in the **Value** field. Click the **Accept** icon to save the entered details.
7. In the Pre/Post Processing section, click the **Search** icon from the **Pre Rule** drop-down and select the required application rule to be executed before the processing of the event. Click the **Accept** icon to save the entered details.
8. Click the **Search** icon from the **Post Rule** drop-down and select the required application rule to be executed after the processing of the event. Click the **Accept** icon to save the entered details.
9. After entering the details, click the **Accept** icon to save the entered details.

7.6.4 Configure Notifications

To configure details in the Notifications tab, follow these steps:


1. Click the **Notification** tab.


Figure 7-10 The Notifications Tab

Activity Details > Notification Group

Notification ID
JOB_1612186040060

Notification name
ECM


Condition
Default 


Generate
On Entry 

Enabled Yes No

Email

Email Required Yes No

Email Template
Global Task Template 



2. Click **Add** icon to display the fields required for the Notifications.
3. Enter a name for the notification in the **Notification Name** field. The **Notification ID** field is auto-populated based on the ID from the application.
4. Click the **Search** icon from the **Condition** drop-down and select the required application rule. Click the **Accept** icon to save the entered details.
5. Select the condition to generate the notification from the **Generate** drop-down.
6. Select **Yes** in the **Enabled** field to activate the notification.
7. In the **Email** section, select **Yes** in the **Email Required** field to select to receive the notification through email. Select **No** to disable this option.
8. Select the required email template from the **Email Template** field.
9. Click the **Accept** icon to save the entered details.

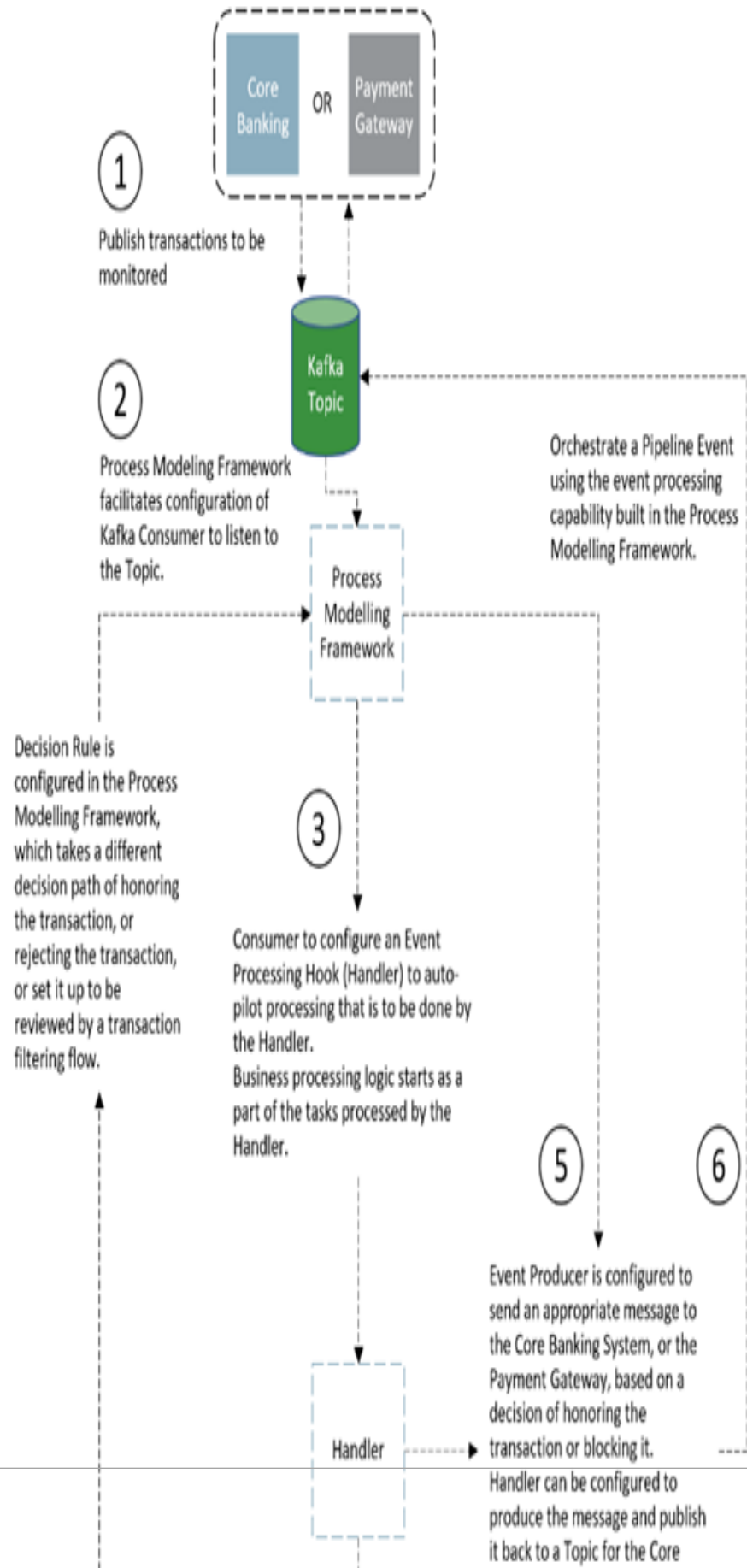
7.7 Use Case: Event-Based Framework Execution in Real-time Transaction Monitoring

The operational mechanism of the Event Processing Pipeline is better understood with a Use Case that describes the execution of real-time Transaction Monitoring.

The following steps describe the Use Case Configuration:

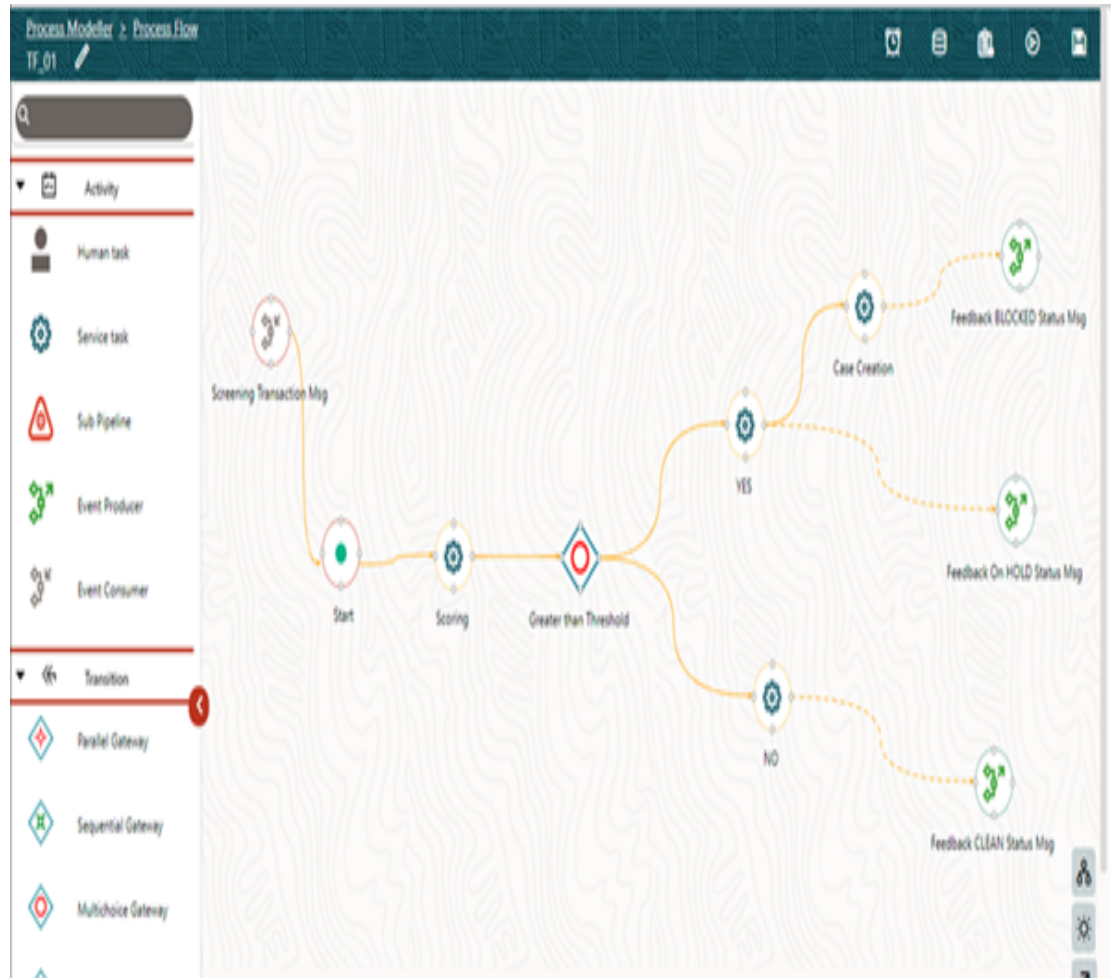
1. The transactions that are to be monitored and honored are published by the Core Banking System, or the Payment Gateway, to a predefined Kafka Topic.
2. The Process Modeling Framework facilitates the configuration of a Kafka Consumer which listens to the Topic.
3. The Consumer allows configuring an Event Processing Hook (Handler) which auto-pilots any further processing that is to be done by the Handler. The Business Processing Logic starts as a part of the tasks processed by the Handler.
4. The Handler computes a complex score based on the various fields that are part of the message.
For example, the transaction amount added to the transaction withdrawal profile set for the customer for a specific look-back period exceeds a threshold and so cannot be honored.
5. A Decision Rule is configured in the Process Modelling Framework, which takes a different decision path of honoring the transaction, or rejecting the transaction, or set it up to be reviewed by a transaction filtering flow.
6. The Event Producer, which shows up further in the flow, is configured to send an appropriate message to the Core Banking System, or the Payment Gateway, based on a decision of honoring the transaction or blocking it. The Handler can be configured to produce the message and publish it back to a Topic for the Core Banking System, or Payment Gateway, to listen.
7. A loosely coupled Pipeline Event can be orchestrated using the event processing capability built in the Process Modelling Framework.

Figure 7-11 Use Case Flow Chart for Transaction Monitoring



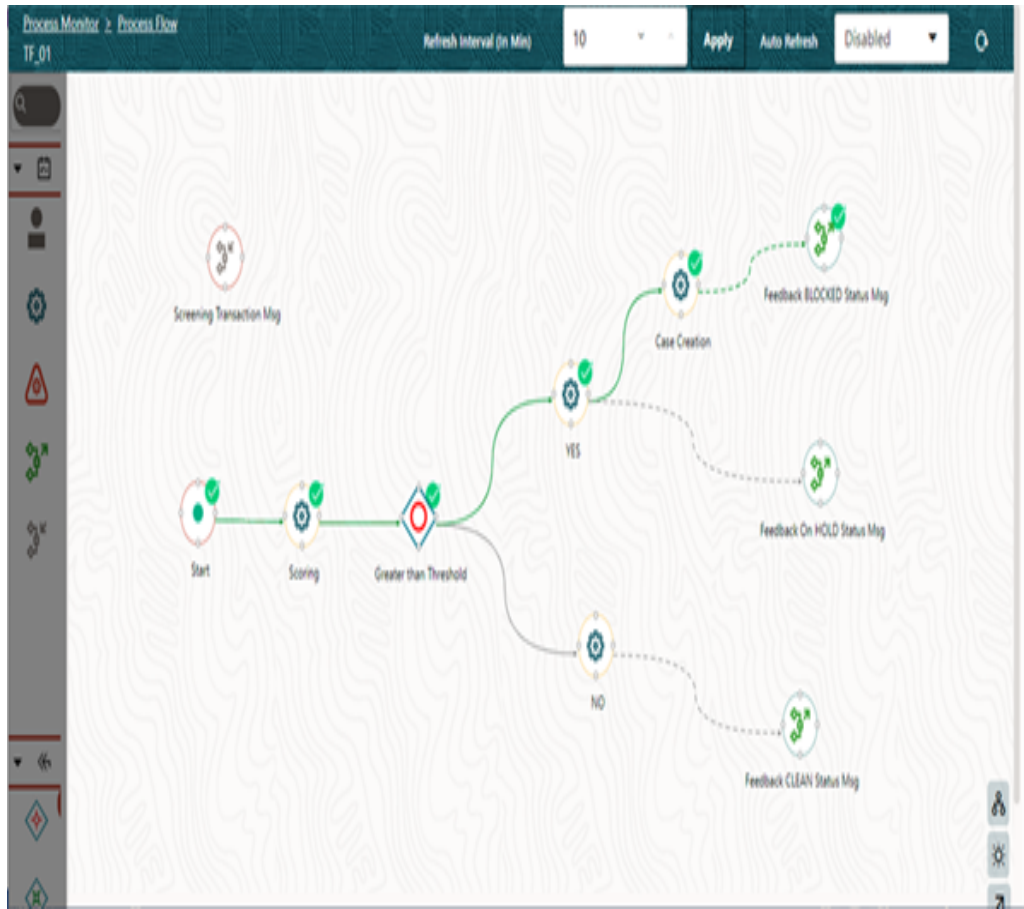
The following illustrations of the Process Monitoring Window provide a visual view of the execution path the message has traveled:

Figure 7-12 Process Monitoring Window Showing Execution Path



The execution path of the transactions and the various data fields that determined the execution path can be monitored using the monitoring capability within PMF. The information shows the lineage of the data flow which has resulted in the transaction being honored or blocked or put on hold.

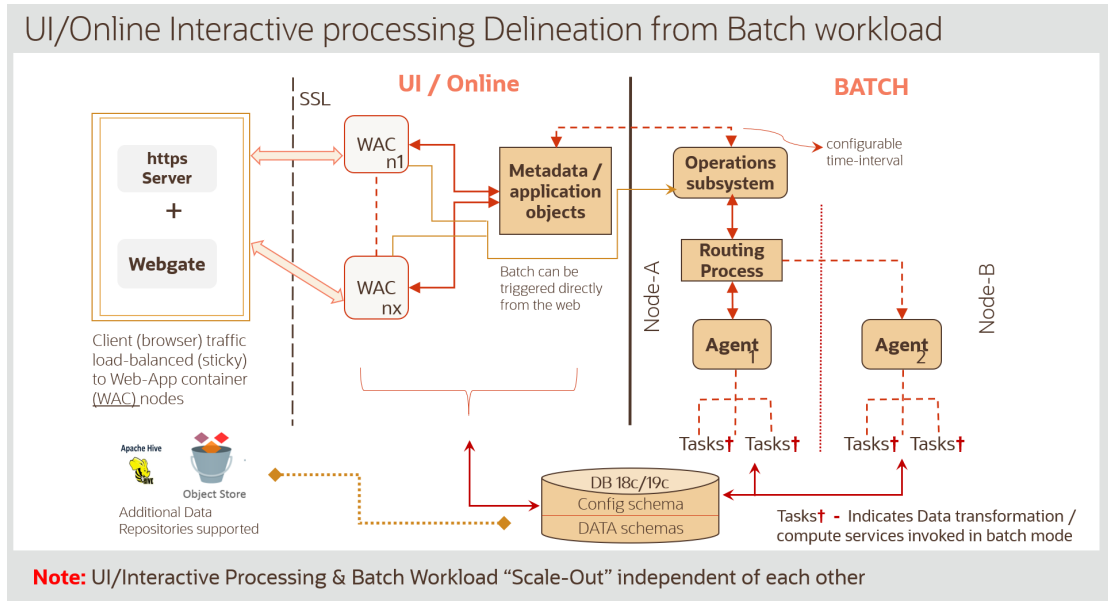
Figure 7-13 Process Monitoring Window Showing Lineage of Data Flow



A

Appendix A - Distributed Activation Manager Deployment

Figure A-1 Illustration of Distributed Activation Manager Deployment



B

Appendix B: Set Up Event Framework Metadata

To configure additional details you must set up the Event Framework Metadata. The sections in this topic provide detailed information on how to set it up.

- [Prerequisites](#)
- [Set Up Topics](#)

B.1 Prerequisites

The Kafka server must be installed and running.

You can download Apache Kafka from the following location: <https://kafka.apache.org/downloads>.

B.2 Set Up Topics

To store and publish data from a Producer writing to a Kafka Topic for Consumers to read the data in the Topic in the OFSAA System certain configurations are required for Topic Metadata details and creation of the Topic in the Kafka Server, which is available in the following sections.

- [Create a Topic in the Kafka Server](#)
- [Add Topic Metadata Details](#)
- [Activate Consumer Group](#)
- [Deactivate Consumer Group](#)
- [Decrease Consumers](#)
- [Increase Consumers](#)
- [Replay Messages](#)
- [Save Consumer Properties](#)
- [Save Producer Properties](#)
- [Start Consumers](#)
- [Stop Consumers](#)

B.2.1 Create a Topic in the Kafka Server

To create a Topic in the Kafka Server, follow this step:

- Execute the `kafka-topics.sh` file (located in the Kafka installation `/bin` directory) using the following syntax:

```
./kafka-topics.sh --create --zookeeper <host of zookeeper:<port in
zookeeper.properties> --replication-factor 1 -partitions <more than
given while creating topic in PMF> --topic <topic_name>
```

For example,

```
./kafka-topics.sh --create --zookeeper 192.0.2.1:7780 --replication-
factor 1 --partitions 4 --topic ECM_TF_SWIFT
```

B.2.2 Add Topic Metadata Details

To add the details for Topic Metadata, perform the following:

- Configure the arguments and Run the script in the following format which adds data to the relevant table in the AAI Config Schema:

```
./saveTopic.sh <TOPIC_NAME> <BOOTSTRAP_SERVERS> <TOTAL_PARTITIONS>
```

Table B-1 Argument Description Table for Topic Metadata

Argument	Description
TOPIC_NAME	The name of the Topic that you want to add. Topics is a virtual group(s) that stores or publishes data. It can have multiple consumers subscribing for the information.
BOOTSTRAP_SERVERS	The details of the server on which Kafka is installed. The Topic listens to this Server.
TOTAL_PARTITIONS	The total number of partitions for the Topic.

For example:

```
./saveTopic.sh ECM_TF_SWIFT 192.0.2.1:7780 4
```

B.2.3 Activate Consumer Group

The script to activate a Consumer Group is used to activate deactivated Consumer Groups. The Consumers are initialized and instantiated at Server startup. Only Consumer Groups that are active are instantiated at Server startup. The activation sets the Active Flag to 'Y' for a given Consumer.

The Consumer ID and Topic ID have to be passed to the Shell Script which requires activation. To activate Consumer Group, perform the following:

- Configure the arguments and run the following script in the Event tables in the AAI Config Schema:

```
./activateconsumergroup.sh <N_CONSUMER_ID> <N_TOPIC_ID>
```

Table B-2 Argument Description Table for Activate Consumer Group

Argument	Description
N_CONSUMER_ID	The Unique Identifier of the Consumer.
N_TOPIC_ID	The Unique Identifier of the Topic.

For example:

```
./activateconsumergroup.sh 1612 SWIFT_TF_01
```

B.2.4 Deactivate Consumer Group

The script to deactivate a Consumer Group is used to deactivate a Consumer Group that is active. The Consumers are initialized and instantiated at Server startup. Only Consumer Groups that are active are instantiated at Server startup. The activation sets the Active Flag to 'N' for a given Consumer. The Consumer Groups that are deactivated are not be instantiated.

The Consumer ID and Topic ID have to be passed to the Shell Script which requires deactivation. To deactivate a Consumer Group, perform the following:

- Configure the arguments and Run the following script in the Event tables in the AAI Config Schema:

```
./deactivateconsumergroup.sh <N_CONSUMER_ID> <N_TOPIC_ID>
```

Table B-3 Argument Description Table for Deactivate Consumer Group

Argument	Description
N_CONSUMER_ID	The Unique Identifier of the Consumer.
N_TOPIC_ID	The Unique Identifier of the Topic.

For example:

```
./deactivateconsumergroup.sh 1612 SWIFT_TF_01
```

B.2.5 Decrease Consumers

This script is used to decrease the number of running Consumers for a given Consumer Group Name, Topic, and Application Node. It applies only till the current application instance is running.

To decrease the number of running Consumers, perform the following:

- Configure the arguments and run the following script in the Event tables in the AAI Config Schema:

```
./consumerdecrease.sh <TOPIC_NAME> <CONSUMERGRPNAM> <DECREASEBY>  
<HOSTNAME:PORT>
```

Table B-4 Argument Description Table for Decrease Consumers

Argument	Description
TOPIC_NAME	The name of the Topic.
CONSUMERGRPNAM	The name of the Consumer's Group.
DECREASEBY	The number of Consumers that you want to decrease.
HOSTNAME:PORT	The details of the server on which Kafka is installed.

For example:

```
./consumerdecrease.sh ECM_TF_SWIFT ECM Users 15 192.0.2.1:7780
```


B.2.6 Increase Consumers

The script is used to increase the number of running Consumers for a given Consumer Group Name, Topic, and Application Node. It applies only till the current application instance is running.

To increase the number of running Consumers, perform the following:

- Configure the arguments and run the following script in the Event tables in the AAI Config Schema:

```
./consumerincrease.sh <TOPIC_NAME> <CONSUMERGRPNAME> <INCREASEBY>
<HOSTNAME:PORT>
```

Table B-5 Argument Description Table for Increase Consumers

Argument	Description
TOPIC_NAME	The name of the Topic.
CONSUMERGRPNAME	The name of the Consumer's Group.
INCREASEBY	The number of Consumers that you want to increase.
HOSTNAME:PORT	The details of the server on which Kafka is installed.

For example:

```
./consumerincrease.sh ECM_TF_SWIFT ECM Users 15 192.0.2.1:7780
```

B.2.7 Replay Messages

This script replays messages persisted to the Message Audit Tables to reprocess messages for failure cases or other reasons. The message is picked from the audit tables and processed instead of the Kafka Topic. The Event ID, which is available as a key in the Message Audit Tables, is used to replay the message. Any message that is read from the Kafka Topic is persisted to Message Audit Table with a Unique Event ID and processed later.

To replay messages, perform the following:

- Configure the arguments and run the following script in the Event tables in the AAI Config Schema:

```
./replaymessage.sh <EVENT_ID>
```

Table B-6 Argument Description Table for Replay Messages

Argument	Description
EVENT_ID	The Unique Identifier of the Event.

For example:

```
./replaymessage.sh JOB1234
```

B.2.8 Save Consumer Properties

This script adds or updates Consumer Properties for a given Consumer ID (Consumer Group). The Properties are used to fine-tune the functioning of the Kafka Consumers.

The Shell Script accepts Consumer ID, Kafka Consumer Property Name, Value, and Datatype.

The properties have to be as given by the Kafka Consumer API. See the following link for a list of properties:

<https://docs.confluent.io/platform/current/installation/configuration/consumer-configs.html>

To update Consumer Properties for a given Consumer ID, perform the following:

- Configure the arguments and run the following script in the Event tables in the AAI Config Schema:

```
./saveConsumerProperties.sh <CONSUMER_ID> <CONSUMER_PROP_NAME>
<CONSUMER_PROP_VALUE> <CONSUMER_PROP_DATATYPE>
```

Table B-7 Argument Description Table for Save Consumer Properties

Argument	Description
CONSUMER_ID	The Unique Identifier of the Consumer.
CONSUMER_PROP_NAME	The name of the Consumer Property that is related to the selected Consumer ID.
CONSUMER_PROP_VALUE	The value of the Consumer Property.
CONSUMER_PROP_DATATYPE	The data type of the Consumer Property.

For example:

```
./saveConsumerProperties.sh 1612 ECM_WR WRITE CHAR
```

B.2.9 Save Producer Properties

This script adds or updates Consumer Properties for a given Producer ID. The properties are used to fine-tune the functioning of the Kafka Producer.

The Shell Script accepts Producer_ID, Kafka Producer Property Name, Value, and Datatype.

The properties have to as given by the Kafka Producer API. See the following link for a list of properties:

<https://docs.confluent.io/platform/current/installation/configuration/producer-configs.html>

To update Consumer Properties for a given Producer ID, perform the following:

- Configure the arguments and Run the following script in the Event tables in the AAI Config Schema:

```
./saveProducerProperties.sh <PRODUCER_ID> <PRODUCER_PROP_NAME>
<PRODUCER_PROP_VALUE> <PRODUCER_PROP_DATATYPE>
```

Table B-8 Argument Description Table for Save Producer Properties

Argument	Description
PRODUCER_ID	The Unique Identifier of the Producer.
PRODUCER_PROP_NAME	The name of the Producer Property that is related to the selected Producer ID.
PRODUCER_PROP_VALUE	The value of the Producer Property.
PRODUCER_PROP_DATATYPE	The data type of the Producer Property.

For example:

```
./saveProducerProperties.sh PROD1234 ECM_SWIFT_WR WRITE CHAR
```

B.2.10 Start Consumers

This Shell Script is used to start Consumers that are stopped for a given Application Node, Consumer Group Name, and Topic.

If you pass ALL in place of the consumergroupname argument, it results in the start of all Consumers in the given Application Node and Topic.

If the number of running Consumers is equal to the configured count in the Consumer Config Tables, no action is applied

To start Consumers, perform the following:

- Configure the arguments and Run the following script in the Event tables in the AAI Config Schema:

```
./startconsumers.sh <TOPIC_NAME> <CONSUMERGRPNAME>/<ALL>  
<HOSTNAME:PORT>
```

Table B-9 Argument Description Table for Start Consumers

Argument	Description
TOPIC_NAME	The name of the Topic.
CONSUMERGRPNAME	The name of the Consumer's Group.
ALL	This value is passed to select all Consumers.
HOSTNAME:PORT	The details of the server on which Kafka is installed.

For example:

```
./startconsumers.sh ECM_TF_SWIFT ECM Users/ALL 192.0.2.1:7780
```

B.2.11 Stop Consumers

This Shell Script is used to stop consumers for a given Application Node, Consumer ID (Consumer Group Name), and Topic. After the servers are restarted, the consumers are initialized and instantiated.

If you pass ALL in place of the consumergroupname argument, it results in the stop of all Consumers in the given Application Node and Topic.

To stop Consumers, perform the following:

- Configure the arguments and Run the following script in the Event tables in the AAI Config Schema:

```
./stopconsumers.sh <TOPIC_NAME> <CONSUMERGRPNAME>/<ALL> <HOSTNAME:PORT>
```

Table B-10 Argument Description Table for Stop Consumers

Argument	Description
TOPIC_NAME	The name of the Topic.
CONSUMERGRPNAME	The name of the Consumer Group.
ALL	This value is passed to select all Consumers.
HOSTNAME:PORT	The details of the server on which Kafka is installed.

For example:

```
./stopconsumers.sh ECM_TF_SWIFT ECM Users 15 192.0.2.1:7780
```

Glossary

Index