AquaLogic BPM 6.0 Upgrade Guide

Version: 6.0

Contents

Upgrade Overview

This document provides the recommended guidelines for upgrading to the latest version of ALBPM and migrating ALBPM projects from previous versions of the product.

Here's an overview of the main steps needed to upgrade ALBPM projects from 5.x to 6.0:

1. Understand the requirements for upgrading, as described in *Upgrade Requirements and limitations* on page 4.
2. Migrate your 5.x projects' code to 6.0. Follow the sections described in *Migrating Project Code* on page 6 in the order presented.
3. Upgrade your 5.x Enterprise environment to 6.0. Follow the sections described in *Migrating Enterprise Environment* on page 10 in the order presented. In short:
   a. Install ALBPM Enterprise 6.0.
   b. If upgrading from version 5.5, run the provided SQL scripts to alter your databases structures (not needed if upgrading from 5.7).
   c. Configure ALBPM 6.0 to point to your existing 5.x Directory database.
   d. Make the needed changes to make your environment compatible with 6.0.
   e. Re-deploy your ALBPM projects on your newly upgraded environment.
   f. Start your Process Execution Engines.

The first time you start ALBPM 6.0 Process Administrator during the upgrade, it will automatically alter the schema structure of the Directory database. Likewise, when you first start each Engine, it will alter the schema structure of the Engine database. This means that pre-existing process instances will continue their execution on the new 6.0 environment.

Upgrade Requirements and limitations

You must satisfy a list of requirements and assumptions before attempting to upgrade an existing ALBPM installation and migrating existing projects to this new version of ALBPM.

Assumptions

This document assumes you are familiar with your ALBPM 5.x installation and understand all the features used by the projects being migrated.

Supported Versions

The guidelines in this document assume you are upgrading from ALBPM version 5.5.11+ (with latest patch) or 5.7.2+ (with latest patch) to version 6.0.3. If your projects are currently developed or deployed on earlier versions of ALBPM, you must first upgrade to the latest 5.5 or 5.7 version before moving to 6.0.3.

You must upgrade to version 6.0.3 or newer. Upgrading to earlier 6.0.x versions is not supported.

This document also includes special steps for upgrading from ALBPM WE 5.7 (WorkSpace Extensions for ALI) environments.

Supported Database Servers

Version 6.0 of ALBPM drops support for Microsoft SQL Server 2000 and Oracle 8 as the back-end store for ALBPM Directory and Engine databases. If your current ALBPM 5.x environment is deployed any of these database server versions, you must upgrade your database server before upgrading ALBPM.

Database privileges

The database (JDBC) users configured for the Directory and Engine Databases must have enough privileges to perform SQL DDL (Data Definition Language) operations (like ALTER, CREATE and DROP statements). Otherwise, the Process Administrator application and the Engine will fail to migrate the structures of their SQL tables and will not to start. This is only needed for the first time you start Process Administrator and each Engine.

ALBPM for WebLogic

Version 6.0 of ALBPM only supports versions 9.2 and 10 of WebLogic. If your current ALBPM 5.x environment is deployed on older versions of WebLogic (8.x), you must upgrade WebLogic before upgrading ALBPM.

Refer to the *ALBPM Installation Guide* for more details about which versions of WebLogic are supported.

For upgrading WebLogic Server, refer to *WebLogic 9.2 Upgrade procedures* and *WebLogic 10 Upgrade procedures*.

ALBPM for WebSphere

Version 6.0 of ALBPM only supports version 6.1.x of WebSphere. If your current ALBPM 5.x environment is deployed on older versions of WebSphere (5.x or 6.0.x), you must upgrade WebSphere before upgrading ALBPM.

Refer to the *ALBPM Installation Guide* for more details about which versions of WebSphere are supported.

Migrating Project Code

These sections provide specific procedures for migrating existing code based on ALBPM 5.7.2 into ALBPM 6.0.x. You must migrate your project code with ALBPM Studio before upgrading an existing ALBPM Enterprise environment.

Import 5.7 project into Studio 6.0

Follow this procedure to import an ALBPM project developed with version 5.7 into ALBPM Studio 6.0.

1. Open your project with ALBPM Studio 5.7.
2. Make sure the project is free from any errors. Select **Run ➤ Check All** on the menu to run the project consistency check.

   You must fix all errors using Studio 5.7 before moving to 6.0.
3. Export your project from ALBPM 5.7.

   From the menu, select **File ➤ Export Project...**. Select the **Include all libraries** option and follow the next screens to generate an `.exp` file (the exported ALBPM project).
4. Start ALBPM Studio 6.0 and import the project you just exported.

   Note: Opening an expanded 5.7 project from Studio 6.0 is not supported. You must first export the project to an `.exp` file with Studio 5.7 (see previous step).

   a) Select **File ➤ Import...** on the menu.
   b) Select **BPM ➤ Exported ALBPM Project into Workspace**. Push **Next** and follow the Wizard to select the exported project file and provide a name for the project.

   As a result, your project is now included into your ALBPM Studio Workspace.

   If there are any compilation errors, they will appear in the **Problems** view.

Fix External Resources

Due to improvements to the way ALBPM handles integration with other external systems, you might need to do some minor changes to the External Resources of your projects.

1. Open your project with ALBPM Studio 6.0.
2. Migrate to the new JDBC drivers provided by BEA.

   ALBPM now supports several SQL databases out of the box (you don't need to provide an external JDBC driver anymore). If your project integrates with any of these databases, it is recommended that you migrate to the new drivers.

   a) Remove references to JDBC Java Libraries.

      On the **Project Navigator** expand **External Resources** and remove those Java Libraries resources holding JDBC drivers (refer to the "External Resources" section of Studio's online help for details).

   b) Modify any existing SQL External Resources to use BEA's JDBC drivers

      On the **Project Navigator** expand **External Resources** and double-click on a SQL database resource. In the **Edit External Resource** dialog change the value of the **Supported Type** drop-down to the corresponding BEA driver for your database.

For advanced JDBC URL configurations refer to the custom properties of the JDBC driver:

- For Oracle: *http://edocs.bea.com/wls/docs100/jdbc_drivers/oracle.html#*
- For Microsoft SQL Server: *http://edocs.bea.com/wls/docs100/jdbc_drivers/mssqlserver.html*
- For DB2: *http://edocs.bea.com/wls/docs100/jdbc_drivers/db2.html*
- For Informix: *http://edocs.bea.com/wls/docs100/jdbc_drivers/informix.html*
- For Sybase: *http://edocs.bea.com/wls/docs100/jdbc_drivers/sybase.html*

3. Re-catalog all components in the Catalog of your project.

   This step is needed to re-generate the metadata associated with each component. You must re-catalog all types of components, including Java classes, SQL tables, Web Services, .Net, JNDI, CORBA, COM and SAP components.

   Note: On version 5.x, Java type `java.util.Date` (when used in method arguments or as return type) was cataloged as `Fuego.Lang.Time`. On version 6 onwards, the type `java.util.Date` is maintained as such when cataloging. This doesn't cause incompatibilities.

   Important: Cataloged Java classes do not include inherited members anymore (see next step).

4. Catalog additional Java classes as needed.

   On version 5.7, a cataloged Java class included all members inherited from parent classes. Starting with version 6.0, cataloged Java classes only include methods and attributes explicitly defined by the class, excluding the ones inherited by its parent class. If there's code in your project using inherited members of a class, you must also catalog the parent Java classes that includes such members.

   For example, if your project has cataloged EJBs (Enterprise JavaBeans), you must also to include the standard `javax.ejb.*` classes and interfaces into your catalog. Otherwise, standard methods like `getEJBHome()` and `getHandle()` do not show as members of your Bean.

Changes in Standard Components

Some standard components of the catalog are no longer available on this version, and others are now deprecated. This version also provides new components, which can cause conflicts when migrating old code.

Removed and Deprecated components

You have to adapt your code if your project depends on components which have been removed on this version of ALBPM. Refer to *Removed and Deprecated Components* on page 32 to see which components have been removed or deprecated on each version of ALBPM.

Note about New Components

Standard components which are new to this version can cause naming conflicts with existing custom components in your project.

For example, a new component named `MailSender` has been added to the `Fuego.Net` module. If your project defines a BPM Object type also named `MailSender`, existing code referring to `MailSender` now becomes ambiguous.

To resolve this conflict, you should use the fully qualified names of those components:

```
sender = MailSender()            // ambiguous
sender = YourModule.MailSender() // explicitly refer to custom BPM Object
sender = Fuego.Net.MailSender()  // explicitly refer to standard component
```

Important: This version of ALBPM supports Java 1.5. Therefore, modules under `Java.*` also include several new classes which may collide with your existing BPM Object names. Use fully qualified names in case of ambiguity.

Changes in Dashboards

The Dashboard components have been improved on ALBPM 6.0. They use a new graphics layer and provide new functionality. As a result, you might need to modify some of your existing 5.7 Dashboards to make them fully compatible with this new version.

Feature changes

Waterfall charts and Multiple-Pie charts are not available on version 6.0. When a project is imported from 5.7 they will be replaced by a bar chart.

Some properties of the Dashboard widgets are not available anymore.

- No canvas plot outline and plot background in pies.
- Horizontal layout is not available for area charts, stacked area charts and line charts.
- On Ctrl-Click. For all chart widget, this event cannot be captured anymore.

Dashboard Display Properties (on Global Activities) are only taken into account when displaying the Dashboards through the classic 5.7 WorkSpace.

Code changes

Existing 5.7 Dashboards featuring "drill down" from a chart to a list of process instances use code that needs to be changed.

The out-of-the-box Workload dashboards provide this feature. To make them compatible with 6.0 you need to change the auto-generated methods `drillDownTo*()` (generated by the dashboard editor) and `drillDownToInstancesView()` (generated with the out-of-the-box dashboards), replacing the following code:

```
...
result = getQueryStringFor(
                ClientBusinessProcess.processService,
                viewId : "viewId")

//Just to open the url
openURL this
using url = result,
      newWindow = false,
      standardBrowser = false
....
```

with:

```
broadcastViewChangeEvent this
    using viewId = viewId
```

This new code is simpler and allows for displaying the dashboard and the list of instances view at the same time in the same page (in different *portlets*).

Changes in Implicit Java Classpath

Implicit directories added into Studio's Java "classpath" have changed in version 6.0.

Version 5.x of ALBPM Studio loaded Java classes and resources from the `studio/ext/` directory. This allowed your project code to load any resources (like property or configuration files) placed in this directory.

On ALBPM Studio version 6.0, directory `studio/ext/` was removed. You should place your resource files on your project's `lib/` directory instead. This is a cleaner solution, since each project has its own `lib/` directory and you don't need to modify your Studio installation directories anymore.

On ALBPM Enterprise, the same resource files should be placed in the `<engine_home_dir>/lib` directory, where <engine_home_dir> is the ALBPM Engines' Home directory, configured via Process Administrator.

On ALBPM Enterprise for J2EE, the resource files should be placed in a directory loaded by your application server's class loaders. For the case WebLogic, refer to *http://edocs.bea.com/wls/docs92/programming/classloading.html#wp1096756*.

Important: When exporting a project from ALBPM Studio for deployment on ALBPM Enterprise, you should select option **Include Versionable Libraries Only**. This prevents non-versionable libraries and resource files from being included in the exported project file. If you don't select this option, resource files included in the project may be loaded before the files placed in `<engine_home_dir>/lib`.

Changes in PBL

Changes in Process Business Language

Comparison Operator

The comparison operator (==) is no longer allowed outside of boolean expressions.

For example, the following code was allowed on version 5.x, but will generate a **Statement is expected** error:

```
Boolean var;
var == false; // error reported on version 6
```

These kind of expressions do not product any effects. If your existing 5.x code fails on version 6 because of this error, you must revisit the code and understand what was the original intention behind it. Most commonly, it was a mistake of using the comparison operator (==) instead of the assignment operator (=).

Changes in Transformations

The deprecated feature of Transformations have been removed on ALBPM version 6.0.

Transformation methods existing on ALBPM 5.x projects are automatically converted to regular PBL code when the project is imported in ALBPM Studio 6.0.3+. Backwards compatibility is preserved, so no change is needed on your existing code base.

Each Transformation method is converted into three new methods:

- `<transformation>Method()` - Main method replacing transformation
- `<transformation>MethodBegin()` - helper method
- `<transformation>MethodEnd()` - helper method

For example, a Transformation method named `toMyObject()` on your 5.x project is converted into these three methods:

- `toMyObject()` - Main method replacing transformation
- `toMyObjectBegin()` - helper method invoked from toMyObject()
- `toMyObjectEnd()` - helper method invoked from toMyObject()

Migrating Enterprise Environment

These sections provide specific procedures for upgrading an existing environment running on ALBPM Enterprise 5.7.2 to ALBPM Enterprise 6.0.x.

Follow the procedures in these section to upgrade to ALBPM 6.0, re-using the existing ALBPM 5.x Directory and Engine databases.

The first time you start ALBPM 6.0 Process Administrator during the upgrade, it will automatically alter the schema structure of the Directory database. Likewise, when you first start each Engine, it will alter the schema structure of the Engine database. This means that pre-existing process instances will continue their execution on the new 6.0 environment.

Important: You must perform a full back-up of your 5.x installation, including all databases involved. Your 5.x installation will become unusable after the upgrade. During the the upgrade procedures, the database structures are converted to the new version and will no longer be compatible with version 5.x.

Important: The database (JDBC) users configured for the Directory and Engine Databases must have enough privileges to perform SQL DDL (Data Definition Language) operations (like ALTER, CREATE and DROP statements). Otherwise, the Process Administrator application and the Engine will fail to migrate the structures of their SQL tables and will not to start. This is only needed for the first time you start Process Administrator and each Engine.

Follow all these sections in the order presented. Some sections only apply to specific environments and may be safely skipped otherwise.

Installing ALBPM 6.0 Enterprise

The first step of the upgrade process is to download and install ALBPM 6.0.

Follow this procedure to install ALBPM Enterprise 6.0. Refer to the *ALBPM Installation Guide* for more details about the installation process and requirements.

1. Obtain the installation file from the *BEA AquaLogic BPM downloads* page, and download your desired edition of AquaLogic BPM Enterprise 6.0 for your platform. If you are not registered, you will need to register with BEA in order to proceed with the download.
2. Execute the installation file.

   Note: To use an already installed Java VM, specify the full path to the `java` command passing the `LAX_VM` parameter to the installer. You must always specify the location to an existing Java installation if a JVM is not included with the installer (refer to *Required Java Virtual Machines32-bit Windows and Linux versions of all ALBPM components include the JVM. In other versions, however, you need to make sure a compatible JVM is installed.* for details). Example:

   ```
   ALBPMEnterpriseSA603_sol.bin LAX_VM /usr/jdk1.5/bin/java
   ```

   The installation program will prepare to proceed with the installation, and display the **Introduction** page of the installation utility.
3. At the **Introduction** page, click **Next.**
   The **Choose Install Folder** will appear.
4. In the **Choose Install Folder**, enter the path or select a folder where you will install Enterprise. If you've entered a path but wish to use the installer's default, click on **Restore Default Folder** to obtain it. Click **Next** to proceed.

   We recommend using the default folder path whenever possible, as this will help simplify support.

In Windows or 32-bit Linux, the **Pre-Installation Summary** page appears, and you should go to step 7. Otherwise, the **Choose J2SE 1.5 SDK** page appears.

5. In the **Choose J2SE 1.5 SDK** page, select the Java 1.5 SDK installed in your system that you want Enterprise to use. The installer will list those SDKs that it has detected. If you wish to specify a JDK that is not on the list, you can click on **Search for Others** for an automatic search, or on **Choose Another...** to look for the SDK manually. When you have selected the Java 1.5 JDK that you want, click **Next**.
   The **Pre-Installation Summary** page appears.
6. Verify the settings shown in the summary. If you need to make any changes, click **Previous** to go to the page where you need to make the change. Once you are ready to proceed, click **Install.**
   The installation is performed. Once it has finished, the **Install Complete** page appears.
7. In the Install Complete page, you can choose to exit the installer, to launch the Admin Center program, or to launch the Configuration Wizard. If you exit, you can run the Admin Center at a later time. You can also execute the Configuration Wizard from Admin Center.
   In all cases, the installation utility will exit.

Upgrade steps for ALBPM WE Environments

Follow these steps only if you are upgrading from an ALBPM WE (WorkSpace Extensions) 5.7 environment.

If you are upgrading an ALBPM environment which does not include ALBPM WE 5.7, skip these sections.

Starting with version 6.0, all editions of ALBPM Enterprise can integrate with AquaLogic User Interaction (ALUI) out of the box. There is no longer the need to install a separate package to use WorkSpace Extensions features for ALUI.

Configuring the AquaLogic Identity Service

This procedure shows you how to configure the AquaLogic Identity Service for integrating ALBPM with ALUI.

Important: If you have a older version of ALI Identity Service installed on your system, you should un-install it first to avoid conflicts.

To install the ALI Identity Service:

1. Install the ALI Identity Service located in `BEA_HOME/albpm6.0/enterprise/ptids/1.2/bin` `BEA_HOME/albpm6.0/j2eewl/ptids/1.2/bin` `BEA_HOME/albpm6.0/j2eews/ptids/1.2/bin` using the following command:

   On Windows: `.\service.bat install`
2. Start the ALI Identity Service.

   On Windows you may start the **BEA ALI Identity Service** from the standard Windows Services panel.

   On Unix environments, you start the service with the following command:

   `./service.sh start`

   `./service.sh console` (to run on the foreground)
3. Verify that the ALI Identity Service started without errors.

   You may check the log file located at: `BEA_HOME/albpm6.0/enterprise/ptids/1.2/logs/service.log` `BEA_HOME/albpm6.0/j2eewl/ptids/1.2/logs/service.log` `BEA_HOME/albpm6.0/j2eews/ptids/1.2/logs/service.log`.

   If the service started successfully, there should be no errors and the final line of the log should state `***Initial Sync Completed***`.

Enabling ALBPM 6.0 for ALUI

You configure ALBPM 6.0 for ALUI integration using the WorkSpace Extension Configuration Wizard tool. Follow the procedure in this section only if upgrading an ALBPM 5.7 WE (WorkSpace Extensions) environment.

If you are upgrading an ALBPM environment which does not include ALBPM WE 5.7, skip this section.

The WorkSpace Extension Configuration Wizard is located at: `ALBPM_HOME\bin\weconfigwizard.exe` (example: `c:\albpm6.0\enterprise\bin\weconfigwizard.exe`).

1. Run the WE Configuration Wizard as outlined in the following table:

Configuration Wizard Page	User Input
ALUI Database Type	Select the database type used by AquaLogic User Interaction (ALUI).
ALUI Database Connection Information	Provide the connectivity information for the ALUI database.
ALI Identity Service Database Type and Service Ports	Select the database type and service ports for the AquaLogic Interaction Identity Service (Hydrogen). You must enter the database type of your existing Identity Service database used by ALBPM WE 5.7.
ALI Identity Service Database Connection Information	Provide the connectivity information for the AquaLogic Interaction Identity Service (Hydrogen) database. You must enter the information to connect to the existing database used by ALBPM WE 5.7.
Show SQL Script	You do not need to run this script, since you are upgrading an existing database. Just continue to the next page.
Enter Image Server URL	Provide the connectivity information to the ALI Image Server. Leave the **Image Server URL** field blank if you are not using ALI Image Server for ALBPM WorkSpace images. After this step, a file named `imageServerResources.zip` will be generated under the `ALBPM_HOME/serverpackages/` directory.
Configuration Finished	Click **Finish** to close the WorkSpace Extensions configuration wizard.

Un-Deploying 5.7 J2EE Applications

On ALBPM 5.7 environments running on WebLogic or WebSphere you must first stop and un-install all ALBPM-related .EAR and .WAR applications.

This section only applies for the J2EE edition of ALBPM. If you are upgrading an ALBPM Standalone Edition environment then you should skip this section.

1. Remove file `fuegoj2ee-lib-all.jar` from your J2EE environment.

   ALBPM 5.7 required this JAR file to be included in your WebLogic and WebSphere environments. You must now remove all references to it. Starting with ALBPM 6.0, this library file is no longer needed: all libraries are now included within each application archive.

2. Login to your application server console, stop and un-install *all* ALBPM-related applications.

   Make sure you stop and un-install all applications, including:

   - Fuego J2EE Deployer.
   - Engine (You may have one or more Engine applications).
   - WorkSpace.
   - WorkSpace Administrator.
   - Archive Viewer.
   - ALBPM projects (you will have one enterprise application for each ALBPM project deployed).

3. Stop your application server nodes.

Make sure all your WebLogic or WebSphere servers where ALBPM 5.7 was running are completely stopped. This includes the administration server and all additional servers running any ALBPM component.

Upgrade ALBPM 5.5 DB structures

If upgrading from version 5.5.x, you must first run a set of SQL scripts on your existing ALBPM databases. You can safely skip this section if upgrading from version 5.7.x.

Follow these steps only if you are upgrading from ALBPM version 5.5.x (and not 5.7.x).

ALBPM 6.0.3 includes SQL scripts for upgrading the structure of all ALBPM 5.5 databases and make them compatible with version 6.0.3.

The provided SQL scripts cover the following ALBPM databases:

- Directory database
- Engine database
- Archiving database
- BAM (Business Activity Monitoring) database
- DataMart database

Upgrade ALBPM 5.5 SQL Server DBs

This section explain how to upgrade the structure of ALBPM 5.5 databases on Microsoft SQL Server.

Follow these steps if you are upgrading from ALBPM version 5.5.x (and not 5.7.x) using Microsoft SQL Server as the database server.

You need a database user with enough privileges to perform SQL DDL (Data Definition Language) operations (like ALTER, CREATE and DROP statements) on the ALBPM databases.

1. Locate the migration SQL scripts for your database.

   ALBPM Enterprise 6.0.3+ (on the first Hotfix) includes SQL scripts for upgrading the structure of ALBPM 5.5 databases on Microsoft SQL Server. These scripts are located under the following directory: `<ALBPM_DIR>/conf/migration/ms-sql/`. Refer to *Upgrade Scripts for 5.5 on SQL Server* on page 33 for a detailed list of files included.

   You have one SQL script for each ALBPM database to upgrade:

   - `migration_directory.sql` (for ALBPM Directory database)
   - `migration_engine.sql` (for ALBPM Engine database)
   - `migration_archiving.sql` (for ALBPM Archiving database)
   - `migration_bam.sql` (for ALBPM Business Activity Monitoring database)
   - `migration_dataMart.sql` (for ALBPM DataMart database -formerly "DataStore" for historical reporting-)

2. Launch Microsoft SQL 2005 Management Studio
3. For each ALBPM 5.5 database, follow the steps below to run the SQL migration script.
   a) Open the SQL script file.

      From the menu select: **File ➤ Open ➤ File...** and select the .sql file.

   b) Enter the connection information to the database
   c) Modify the SQL script to match your database name and user ID.

      1. Modify the `USE fuego_55_xxxxx` line of the script to use the name of your database. Example: If the name of your directory database is `my_fuego_dir`, the line should read:

         ```
         USE my_fuego_dir
         ```

2. Modify the `SET @sql_user = 'fuego_55_xxxxx'` line of the script to use your database user ID. Example: If the ID of your database use is `my_fuego_user`, the line should read:

   ```
   SET @sql_user = 'my_fuego_user'
   ```

d) Review the script and ensure it fits your environment.

Each `.sql` script may include specific instructions on how it works and how to use it. Make sure you read them before executing any SQL statements.

e) Execute the SQL script to upgrade the database structure.

From the menu, select: **Query ➤ Execute**.

After the execution of each script, you should see a **Query Executed Successfully** message in the status bar at the bottom of the screen.

Upgrade ALBPM 5.5 Oracle DBs

This section explain how to upgrade the structure of ALBPM 5.5 databases on Oracle.

Follow these steps if you are upgrading from ALBPM version 5.5.x (and not 5.7.x) using Oracle as the database server.

You need a database user with enough privileges to perform SQL DDL (Data Definition Language) operations (like ALTER, CREATE and DROP statements) on the ALBPM databases.

1. Locate the 5.5 to 6.0 migration SQL scripts for Oracle.

   ALBPM Enterprise 6.0.3+ (on the first Hotfix) includes SQL scripts for upgrading the structure of ALBPM 5.5 databases on Oracle. These scripts are located under the following directory: `<ALBPM_DIR>/conf/migration/oracle/`. Refer to *Upgrade Scripts for 5.5 on Oracle* on page 33 for a detailed list of files included.

   You have one directory for each ALBPM database to upgrade:

   - `directory/` (for ALBPM Directory database)
   - `engine/` (for ALBPM Engine database)
   - `archiving/` (for ALBPM Archiving database)
   - `bam/` (for ALBPM Business Activity Monitoring database)
   - `dataMart/` (for ALBPM DataMart database -formerly "DataStore" for historical reporting-)

2. For each ALBPM 5.5 database, follow the steps below to run the SQL migration scripts.

   a) Modify the SQL scripts to match your database schema name.

   Open each `.sql` script file with a text editor and replace the text `<schema_name>` with the actual name of your schema.

   b) Review the scripts and ensure they fit your environment.

   Each `.sql` script may include specific instructions on how it works and how to use it. Make sure you read them before executing any SQL statements.

   c) Execute the SQL scripts to upgrade the database structure.

   Important: You must run more than one SQL script on each database, in the right order. The file name of each SQL script is prefixed with a number indicating the order in which they must be executed.

   For example, for upgrading the Directory database, you must execute the following scripts, in this order:

   1. `1DisableConstraints.sql`
   2. `2AlterTables.sql`
   3. `3RebuildIndexes.sql`
   4. `4EnableConstraints.sql`

5. `5UpdateFDIVersionInfo.sql`

Important: Some scripts (like `1DisableConstraints.sql` and `4EnableConstraints.sql`) do not modify your database. Instead, they construct and return a list of SQL statements which you must capture and execute on your database.

You can use Oracle's `sqlplus` command to execute each script, following this syntax:

```
sqlplus <user_id>/<password> @<file_name>
```

Example:

```
sqlplus system/manager @2AlterTables.sql
```

Configuring Directory DB

After installing ALBPM 6.0, you must run ALBPM Admin Center to create a new Directory configuration that will connect to your existing 5.x Directory database.

Before continuing, you must shutdown all existing ALBPM 5.x applications and services, including:

- Process Engines
- All Web Applications and containers, including Process Admin, WorkSpace, WorkSpace Admin, Archive Viewer
- Admin Center
- BAM Updater
- Custom applications accessing the Directory Database

The following procedures show you how to configure the ALBPM Directory.

1. Start the Admin Center.
2. Click **Configuration** and press the **Add** button from the **Directory** tab.
   The Configuration Wizard appears.
3. Select **Use existing Directory Service**.
4. Make sure options **Create Process Engine** and **Publish and Deploy Sample Project** are **not** selected.
5. Press **Next** and select a provider type for your ALBPM Directory.

   If you are upgrading from an ALBPM WE (WorkSpace Extensions) 5.7 environment, select **Use an external directory service provider plus a database managed by ALBPM**. This will allow you to configure ALBPM with ALI Identity Service, which allows ALBPM to read ALUI's database of users.

   If you are upgrading an ALBPM environment which does not include ALBPM WE 5.7, select **Use a database managed by ALBPM**.
6. Press **Next** and enter the Directory information.

   Under **Directory Provider** select the JDBC driver corresponding to your existing Directory database. Enter the **BPM Administrator User** and **BPM Administrator Password** of your Directory.

   Note: This is **not** a new user and password. You must enter the Administrator user and password of your existing 5.x Directory. If you are upgrading from an ALBPM WE 5.7 environment, BPM Administrator User must be an existing user in ALUI within the *Process Administrator* Group. The user and password are case-sensitive.

   If you are upgrading from an ALBPM WE 5.7 environment, select **ALI Identity Service** as you **Organization Provider**.

7. Press **Next** and enter the database connectivity information to your existing Directory.

   Mind that this version of ALBPM comes with its own set of JDBC drivers from BEA.

   For advanced JDBC URL configurations refer to the custom properties of the JDBC driver:

   - For Oracle: *http://edocs.bea.com/wls/docs100/jdbc_drivers/oracle.html#*
   - For Microsoft SQL Server: *http://edocs.bea.com/wls/docs100/jdbc_drivers/mssqlserver.html*
   - For DB2: *http://edocs.bea.com/wls/docs100/jdbc_drivers/db2.html*

8. If you are upgrading from an ALBPM WE 5.7 environment, enter the information to locate the ALI Identity Service and ALUI's EDK service.

   The wizard skips this step if you selected **Use a database managed by ALBPM** on the Directory Provider Type step.

   The **Service Endpoint URL** is that of ALUI's EDK web service.

9. Press **Next** and then **Finish** to complete your Directory configuration.

   After successfully running the Configuration Wizard, the newly created directory appears in the list of available directory configurations.

10. To test connectivity to the Directory database highlight the new Directory configuration, press **Edit...** and then **Check...**.

    A confirmation message appears. If you get an error message instead, revisit all connection properties and try again.

    a) Press **OK** to close the confirmation dialog.
    b) Press **OK** again to close the **Edit Directory Configuration** dialog.

11. Press **OK** to close the **Configuration** dialog.

Starting Process Administrator

After creating a new Directory configuration pointing to your existing 5.x Directory database you must start ALBPM's Process Administrator to continue with the upgrade procedures.

1. From the main window of the Admin Center, click on **Start BPM Web Applications**.

   The Admin Center starts the Web Applications. When the Process Administrator application starts, it will automatically migrate the Directory database schema to the new version.

2. When the applications have finished starting, click on **Launch Process Administrator** to open a web browser to access the Process Administrator application.

   Note: You may also open your preferred web browser from outside Admin Center. The default URL to the Process Administrator application is: `http://localhost:8686/webconsole/`.

3. Login to Process Administrator using the same administrator user and password you used to specify for 5.x.

   You should see all the existing information and configuration data of your Directory.

   Important: Do NOT start any Process Execution Engines yet. You must first re-configure the Database properties and re-deploy your migrated projects.

4. If running the J2EE edition of ALBPM, you should now start your application servers.

   If you are upgrading ALBPM for WebLogic or WebSphere, start the Administration server and any additional server in your cluster where you will deploy the ALBPM applications.

Re-Configuring Engine DB

This version of ALBPM includes new JDBC drivers provided by BEA. You must re-configure your Process Execution Engines to use the new drivers.

Follow this procedure for each of your Process Execution Engines.

1. Login to the Process Administrator.
2. Click on **Engines ➤ YOUR_ENGINE_ID ➤ Edit Engine Database Configuration**.
3. Click on **Change the subtype** link in order to change the JDBC driver.
4. Select the BEA JDBC driver for your Database from the drop-down menu and press **Save**.

Your Execution Engine is now configured to use the new JDBC drivers.

Important: Do NOT start your Process Execution Engines yet. You must first re-deploy your migrated projects.

Validate Engine properties

Validate that the existing configuration properties of your Engine still apply for your new ALBPM 6.0 environment.

Follow this procedure for each of your Process Execution Engines.

1. Login to the Process Administrator.
2. Click on **Engines ➤ YOUR_ENGINE_ID**.
3. Go through the configuration tabs and make sure all properties are still valid on your new ALBPM 6.0 environment.

   In particular, you should change the **Home Directory** and **Log Directory**, which by default point to your previous ALBPM installation directory.

Important: Do NOT start your Process Execution Engines yet. You must first re-deploy your migrated projects.

Re-Configuring External Resources

Some External Resources (formerly "Service Endpoints") need to be re-configured for 6.0. You must change your SQL Database External Resources to use the new JDBC drivers provided by BEA.

1. Login to the Process Administrator.
2. Click on **External Resources**.
3. For each External Resource of type **SQL Database** and sub-type **Remote JDBC** you may need to change the associated J2EE configuration:
   a) Click on the External Resource ID.
   b) Change the selection of the **J2EE** drop-down from "Local configuration for ENGINE_ID Engine" (where ENGINE_ID is the name of your Process Engine) to "Local J2EE Configuration".
   c) Push **Save**.
4. For all other External Resources of type **SQL Database**:
   a) Click on the External Resource ID.
   b) Click on **Change the subtype** link in order to change the JDBC driver.
   c) Select the BEA-provided JDBC driver for your Database from the drop-down menu.
   d) Push **Save**.

5. For each External Resource of type **Web Service** you will have to force a **Save** operation in order to re-generate the configuration information:
   a) Click on the External Resource ID.
   b) Push **Save**.

All your External Resources should now be ready.

Important: Do NOT start your Process Execution Engines yet. You must first re-deploy your migrated projects.

Re-Deploying Migrated Projects

Before starting your Process Execution Engines, you must have your projects migrated to 6.0 and re-publish them to your Enterprise environment.

You must first migrate the code of all the ALBPM projects that are published. Refer to *Migrating Project Code* on page 6 for detailed procedures on how to migrate existing 5.x projects to 6.0. Also, make sure the migrated project code is compatible with the previous versions of the project.

You must follow this procedure for all ALBPM projects (and versions) you have published on your Enterprise environment.

1. Login to the Process Administrator.
2. Click on **Projects**.
3. For each project that you must re-publish:
   a) Press **Publish**.
   b) Select **Exported Project** and browse for the exported (`.exp`) project file.
   c) Check the **Republish a previous version** and **Deploy processes after publishing them** options.
   d) Press **OK** to continue with the normal publish&deploy procedures.

      The **Choose Version** page will appear
   e) On the **Version** drop-down, select the project version you are replacing and press **Next**.

      Note: If you get an "Incompatible Version" message, it means the code you are trying to publish is not compatible with the currently published code. Potential causes:

      - You accidentally broke compatibility of your project code when migrating the project code from 5.x to 6.0.
      - You saved and exported the project with a version of Studio prior to 5.7.2 before opening it in 6.0.
      - You saved and exported the project with a version of Studio prior to 6.0.1 before publishing via Process Administrator.

   f) Press **Next** to go to the **Publication Info** page.

      You will see the publication information, including the mappings to existing variables, roles and external resources.
   g) Press **Publish** to go to the **Deployment Topology** page.

      You should accept the default selections to re-deploy the project to the same Engine.
   h) Press **OK** to finish the deployment of the project.

WebSphere-Specific Upgrade Steps

If you are upgrading an ALBPM for WebSphere environment, follow these additional steps.

This procedure assumes you are modifying an existing WebSphere environment that was running ALBPM 5.x and was upgraded to a version of WebSphere supported by ALBPM 6.0 (refer to *Upgrade Requirements and limitations* on page 4 for more details). If you are configuring a clean WebSphere environment, refer to the *ALBPM Configuration Guide (WebSphere edition)*.

Install ALBPM Deployer

Install ALBPM Deployer to simplify the deployment and general management of ALBPM Enterprise applications deployed in WebSphere.

ALBPM Deployer is a j2ee application which must be installed on WebSphere's Admin server and acts as a bridge between ALBPM's Process Administrator and WebSphere. Once you have ALBPM Deployer installed and running, you can manage (start, stop, install/uninstall) your ALBPM applications directly from the ALBPM Process Administrator (without the need to use WebSphere's console).

1. Login to WebSphere's Administration Console.

   By default, it runs on `http://host:9060/ibm/console`.

2. Go to **Applications ➤ Install New Applications** and select ALBPM Deployer EAR file.

   The ALBPM Deployer consists of an EAR file provided with the installation of ALBPM Enterprise. It is located under the following directory: `BEA_HOME/albpm6.0/j2eews/j2ee/websphere/deployer`.

   Two copies are provided. You should pick one depending on whether your WebSphere environment has *Global Security* enabled:

   - `wasj2eedeployer.ear`: If security is disabled.
   - `wasj2eedeployersecured.ear`: If security is enabled.

   Note: If WebSphere's Administration Console didn't prompt you for a password at login, it probably means *Global Security* is disabled.

3. Click **Next** and review the installation options.

   Accepting the default installation options works for most cases. Always refer to WebSphere's official documentation for details.

4. Click **Next** and select the WebSphere server where to install ALBPM Deployer.

   On a single-server setup, you must install it on the only server available.

   Restriction: On a clustered WebSphere environment, ALBPM Deployer must be installed in the Deployment Manager server (*dmgr*). WebSphere's Console may not allow you to pick the Deployment Manager server as a target for installing applications; If this is the case, you must install ALBPM Deployer using other means, such as WebSphere's `wsadmin` command-line tool.

5. Click **Next** again, and **Finish** after reviewing all selected options.
   You should get confirmation that *Fuego WebSphere Deployer* application was successfully installed.
6. Save your WebSphere configuration changes.
7. Go to **Applications ➤ Enterprise Applications** and start Fuego WebSphere Deployer application.

The ALBPM WebSphere Deployer is now installed and ready.

Configure Work Manager

You need to create a Work Manager configuration in WebSphere to provide asynchronous transaction processing capabilities to ALBPM Engine.

1. Login to WebSphere's Administration Console. By default, it runs on `http://host:9060/ibm/console`.
2. Go to **Resources ➤ Asynchronous beans ➤ Work managers** and press **New** to create a new Work Manager.

> **Note:** You must select a WebSphere configuration *scope* for your new resources. You may select *cell* scope. Refer to WebSphere's official documentation for more details about its configuration scoping rules.

3. Enter the following configuration values for the new Work manager:

Option	Value
Name	ALBPM WorkManager
JNDI name	wm/albpmWorkManager

4. Press **OK** after reviewing all selected options.
5. Click on **Save** to persist your WebSphere configuration changes.

The new Work manager for ALBPM Engine is now configured.

Configure JVM Properties

You need to configure each WebSphere Server to use additional system properties when launching the Java Virtual Machine.

1. Login to WebSphere's Administration Console. By default, it runs on `http://host:9060/ibm/console`.
2. Go to **Servers ➤ Application servers**. You will see a list of all available servers.
3. Follow this procedure for each Server in which you will install ALBPM applications:
   a) Expand **Java and Process Management** and click on **Process Definition**
   b) Click on **Java Virtual Machine** then on **Custom Properties**.
   c) Add the following properties:

New Property	Value
java.awt.headless	true

Restart WebSphere Servers

After configuring all the needed resources in your WebSphere installation, you must re-start all your WebSphere Servers, Node managers and Deployment manager.

After re-starting WebSphere, you may want to check your Database connections:

1. Login to WebSphere's Administration Console. By default, it runs on `http://host:9060/ibm/console`.
2. Go to **Resources ➤ JDBC ➤ Data sources**.
3. Select your ALBPM Engine and ALBPM Directory data sources.
4. Click **Test Connection** button.
   You might want to check WebSphere's log files in case of errors.

Set ALBPM properties for WebSphere

You must configure your Process Engine with some WebSphere-related properties via ALBPM Process Administrator.

You must have an ALBPM Engine for WebSphere configured in order to define this configuration properties.

1. Login to ALBPM Process Administrator. By default, it runs on `http://host:8686/webconsole`.
2. Click on **Engines** and then click on the name of your ALBPM Engine for WebSphere.
   You should see the configuration properties for your Engine.
3. Click on the **Application Server** tab.

4. In the **JMX Engine Management Configuration** section you must specify the **Host** and **Port** of one of the WebSphere servers that will be running ALBPM Engine.

   The port number is that of the WebSphere server's SOAP Connector address. You can obtain this value from the WebSphere Administration console: go to **Servers ➤ Application servers ➤ YOUR_SERVER (e.g. server1) ➤ Ports**, and look for the value of the **SOAP_CONNECTOR_ADDRESS** property.

   For example, on a default single-node WebSphere configuration where ALBPM Process Admin is running on the same host as WebSphere you would use:

Host	localhost
Port	8880

5. Locate the **BPM Application Deployer URL** field, and change its value to match your WebSphere configuration.

   For example, on a default single-node WebSphere configuration, it would be:

   `http://localhost:9080/fuego/deployer/servlet/worker`

   The URL should be that of the WebSphere server where ALBPM Deployer is installed, which must be your WebSphere's Deployment Manager server (dmgr).

   Refer to *Install ALBPM Deployer* on page 19 for more details.

6. Change the **WebSphere Server/Cluster Name** field to match your WebSphere configuration. This is the name of the WebSphere Server or Cluster where you want to install the ALBPM applications.

   For example, on a default single-node WebSphere configuration, it would be: `server1`.

Deploy ALBPM Apps in WebSphere

After WebSphere is properly configured, you can generate and install the ALBPM applications into WebSphere servers to complete your ALBPM Enterprise configuration on WebSphere.

Prepare ALBPM Applications

Some of the ALBPM applications need additional configuration changes before deploying them into WebSphere.

1. Copy file `albpm6.0/j2eews/j2ee/websphere/was6stubs/fuegoj2ee-was6-stubs.jar` into each of following directories:

   ```
   albpm6.0/j2eews/webapps/archivingviewer/WEB-INF/lib
   albpm6.0/j2eews/webapps/feeds/WEB-INF/lib
   albpm6.0/j2eews/webapps/mobile/WEB-INF/lib
   albpm6.0/j2eews/webapps/papiws/WEB-INF/lib
   albpm6.0/j2eews/webapps/portal/WEB-INF/lib
   albpm6.0/j2eews/webapps/portaladmin/WEB-INF/lib
   albpm6.0/j2eews/webapps/webconsole/WEB-INF/lib
   albpm6.0/j2eews/webapps/workspace/WEB-INF/lib
   ```

   This is to allow client applications to connect with the EJB-based Process Engine running on WebSphere.

2. Create a new directory named `jmxextensions` under `albpm6.0/j2eews/webapps/webconsole/WEB-INF/`.
3. Copy file `WEBSPHERE_HOME/AppServer/runtimes/com.ibm.ws.admin.client_6.1.0.jar` to the `jmxextensions` directory you just created on the previous step: `albpm6.0/j2eews/webapps/webconsole/WEB-INF/jmxextensions/`.

   This is to allow the Process Administrator application to connect with the Process Engines via JMX.

Build and Deploy Applications (.ear)

The ALBPM Process Administrator allows you to bundle the ALBPM applications as `.ear` files for installation on WebSphere.

Before creating the ALBPM application archives, you must have an ALBPM Engine for WebSphere configured.

1. Login to ALBPM Process Administrator. By default, it runs on `http://host:8686/webconsole`.
2. Click on **Engines** and then click on the name of your ALBPM Engine for WebSphere.
   You should see the configuration properties for your Engine.
3. Click on the **Basic Configuration** tab and then on **J2EE Application Server Files**.

   This page allows you to (re)create the .ear files of those ALBPM applications associated with this Engine.

   Note: When you access this page, the Process Administrator gets the status of each of the applications by contacting ALBPM Deployer. You will get a warning message at the bottom of the page if there was any problem contacting ALBPM Deployer. If this is the case, make sure the **BPM Application Deployer URL** (within the **Application Server** tab) is correct and that ALBPM Deployer is up and running on WebSphere.

4. Click on the "new" icon () next to each of the applications you want to install.
5. Click on the "install" icon () next to each of the applications you want to install.

   Attention: This may take several minutes. Do not click any link on the page and do **back** in your browser until the page is automatically reloaded. When you click on the icon, ALBPM Process Administrator transfers the file over to WebSphere's Deployment Manager (by means of ALBPM Deployer) and then WebSphere goes through the application installation process.

WebLogic-Specific Upgrade Steps

If you are upgrading an ALBPM for WebLogic environment, follow these additional steps.

This procedure assumes you are modifying an existing WebLogic environment that was running ALBPM 5.x and was upgraded to a version of WebLogic supported by ALBPM 6.0 (refer to *Upgrade Requirements and limitations* on page 4 for more details). If you are configuring a clean WebLogic environment, refer to the *ALBPM Configuration Guide (WebLogic edition)*.

Create user for ALBPM Deployer

ALBPM Deployer is an application that simplifies the deployment and general management of ALBPM Enterprise applications deployed in WebLogic. You must create a special user with Administration privileges. The ALBPM Deployer application runs as this user in order to install and manage applications on WebLogic.

1. Login to WebLogic's Administration Console. By default, it runs on `http://host:7001/console`.
2. On the left panel of WebLogic Console click on **Security Realms**.
3. On the right panel, click on the name of the realm you are using (example: **myrealm**).
4. Click on the **Users and Groups** tab and then press the **New** button to add a new user.
5. Enter `FuegoWebLogicDeployer` in the **Name:** field.

   Make sure you type it exactly as shown, all in one word and respecting case.

6. Enter a new password in the **Password:** and **Confirm Password:** fields.
7. Click **OK** to save the new user.

   Note:

8. Click on the **FuegoWebLogicDeployer** user you just created and click on the **Groups** tab.
9. Add the **Administrator** groups to the user.

   To do this, highlight **Administrator** on the **Available** panel on the left and click on the right-pointing arrow. This moves **Administrator** to the **Choosen** panel on the right.
10. Click the **Save** button to finish.

Install ALBPM Deployer

Install ALBPM Deployer to simplify the deployment and general management of ALBPM Enterprise applications deployed in WebLogic.

ALBPM Deployer is a j2ee application which must be installed on WebLogic's Admin server and acts as a bridge between ALBPM's Process Administrator and WebLogic. Once you have ALBPM Deployer installed and running, you can manage (start, stop, install/uninstall) your ALBPM applications directly from the ALBPM Process Administrator (without the need to use WebLogic's console).

1. Login to WebLogic's Administration Console. By default, it runs on `http://host:7001/console`.
2. In the **Change Center** pane on the top-left, click the **Lock&Edit** button. This allows you to make changes to your WebLogic domain configuration.
3. On the left panel of WebLogic Console click on **Deployments**.
4. On the right panel, click on the **Install** button.

   The ALBPM Deployer consists of an `.EAR` file provided with the installation of ALBPM Enterprise. It is located under the following directory: `BEA_HOME/albpm6.0/j2eewl/j2ee/weblogic/deployer`.
5. On the **Location:** section, click through your drives and folders to select the location of the ALBPM Deployer `.ear` file.

   Once you get to the containing folder, select `wlj2eedeployer.ear`.
6. Click **Next**.
7. Select **Install this deployment as an application** and click **Next**.
8. On step **Select deployment targets** d you must select the Administration server of your domain (example: **AdminServer**) and click **Next**.

   Important: Do NOT select a cluster or a managed server as the deployment target. The ALBPM Deployer application must be deployed to the Admin server.
9. On step **Option Settings** you can accept the defaults and click **Next**.
10. Review the information presented on **Review your choices and click Finish** and click **Finish** to complete the installation.
11. In the **Change Center** pane on the top-left, click the **Activate Changes** button. This confirms the configuration changes to your domain configuration.

Set ALBPM properties for WebLogic

You must configure your Process Engine with some WebLogic-related properties via ALBPM Process Administrator.

You must have an ALBPM Engine for WebLogic configured in order to define this configuration properties.

1. Login to ALBPM Process Administrator. By default, it runs on `http://host:8686/webconsole`.
2. Click on **Engines** and then click on the name of your ALBPM Engine for WebLogic.
   You should see the configuration properties for your Engine.
3. Click on the **Application Server** tab.
4. In the **JMX Engine Management Configuration** section you must specify the **Host**, **Port** and **Credentials** to connect to one of the WebLogic servers that will be running ALBPM Engine.

Enter the host and port number of any of the managed WebLogic servers running ALBPM Engine. You can obtain this value from the WebLogic Administration console: click on **Environment ➤ Servers** to see a table showing all servers and their corresponding **Listening Port**.

On the **Credentials** field, enter the user id and password to connect to WebLogic. This must be a user with Administration privileges. You may use the one defined for ALBPM Deployer, as explained in *Create user for ALBPM Deployer* on page 22.

For example, on a default single-node WebLogic configuration where ALBPM Process Admin is running on the same host as WebLogic you would use:

Host	localhost
Port	7001
Credentials	weblogic / password

5. Locate the **BPM Application Deployer URL** field, and change its value to match your WebLogic configuration.

   For example, on a default single-node WebLogic configuration, it would be:

   `http://localhost:7001/fuego/deployer/servlet/worker`

   The URL should be that of the WebLogic server where ALBPM Deployer is installed, which must be your WebLogic's Administration server (AdminServer).

   Refer to *Install ALBPM Deployer* on page 23 for more details.

6. Change the **WebLogic Server/Cluster Name** field to match your WebLogic configuration. This is the name of the WebLogic Server or Cluster where you want to install the ALBPM applications.

   For example, on a default single-node WebLogic configuration, it could be your Administration server: `AdminServer`.

Build and Deploy Applications (.ear)

The ALBPM Process Administrator allows you to bundle the ALBPM applications as `.ear` files for installation on WebLogic.

Before creating the ALBPM application archives, you must have an ALBPM Engine for WebLogic configured.

1. Login to ALBPM Process Administrator. By default, it runs on `http://host:8686/webconsole`.
2. Click on **Engines** and then click on the name of your ALBPM Engine for WebLogic.
   You should see the configuration properties for your Engine.
3. Click on the **Basic Configuration** tab and then on **J2EE Application Server Files**.

   This page allows you to (re)create the .ear files of those ALBPM applications associated with this Engine.

   Note: When you access this page, the Process Administrator gets the status of each of the applications by contacting ALBPM Deployer. You will get a warning message at the bottom of the page if there was any problem contacting ALBPM Deployer. If this is the case, make sure the **BPM Application Deployer URL** (within the **Application Server** tab) is correct and that ALBPM Deployer is up and running on WebLogic.

4. Click on the "new" icon () next to each of the applications you want to install.
5. Click on the "install" icon () next to each of the applications you want to install.

   Attention: This may take several minutes. Do not click any link on the page and do **back** in your browser until the page is automatically reloaded. When you click on the icon, ALBPM Process Administrator transfers the file over to WebLogic's Deployment Manager (by means of ALBPM Deployer) and then WebLogic goes through the application installation process.

Start Execution Engines

Before starting a Process Execution Engine you must have all projects deployed to that Engine properly migrated and re-published.

Follow this procedure for each Process Execution Engines you to start.

1. Login to the Process Administrator.
2. Click on **Engines**.

   The **Engines** pane will appear, with a list of available engines. At this point, your engines should be stopped, showing a status of *Not running*.
3. Before starting the Engine for the first time after upgrading to 6.0, it is recommended you set the Engine Log severity level to **INFO** or below.
4. In the **Engines** pane, on the line of the engine you wish to start, click on the **Start** icon ().

   The engine will start. If the engine starts successfully, a status of *Ready* will be shown.

   Important: If this is the first time you start the Engine after upgrading to 6.0, you should check the Engine log file to ensure there are no errors. If the Log severity level is set to **INFO** (or lower) you should see several messages like these on the Engine log file:

   ```
    info  Migrating Schema to Build #60008  Nov 30, 2007 4:46:13 PM Engine Main <0>
   main
    info  Schema migration successful       Nov 30, 2007 4:46:13 PM Engine Main <0>
   main
   ```

   These messages correspond to the automatic migration of the Engine's database schema to the new structure.

Your Engines are now up and running. You should now log into WorkSpace and test your processes to make sure they behave as expected and all running process instances are running properly on your new 6.0 environment.

Final steps for ALBPM WE Environments

If you are upgrading from an ALBPM WE (WorkSpace Extensions) 5.7 environment, follow these additional steps.

If your ALBPM environment does not include ALBPM WE 5.7, skip these whole section.

Upgrade ALI Portlets

This section shows you how to un-install the old ALBPM WE 5.7 portlets from ALUI, and replace them with the new ALBPM 6.0 WorkSpace portlets.

1. Login to ALUI with an Administrator user.
2. Click on the **Administration** tab.
3. Click on the **Process** folder.
4. Remove community **Process Models Community**.

   Select the checkbox next to **Process Models Community** folder and then click on the delete button (red cross).
5. Click on **Portlet** and remove the following portlets:

   1. Process-Unified
   2. Process-Models

6. Click on **Web Service** and remove the following web services:
   1. Process-Unified
   2. Process-Models
7. Click on **Group** and remove the **Process Model Users** group.
8. Import file `ALBPM-60-ALI-template.pte` from your ALBPM 6.0 installation into ALUI.
   a) Click on the **Administration** tab.
   b) Select **Migration - Import** from the **Select Utility...** drop-down list.
   c) Click **Browse** and select the file to import.

      The file is located at: `ALBPM_HOME/serverpackages/ALBPM-60-ALI-template.pte`.

   d) Select the **Overwrite Remote Servers** option in the **Import Settings** section.
   e) Click the **Load Package** button.
   f) Click on **Finish** button at the top of the page.
9. Re-start **BEA ALI API Service**, **BEA ALI Identity Service** and **BEA ALI Collaboration** services.

   On Windows environments, you re-start these services from the standard Services administrative tool.
10. Re-start ALUI application server (i.e. Tomcat).

Configure ALBPM WE 5.7 Attachments

Follow this procedure only if you are upgrading from an ALBPM WE (WorkSpace Extensions) 5.7 environment.

1. Login to ALUI with an Administrator user.
2. Select the **Administration** tab.
3. Select **Process Administrator** from the **Select Utility...** drop-down list.
4. Click on **Attachments**.
5. Select the same **Attachments repository type** that you had on WE 5.7.

   If the ALBPM attachments are stored in ALI Collaboration, select the **ALI Collaboration** option. Otherwise, select **ALBPM Engine**.

   Note: If **ALI Collaboration** doesn't appear as an option, make sure the ALI Identity Service is running an properly configured. See *Configuring the ALI Identity Service Database This topic describes how to create the AquaLogic Identity Service database.* .

6. If you are using **ALI Collaboration** as your attachments repository, you must define the attachments mapping from ALBPM Processes to ALI Collaboration projects and folders.
   a) Click on the **default** tab, in the **default mapping** section.
   b) Click on the link next to the **Collaboration Server Project**.

      This opens a new window to browse for Collaboration projects. Select the project and folder where you want ALBPM to store process instance attachments.

Migrate Collab attachments from WE 5.7

Follow this procedure only if you are upgrading from an ALBPM WE (WorkSpace Extensions) 5.7 environment and want to use ALI Collaboration as the back-end storage for ALBPM processes attachments.

Important: The first time you start an upgraded Process Execution Engine in 6.0 it will migrate its database structure. Before running this attachments migration procedure, make sure your newly upgraded Engine has been started at least once in the new ALBPM 6.0 environment.

Important: To prevent data corruption, the Process Execution Engine must be stopped while running this procedure.

The migration of existing ALBPM WE 5.7 attachments to the new ALBPM 6.0 installation is automated with Ant scripts. You must install *Apache Ant* version 1.6.5 in order to run the migration script. Refer to the *ALBPM Ant Tasks* for details own using Ant with ALBPM.

1. Make sure Ant is configured and running on your environment.
2. Create a `build.xml` script file like the following example (you may also download it from *http://edocs.bea.com/albsi/docs60/resources/upgradeguide/build.xml*).

   Make sure you change all property values in the script to match your environment. Refer to the *ALBPM Ant Tasks* for details about the <fuego:migrateWE57to60> task.

```
<project name="ALBPM WE Migration" default="Migration573_60_we"
       basedir="." xmlns:fuego="antlib:fuego.tools.ant.enterprise">

  <description>
    Example script for migrating ALBPM 5.7 WE attachments to ALBPM 6.0.
  </description>

  <!-- Change these properties to match your environment -->
  <property name="bea.home" value="C:/bea/albpm6.0"/>
  <property name="fuego.basedir" value="C:/bea/albpm6.0/enterprise"/>
  <property name="engine.name" value="engineStandalone"/>

  <target name="Migration573_60_we">
    <fuego:migrateWE57to60 action="migrate_mappings" engineId="${engine.name}"
                    directoryFile="${fuego.basedir}/conf/directory.xml">

      <!-- Change these connection properties to your MS-SQLServer database with ALBPM
 WE
    5.7 attachments -->
      <msSqlServerConfiguration host="we-standalone" port="1433"
                                user="processdbinstalluser" password="password"
                             database="processdbInstall"/>

      <!-- For Oracle, use the following instead of <msSqlServerConfiguration>:
  <oracleConfiguration
    host="hostname" port="1433"
    user="processdb" password="password"
    sid="sidname" schema="processdb"/>
      -->

    </fuego:migrateWE57to60>

    <fuego:migrateWE57to60 action="migrate_attachments" engineId="${engine.name}"
                    directoryFile="${fuego.basedir}/conf/directory.xml">

      <!-- Change these connection properties to your MS-SQLServer database with ALBPM
 WE
    5.7 attachments -->
      <msSqlServerConfiguration host="we-standalone" port="1433"
                                user="processdbinstalluser" password="password"
                             database="processdbInstall"/>

      <!-- For Oracle, use the following instead of <msSqlServerConfiguration>:
  <oracleConfiguration
    host="hostname" port="1433"
    user="processdb" password="password"
    sid="sidname" schema="processdb"/>
      -->
    </fuego:migrateWE57to60>

  </target>

</project>
```

3. Open a command-line session and run the migration script using Ant.

   Example on Windows:

```
C:\>cd folder_containing_script
C:\>ant -lib c:\bea\albpm6.0\enterprise\ext -lib c:\bea\albpm6.0\enterprise\lib
```

Example on Unix:

```
$ cd directory_containing_script
$ ant -lib /bea/albpm6.0/enterprise/ext:/bea/albpm6.0/enterprise/lib
```

The script will output informational messages about its progress. Example output:

```
C:\migration>ant -lib C:\bea\albpm6.0\enterprise\lib -lib C:\bea\albpm6.0\enterprise\ext
Buildfile: build.xml

Migration573_60_we:
[fuego:migrateWE57to60] fuego base dir [C:\bea\albpm6.0\enterprise]
[fuego:migrateWE57to60] setting bea.home [C:\bea\albpm6.0]
############### Product: AquaLogic Business Process Management
############### Application: BEA AquaLogic(TM) BPM Enterprise Standalone
############### Component: enterprise
############### Release: 6.0
### bea.home: C:\bea\albpm6.0
[fuego:migrateWE57to60] Starting ALBPM WE Migration - 5.7 to 6.0 -
[fuego:migrateWE57to60] IMPORTANT: Migration task migrates Attachment data from ALBPM
 WE 5.7 ProcessDB to ALBPM WE 6.0 Engine DB.
[fuego:migrateWE57to60] IMPORTANT: ALBPM Engine DB must have been migrated to 6.0 before
 executing this task!
[fuego:migrateWE57to60] Directory file has been specified:
C:\bea\albpm6.0\enterprise\conf\directory.xml
[fuego:migrateWE57to60] Migrating to Engine 'engineStandalone'
[fuego:migrateWE57to60] Action = 'migrate_mappings'
Creating connector [ProcessDBConfiguration:SQL]
Creating connector [fuego:SQL]
Creating connector [organization:HYDROGEN]
OpenLog: verbosity level = 2
Creating connector [engineStandalone_ENGINE_DB_FUEGOLABS_ARG:SQL]
[fuego:migrateWE57to60] Migrating 1 Process-to-Project Mappings. Duplicate
Process-to-Project Mappings will be overwritten.
[fuego:migrateWE57to60] Migration Task completed successfully.
[fuego:migrateWE57to60] Starting ALBPM WE Migration - 5.7 to 6.0 -
[fuego:migrateWE57to60] IMPORTANT: Migration task migrates Attachment data from
ALBPM WE 5.7 ProcessDB to ALBPM WE 6.0 Engine DB.
[fuego:migrateWE57to60] IMPORTANT: ALBPM Engine DB must have been migrated to 6.
0 before executing this task!
[fuego:migrateWE57to60] Directory file has been specified: C:\bea\albpm6.0\enter
prise\conf\directory.xml
[fuego:migrateWE57to60] Migrating to Engine 'engineStandalone'
[fuego:migrateWE57to60] Action = 'migrate_attachments'
Creating connector [ProcessDBConfiguration:SQL]
Creating connector [fuego:SQL]
Creating connector [organization:HYDROGEN]
OpenLog: verbosity level = 2
Creating connector [engineStandalone_ENGINE_DB_FUEGOLABS_ARG:SQL]
[fuego:migrateWE57to60] Migrating 2 instance properties.
[fuego:migrateWE57to60] Migrating 4 attachments.
[fuego:migrateWE57to60] Migration Task completed successfully.

BUILD SUCCESSFUL
Total time: 36 seconds
```

Note: You may pass an additional `-v` (*verbose*) argument to the `ant` command to get more detailed output, including information about each attachment being migrated.

After running the script successfully, all attachments from your 5.7 WE environment are now available on your new ALBPM 6.0 installation. You may now re-start your Engine and login to ALBPM WorkSpace to verify your process instances contain the old attachments.

Upgrade Reference Material

What's New in ALBPM 6.0 Studio

This topic provides an overview of the main new features, improvements and changes in this release of AquaLogic BPM Studio.

Standards Support

- Process models in ALBPM are now compliant with the XPDL 2.0 standard.
- Support for BPEL 2.0. You can import BPEL 2.0 models into an ALBPM Project, and new models can be designed within ALBPM Studio. The Process Execution Engine is now capable of executing BPEL 2.0 natively.
- ALBPM Studio application is now built on top of the Eclipse platform.

Studio IDE

- Studio now includes a software agent for automatic problem reporting and feedback. In case of unexpected errors in Studio, an automatic report will be sent to BEA for analysis. Studio will prompt you for approval before enabling this feature. We also encourage you to send us feedback using the **Help ➤ Feedback...** menu option.
- When you first start ALBPM Studio, you have to select one of the available profiles according your skill set: Business Analyst, Business Architect, Developer. ALBPM Studio presents a different subset of features depending on the selected profile. This keeps the user interface uncluttered, hiding what you don't need. All available features are visible under the Developer profile. The on-line documentation in Studio is also filtered depending on the active profile. To switch profiles go to ***Help ➤ Welcome***.
- This new release introduces the concept of Project Variables, replacing the External and Business Variables of previous versions. Project Variables are functionally equivalent to the old External Variables but are simpler to use: they are available to all processes in the project, with no need to "promote" them from External to Instance. When the new property **Business indicator** is enabled, Project Variables behave as the old Business Variables (they are used for BAM reporting).
- ALBPM project directories do not use the .fpr extension anymore.
- The Organization data and Simulation definitions are now accessed as nodes in the project tree.
- On previous version of Studio the Business Parameters of the project were accessible from the Variables panel on right. Now you access them from the Business Parameters node under the Organization node of the project tree.
- Integration with Version Control System feature (VCS) was re-implemented to leverage the Eclipse platform. This paves the way for supporting virtually any Source Control systems compatible with Eclipse.
- Each resource that is independently stored as part of an ALBPM Project is modified using an "Editor" tabbed panel, and you must explicitly save your changes on each resource with File > Save . For example, on earlier versions of Studio you add or modify a Participant using a separate dialog window. Now a special Participants editor opens in a new tab of the edition area. This makes it easier to work with Version Control systems, as each resource is managed and saved independently.
- Some editors may open nested editors (accessible via smaller tabs at the bottom of the editor). For example, the editor for Process models uses independent sub-tabs for the process diagram and for each opened process method.

Process Designer

- You can now open several projects at the same time. Before opening a project, you first need to add it to your Studio workspace.
- Incremental compilation: There is no need for Publish&Deploy anymore. Once you start Studio's Process Execution Engine, the project is running. While it is running, the Execution Engine immediately applies changes you make to your project design and code.

- A new type of Interactive activity: Decision activities. This type of activity allows the end user to decide the next path a process instance will take (one of the possible outgoing transitions), based on the value of certain instance variables. The Process Execution Engine keeps track of those decisions over time and presents the end user with recommendations on what decision to take based on past experience.
- Business Rules: ALBPM Studio now provides a way of defining business rules using a graphical rules editor, without requiring any coding. After the project is deployed, authorized end users can also modify these rules on-the-fly, while the processes are executing. They can do so right from the ALBPM WorkSpace UI.
- Round-trip Simulation: You can now create Simulation models from the actual execution of the processes during a given period of time. This makes it easier to create realistic Simulation models.

User Interface

- ALBPM WorkSpace has been re-designed and re-implemented from the ground up. It is based on a modern modular architecture which makes it easier to customize and integrate naturally with AquaLogic UI and WebLogic Portal. The old WorkSpace is still provided for backward compatibility but may be removed in future versions.
- Dashboards provide better quality graphics and end user interaction (i.e. rotation, detaching of pie sections).

Integration

- Native integration with ALSB. You can now easily consume ALSB services from ALBPM and also register a business process in ALSB.
- Web Services in ALBPM now include support for WS-Security, Document-Literal style and WS-I compliance.
- ALBPM Studio now includes JDBC drivers for the most popular DBMS. This means you can integrate with Oracle, DB2 and Microsoft SQL Server right out of the box.
- PAPI has deprecated several methods in favor of new ones which follow a new naming convention. PAPI methods which where deprecated in ALBPM 5.7 have been deleted from the API.
- PAPI WebService 1.0 has been deprecated in favor of the new PAPI WebService 2.0. PAPI-WS 1.0 is accessible through ALBPM WorkSpace while PAPI-WS 2.0 is accessible through its own new Web Application (papiws). This new version is functionally equivalent to the native Java PAPI, and adheres to the WS-Security specification using the UserNameToken Profile implementation as well as HTTP Basic Authentication.

What's New in ALBPM 6.0 Enterprise

This section provides an overview of the main new features, improvements and changes in this release of AquaLogic BPM Enterprise (all editions).

New features

- There is a new Configuration Wizard which provides a simple way for configuring a complete ALBPM Enterprise installation. It covers the most common configuration tasks, including: creation of Directory Service database, creation of Engine definition and database, complete creation of WebLogic domain, and deployment of a sample project.
- ALBPM Directory Service can be configured in a *Hybrid* configuration where authentication and authorization can be delegated to Microsoft Active Directory or Sun One Directory Service while the rest of the metadata resides in a transactional RDBMS. This avoids the need for replication of participants and entitlements data.
- ALBPM now includes JDBC drivers for the most popular DBMS. This means you can integrate with Oracle, DB2 and Microsoft SQL Server right out of the box.
- ALBPM RSS Feeds Web Application allows end users to participate in business processes using their RSS Reader of choice being able to authentication and register with a specific view RSS Feed. Each View accessible through WorkSpace can be accessed from an RSS Reader like Outlook.
- PAPI WebService 1.0 has been deprecated in favor of the new PAPI WebService 2.0. PAPI-WS 1.0 is accessible through ALBPM WorkSpace while PAPI-WS 2.0 is accessible through its own new Web Application (papiws). This

new version is functionally equivalent to the native Java PAPI, and adheres to the WS-Security specification using the UserNameToken Profile implementation as well as HTTP Basic Authentication.
- Native integration with AquaLogic Service Bus 2.6. You can now easily consume ALSB services from ALBPM and also register a business process in ALSB. In addition, a Custom Transport has been implemented over RMI to enforce security and transaction propagation when ALSB and ALBPM run on the same domain. This transport is provided as an Enterprise Application (.ear) which serves as a plugin for ALSB.
- Web Services in ALBPM now include support for WS-Security, Document-Literal style and WS-I compliance.

Changes and Improvements

- Configuring ALBPM Directory Service purely on top of LDAP provider is no longer possible.
- ALBPM project directories do not use the `.fpr` extension anymore.
- The Directory configuration file `directory.properties` is now XML-based and changes its name to `directory.xml`.
- ALBPM JSR-168 Portlets have been deprecated in favor of the new WorkSpace deployable in WebLogic Portal (WLP) 10.0 and ALUI 6.1MP1. This new interface matches the WorkSpace functionality as well as supporting SSO.
- It is now simpler to deploy ALBPM projects on WebLogic and WebSphere editions of ALBPM Engine. It is no longer necessary to generate and deploy an EAR file for each ALBPM project. Only the Engine EAR needs to reside in the J2EE container. The Engine application dynamically loads the project models and executable code from the Directory Service.
- ALBPM WorkSpace Extensions (for integration with ALUI) is now included with ALBPM Enterprise. This consolidates the installation and setup on a single install package.

Removed and Deprecated Components

Some standard components of the catalog are no longer available on this version, and others are now deprecated.

Removed and Deprecated components

The following table lists the standard components and methods which have been removed and those which have been flagged as deprecated on each version of ALBPM.

Module.SubModule.Component.method()	Status on 5.5	Status on 5.7	Status on 6.0
`Int.intValue()`			Deprecated
`Int.longValue()`			Deprecated
`Fuego.Net.JSP`		Deprecated	Deprecated
`Fuego.Ui.FileChooser`			Deprecated
`Fuego.Io.ClientFile.listFiles()`		Deprecated	Removed
`Fuego.Io.DelimitedFile.DelimitedFile(String name, char fieldSeparator, String lineSeparator)`		Deprecated	Removed
`Fuego.Io.DelimitedFile.DelimitedFile(String name, boolean append, char fieldSeparator, String lineSeparator)`		Deprecated	Removed
`Fuego.Io.TextFile.TextFile(String name, String lineSeparator)`		Deprecated	Removed
`Fuego.Io.TextFile.TextFile(String name, boolean append, String lineSeparator)`	Deprecated	Deprecated	Removed

Module.SubModule.Component.method()	Status on 5.5	Status on 5.7	Status on 6.0
`Fuego.Util.Program.dispose(int handle)`	Deprecated	Deprecated	Removed
`Fuego.Util.Program.isAlive(int handle)`	Deprecated	Deprecated	Removed
`Fuego.Util.Program.public void kill(int handle)`	Deprecated	Deprecated	Removed
`Fuego.Util.Program.run(String name)`	Deprecated	Deprecated	Removed
`Fuego.Util.Program.waitFor(int handle)`	Deprecated	Deprecated	Removed

Upgrade Scripts for 5.5 on SQL Server

ALBPM Enterprise 6.0.3 (on the first Hotfix) includes SQL scripts for upgrading the structure of ALBPM 5.5 databases on Microsoft SQL Server. These scripts are located under the following directory: `<ALBPM_DIR>/conf/migration/ms-sql/`.

Here is the list of files included for upgrading ALBPM 5.5 Microsoft SQL Server databases to ALBPM 6.0:

```
ms-sql/
|-- ArchivingDatabaseDifferencesReport5511to602.html
|-- BamDatabaseDifferencesReport5511to602.html
|-- DataMartDatabaseDifferencesReport5511to602.html
|-- EngineDatabaseDifferencesReport5511to602.html
|-- FDIDatabaseDifferencesReport5511to602.html
|-- migration_archiving.sql
|-- migration_bam.sql
|-- migration_dataMart.sql
|-- migration_directory.sql
`-- migration_engine.sql
```

The `.sql` files are the SQL scripts you must execute for converting the 5.5 schemas to 6.0.2, as described in *Upgrade ALBPM 5.5 SQL Server DBs* on page 13.

The `.html` files are reports of the schema differences between version 5.5 and 6.0, included for your reference.

Upgrade Scripts for 5.5 on Oracle

ALBPM Enterprise 6.0.3+ (on the first Hotfix) includes SQL scripts for upgrading the structure of ALBPM 5.5 databases on Oracle. These scripts are located under the following directory: `<ALBPM_DIR>/conf/migration/oracle/`.

The scripts are grouped in sub-directories for each ALBPM database: archiving, bam, dataMart, engine, directory.

Here is the list of files included for upgrading ALBPM 5.5 Oracle databases to ALBPM 6.0:

```
oracle/
|-- archiving/
|   |-- 0ReadmeFirst.txt
|   |-- 1AlterTables.sql
|   |-- 2RebuildIndexes.sql
|   |-- ArchivingDatabaseDifferencesReport5511to602.html
|   `-- OracleArchiving602DatabaseSchema.sql
|-- bam/
|   |-- 0ReadmeFirst.txt
|   |-- 1DisableConstraints.sql
|   |-- 2AlterTables.sql
```

```
|   |-- 3RebuildIndexes.sql
|   |-- 4EnableConstraints.sql
|   |-- 5UpdateExternalVariablesIfAny.sql
|   |-- BamDatabaseDifferencesReport5511to602.html
|   `-- OracleBam602DatabaseSchema.sql
|-- dataMart/
|   |-- 0ReadmeFirst.txt
|   |-- 1DisableConstraints.sql
|   |-- 2AlterTables.sql
|   |-- 3RebuildIndexes.sql
|   |-- 4EnableConstraints.sql
|   |-- 5UpdateExternalVariablesIfAny.sql
|   |-- DataMart602DatabaseSchema.sql
|   `-- DataMartatDabaseDifferencesReport5511to602.html
|-- directory/
|   |-- 0ReadmeFirst.txt
|   |-- 1DisableConstraints.sql
|   |-- 2AlterTables.sql
|   |-- 3RebuildIndexes.sql
|   |-- 4EnableConstraints.sql
|   |-- 5UpdateFDIVersionInfo.sql
|   |-- FDIDatabaseDifferencesReport5511to602.html
|   `-- OracleFDI602DatabaseSchema.sql
`-- engine/
    |-- 0ReadmeFirst.txt
    |-- 1AlterTables.sql
    |-- 2RebuildIndexes.sql
    |-- 3UpdateExternalVariablesIfAny.sql
    |-- ENGINEDatabaseDifferencesReport5511to602.html
    `-- OracleEngine602DatabaseSchema.sql
```

The `0ReadmeFirst.txt` files include additional instructions about the scripts and how to run them.

The `.sql` files prefixed with a number are the SQL scripts you must execute for converting the 5.5 schemas to 6.0.2, as described in *Upgrade ALBPM 5.5 Oracle DBs* on page 14.

Important: You must run more than one SQL script on each database, in the right order. The file name of each SQL script is prefixed with a number indicating the order in which they must be executed.

For each database, a complete SQL script describing the 6.0 schema structure is also included for your reference.

The `.html` files are reports of the schema differences between version 5.5 and 6.0, included for your reference.