Embedded Cloud Workflow Engine

Embedded Cloud Workflow Engine (ECW) is a self contained J2EE complaint application. You can download and install on any J2EE compliant container and get started in no time.

Steps to Get Started with ECW

- Download and configure the Web Application on your J2EE compliant container. The application comes with a pre-configured Tomcat instance.
- Create a database instance which will store user configuration data
- Create a database which will store all the data flow logs. It can be the same as the configuration database or separate. Since we expect the log database to grow rapidly we recommend that the log database be separate to that of configuration database.
- Setup Web application config files to the configuration database and log database
- Setup Web application with the default Processes to run when an application is invoked
- Setup Web application with the default configuration template to use.

Download & Install

The following distributions are available for use:

Zip	TODO
Docker	`docker pull dbsync/dbsync-ecw`
Amazon AWS	Linux: Windows:

Database Setup

ECW comes with a default setup where the Partner app needs to provide a storage for the User Configurations. These configurations are "connections" that end users will setup to connect to their end application. For example, if the Partner App relies on a database to run its platform, then the one end is likely to be static, while the connecting application for example an Accounting application like QuickBooks, will vary from customer to customer. 'sys_partner_app_config' is the table where the engine will store and fetch these connector access / login details to perform the integration. Note that DBSync Cloud Platform will not ever store the customer specific connection information.

- **User Config Table:** This table stores user configuration
- **User Runtime Table**: This table stores runtime and session variables like last successful run etc.

User Config Table Schema

```
CREATE TABLE `sys_partner_app_config` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `profile` longtext,
  `created_date` timestamp NULL DEFAULT NULL,
  `modified_date` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id` (`id`),
  KEY `username` (`username`)
);

User Runtime Table Schema
CREATE TABLE `sys_partner_app_runtime` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `process` varchar(50) NOT NULL,
  `session_properties` text,
  `created_date` timestamp NULL DEFAULT NULL,
  `modified_date` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `username` (`username`),
  KEY `process` (`process`)
);
```

When integrations run, it will produce data flow logs. Each log will be saved in 'sys_dataflow_log' table. Partner app developer should design and administer these tables and manage storage. ECW will "NOT" be truncating or managing this table. It is recommended to add a cron script to delete logs every so many days.

Log Table Schema

```
CREATE TABLE `sys_dataflow_log` (
  `id` bigint(20) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `name` varchar(50) DEFAULT NULL,
  `records_processed` varchar(20) DEFAULT NULL,
  `message` text,
  `log_messages` longtext COMMENT 'Log messages for processed data for
c2c',
  `created_time` datetime DEFAULT NULL,
  PRIMARY KEY (`id`,`username`),
  UNIQUE KEY `id` (`id`)
)
```

Setting up Configuration Files

There are 2 sets of files that will need to be configured

- Configuration files to let engine know how to interact with user and log storage
- The "Process" files that needs to run for a given application.

-

partner.properties

Location: (inside “www/WEB-INF/conf/” folder)

Property		Description
partnerAppKey	Required	Will be provided by DBSync
partnerAppToken	Required	Will be provided by DBSync
partner.database.driverClassName	Required	Profile Details and Logs are stored in Database, for this property, provide the Database driver class name. For example : for mysql database, this property will be com.mysql.jdbc.Driver
partner.database.url	Required	Profile Details and Logs are stored in Database, for this property, provide the Database Url along with Database name. For example : for mysql database, this property will look like jdbc:mysql://<ServerName>:<Port>/<Database Name>.
partner.database.username	Required	Profile Details and Logs are stored in Database, for this property, provide the Database Username.
partner.database.password	Required	Profile Details and Logs are stored in Database, for this property, provide the Database Password.
dbsync.logger	Optional	If you don’t want to store the logs in the default database and want to store the logs in other database, then enable this property. Property value will be constant : com.mydbsync.cw.embedded.EmbeddedLogger If you want to store the logs in the default database itself, then remove or comment this property. If you enable dbsync.logger, below 4 properties will be used for storing the logs in the different database.
log.database.driverClassName	Optional	Provide the Driver class name. Ex: If mysql, then com.mysql.jdbc.Driver
log.database.url	Optional	Provide the Database Url along with Database Name. Ex: If my sql, then jdbc:mysql://<ServerName>:<Port>/<Database Name>.
log.database.username	Optional	Provide the Logging database Username.
log.database.password	Optional	Provide the database Password.
partner.profile.template	Optional	profile.xml is the file, where all connector properties are stored. If you have profile.xml and want to use it as template, then provide the absolute path of profile.xml Ex: <Your path>/profile.xml
partner.app.<process>.pdl	Required	<process> is the built process accessible to this runtime engine. For example: partner.app.YourApp-Quickbooks.pdl will point the the project / process that is developed from Cloud Workflow - Development Studio. You can also Download the project (as Zip) and extract in a folder of your choice. Ex : If your pdl name is QuickBooksPdl, and process xml name is processdefinition_CustomerToAccount.xml, then the path will be partner.app.YourApp-Quickbooks.pdl=<Path to the Directory>/processdefinition_CustomerToAccount.xml You will need one entry for each process available to your application. As an example you will have one for QuickBooks, another for Microsoft Dynamics GP and so forth.

partner_profile.xml

Location: (inside “www/WEB-INF/conf/” folder). Also see "partner.profile.template" property above in case you want to store it outside of the default installation (Recommended)

This file contains all connector properties. Connector properties are the application properties, from where and to where, you sync the data. All connector properties are closed within <adapters> root tag. Each specific connector will be enclosed within <adapter> tag.

There are 2 ways to generate this file

1. User CloudWorkflow to define a process that you would like to deploy on ECW. Download the Zip file, the default "profile.xml" will be your starting point.
2. Use notepad to create your default setup (Advanced Users)

Sample Profile

```
<?xml version="1.0" encoding="UTF-8"?>
<profile id="localhost@avankia.com" name="Salesforce_to_Database"
loglevel="ALL" email="" guid="325735D5">

  <adapter name="mysql"
type="com.avankia.appmashups.engine.conversion.adapters.DatabaseAdapt
er">
    <property name="password" encrypted="false" >XXXX</property>
    <property name="url">jdbc:mysql://localhost:3306/dbsync</property>
    <property name="autocommit">true</property>
    <property name="adp_name">mysql</property>
    <property name="driver">mysql</property>
    <property name="dbname">dbsync</property>
    <property name="username">root</property>
  </adapter>

  <adapter name="Salesforce"
type="com.avankia.appmashups.engine.conversion.adapters.SalesforceAda
pter">
    <property name="securityToken" />
    <property name="sid">XXXX</property>
    <property name="transport.compression" />
    <property name="username">abc@def.com</property>
    <property name="refreshToken">AAA</property>
    <property name="password" encrypted="false">YYYY</property>
    <property
name="endpoint">https://login.salesforce.com/services/Soap/u/32.0</pr
operty>
    <property name="adp_name">Salesforce</property>
  </adapter>
</profile>
```