

Axiata Approval Workflow Deployment Guide – HUB - V2

Products:

wso2am-1.7.0
wso2bps-3.2.0

Step 01:

Set the carbon offset values for each instance as follows;

Filename: <CARBON_HOME>/repository/conf/carbon.xml

Carbon Product	Offset
wso2am-1.6.0	0
wso2bps-3.2.0	2

Step 02:

To enable workflow execution in API Manager, change the configurations as follows;

Log in to APIM carbon console → Main → Resources → Browse → Navigate to “/_system/governance/apimgt/applicationdata” → Click on “workflow-extensions.xml” → Click on “Edit as Text”

- To enable application creation workflow;

Comment the below line;

```
<ApplicationCreation  
executor="org.wso2.carbon.apimgt.impl.workflow.ApplicationCreationSimpleWorkflowExecutor"/>
```

Uncomment the below section;

```
<ApplicationCreation  
  executor="org.wso2.carbon.apimgt.impl.workflow.ApplicationCreationWSWorkflowExecutor">  
  <Property  
name="serviceEndpoint">http://localhost:9765/services/ApplicationApprovalWorkFlowProcess/</Property>  
  <Property name="username">admin</Property>  
  <Property name="password">admin</Property>  
  <Property name="callbackURL">https://localhost:8243/services/WorkflowCallbackService</Property>  
</ApplicationCreation>
```

Entry/ Property Name	Value
executor	To set the application creation workflow executor customized for Axiata, replace executor “org.wso2.carbon.apimgt.impl.workflow.ApplicationCreationWSWorkflowExecutor” with “org.wso2.carbon.am.axiata.workflow.AxiataApplicationCreationWSWorkflowExecutor”.
serviceEndpoint	The endpoint URL of the “ApplicationApprovalWorkFlowProcess” which is deployed on BPS. Change the ip with the ip address of the server BPS instance is running.
username	Username to access the service.
password	Password to access the service.
callbackURL	Callback service URL. Replace the ip with the ip address of the server APIM is running.

- To enable subscription creation workflow;

Comment the below line;

```
<SubscriptionCreation
executor="org.wso2.carbon.apimgt.impl.workflow.SubscriptionCreationSimpleWorkflowExecutor"/>
```

Uncomment the below section;

```
<SubscriptionCreation
executor="org.wso2.carbon.apimgt.impl.workflow.SubscriptionCreationWSWorkflowExecutor">
  <Property
name="serviceEndpoint">http://localhost:9765/services/SubscriptionApprovalWorkFlowProcess/</Property>
  <Property name="username">admin</Property>
  <Property name="password">admin</Property>
  <Property name="callbackURL">https://localhost:8243/services/WorkflowCallbackService</Property>
</SubscriptionCreation>
```

Entry/ Property Name	Value
executor	To set the subscription creation workflow executor customized for Axiata, replace executor “org.wso2.carbon.apimgt.impl.workflow.SubscriptionCreationWSWorkflowExecutor” with “org.wso2.carbon.am.axiata.workflow.AxiataSubscriptionCreationWSWorkflowExecutor”.
serviceEndpoint	The endpoint URL of the “SubscriptionApprovalWorkFlowProcess” which is deployed on BPS. Change the ip with the ip address of the server BPS instance is running.

username	Username to access the service.
password	Password to access the service.
callbackURL	Callback service URL. Replace the ip with the ip address of the server APIM is running.

Step 03:

Configure both APIM and BPS instances to point to the same "WSO2_CARBON_DB" datasource. To setup APIM with MySQL database, follow the below guide;

<https://docs.wso2.com/display/AM170/Setting+up+with+MySQL>

Once the APIM "WSO2_CARBON_DB" is configured to work with MySQL, change the "WSO2_CARBON_DB" configurations of BPS to point to the same database.

Filename: <BPS_HOME>/repository/conf/datasources/master-datasources.xml

Step 04:

Configure APIM instance "WSO2AM_STATS_DB" to point to Axiata stats database (MySQL). This configuration is required to publish audit data related to application and subscription approvals.

Filename: <AM_HOME>/repository/conf/datasources/master-datasources.xml

Make sure "WSO2AM_STATS_DB" datasource is enabled in below configuration file section;

File Location: <AM_HOME>/repository/conf/
Filename: api-manager.xml
Section: APIUsageTracking → DataSourceName

Step 05:

Configure APIM instance "WSO2AM_DB" to point to Axiata am database (MySQL).

Filename: <AM_HOME>/repository/conf/datasources/master-datasources.xml

Step 06:

Copy MYSQL connector into both APIM and BPS to enable successful communication with the MYSQL database.

Filename: *mysql_connector_java_<MYSQL_VERSION>_bin.jar*
E.g.: mysql_connector_java_5.0.8_bin.jar
File Location: *<AM_HOME>/repository/components/lib/*
<BPS_HOME>/repository/components/lib/

Step 07:

Copy relevant endpoint references in to BPS.

The .epr files can be found at the below Git repository location;
<GIT_REPO>/<MIFEHUB_BRANCH>/mife/workflow/endpoints/

The files need to be copied to the following location;
<BPS_HOME>/repository/conf/epr/

If the “epr” directory is not available in the above location, manually create it before copying the files.

The list of endpoint reference files related to each workflow process is as follows;

Workflow	Endpoint Reference
Application Approval	ApplicationService.epr
	ApplicationCallbackService.epr
	AxiataApprovalHandlerService.epr
Subscription Approval	SubscriptionService.epr
	SubscriptionCallbackService.epr
	AxiataSubscriptionApprovalService.epr
<Common>	AxiataWorkflowApprovalAuditService.epr

Each .epr file needs to be configured to include relevant ip addresses of each service endpoint. Default endpoints are configured to “localhost”.

Step 08:

Deploy “AxiataWorkflowHandler” module.

“AxiataWorkflowHandler” is an OSGi bundle (E.g.: AxiataWorkflowHandler-1.0.0.jar) which is responsible for handling Axiata specific database invocations.

The module implementation can be found at the below Git repository location;

`<GIT_REPO>/<MIFE_BRANCH>/mife/workflow/AxiataWorkflowHandler/`

Checkout and build the code. Once the build is successful, the deployable artifact (E.g.: AxiataWorkflowHandler-1.0.0.jar) can be found at;

`/AxiataWorkflowHandler/target/`

Copy the above artifact in to below AM location;

`<AM_HOME>/repository/components/dropins/`

Step 09:

“AxiataWorkflowHandler” module is dependent on “Dbutils” module. Hence that module is also required to be available to facilitate the functionality.

Filename: `Dbutils-<VERSION>-SNAPSHOT.jar`

E.g.: Dbutils-1.0-SNAPSHOT.jar

File Location: `<AM_HOME>/repository/components/lib/`

It is required to modify the database configurations (Database name, username, password) bundled with the jar file.

Filename: `dbconfig.properties`

File Location: `Dbutils-<VERSION>-SNAPSHOT.jar/com/axiata/dialog/dbutils/`

Step 10:

Configure workflow host object.

Filename: `module.xml`

File Location: `<AM_HOME>/modules/ dialogbilling/`

Create the above folder and file if does not exist.

Enter below host object configuration entry;

```
<hostObject>
  <className>org.dialog.custom.hostobjects.WorkflowHostObject</className>
  <name>AxiataWorkflow</name>
</hostObject>
```

The host object implementation can be found at below Git repository location;
<GIT_REPO>/<MIFEHUB_BRANCH>/org.dialog.custom.hostobject/1.0.0/

Build the above source code and copy the resulting Osgi bundle to the below APIM location;
<AM_HOME>/repository/ components/ dropins/

Step 11:

Configure the proxy services used to perform Axiata specific database invocations. These proxy services expose the service implementations available in “AxiataWorkflowHandler” module.

The proxy configurations can be found at the below Git repository location;
<GIT_REPO>/<MIFEHUB_BRANCH>/mife/workflow/proxy-services/

The files need to be copied to the following location;
<AM_HOME>/repository/deployment/server/synapse-configs/default/proxy-services/

The proxy service files related to each workflow process is as follows;

Workflow	Proxy Service
Application Approval	AxiataApprovalHandlerProxy.xml
Subscription Approval	AxiataSubscriptionApprovalService.xml
<Common>	AxiataWorkflowApprovalAuditService.xml
	WorkflowCallbackService.xml

Step 12:

Copy the latest jaggery applications (publisher, store, manage) in to APIM.

Jaggery apps can be found at the following Git repository location;
<GIT_REPO>/<MIFEHUB_BRANCH>/mife/jaggery-apps/

Copy the jaggery apps in to below APIM location;
<AM_HOME>/repository/deployment/server/jaggeryapps/

Step 13:

Start the APIM and BPS servers.

Step 14:

Update APIM tiers.xml file with the latest content.

Log in to APIM carbon console → Main → Resources → Browse → Navigate to “/_system/governance/apimgt/applicationdata” → Click on “tiers.xml” → Click on “Edit as Text”

Step 15:

Deploy the approval workflow business processes in BPS.

The workflow artifacts can be found at below Git repository locations;

Application Approval Workflow

`<GIT_REPO>/<MIFEHUB_BRANCH>/mife/workflow/bpel/ApplicationApprovalWorkFlowProcess/`

Subscription Approval Workflow

`<GIT_REPO>/<MIFEHUB_BRANCH>/mife/workflow/bpel/SubscriptionApprovalWorkFlowProcess/`

In order to create the deployable process archive using above workflow artifacts, compress the above source folders in to .zip archives.

E.g.: ApplicationApprovalWorkFlowProcess-1.0.0.zip
SubscriptionApprovalWorkFlowProcess-1.0.0.zip

Log in to BPS admin console and upload the processes.

Main → Manage → Processes → Add

Step 16:

Deploy the humantasks in BPS.

The humantask artifacts can be found at below Git repository locations;

Application Approval Humantask

`<GIT_REPO>/<MIFEHUB_BRANCH>/mife/workflow/humantasks/ApplicationsApprovalTask/`

Subscription Approval Humantask

`<GIT_REPO>/<MIFEHUB_BRANCH>/mife/workflow/humantasks/SubscriptionsApprovalTask/`

In order to be able to deploy the humantask artifacts, compress the above projects in to .zip archives.

E.g.: ApplicationsApprovalTask-1.0.0.zip
SubscriptionsApprovalTask-1.0.0.zip

Log in to BPS admin console and upload the humantasks.

Main → Manage → Human Tasks → Add

Step 17:

To enable opCo level approvals, for each opCo a separate role needs to be created in the below syntax with specified permissions;

Login to APIM management console → Configure → Users and Roles → Roles → Add New Role

Role Syntax: <opCo_name>-admin-role
 E.g.: dialog-admin-role

Permissions: Login
 View Task List

The create one or more users with the above role. These users are able to perform opCo level task approvals on behalf of the opCo specified in the role name.

Login to APIM management console → Configure → Users and Roles → Users → Add New User

Step 18:

Configure “manage” jaggery app to enable newly created opCo roles.

Filename: site.json
File Location: <AM_HOME>/repository/deployment/server/jaggeryapps/manage/site/conf/
Section: “allowedRoles”

Add the new opCo role in to comma seperated “allowedRoles” list. This enables the role to login to “manage” module and perform tasks authorized for OpCo users.