

REMOVE-Upgrading from WSO2 Message Broker

This page walks you through the process of upgrading to WSO2 Enterprise Integrator (WSO2 EI) 6.6.0 from WSO2 Message Broker (WSO2 MB) 3.1.0.



Before you begin, note the following:

- If you are upgrading from a version older than WSO2 MB 3.1.0, you should first upgrade to WSO2 MB 3.1.0 and then upgrade to WSO2 EI 6.6.0.
- For information on what is new in this release and why you should upgrade, see [About this Release](#).
- For more information on ports, see [Default Ports of WSO2 Enterprise Integrator](#).
- The distribution folder structure has changed from WSO2 MB 3.1.0 to WSO2 EI 6.6.0.

Message Broker 3.1.0	Enterprise Integrator 6.6.0
<MB_HOME>/repository/conf	<EI_HOME>/wso2/broker/conf
<MB_HOME>/repository/conf/axis2	<EI_HOME>/wso2/broker/conf/axis2
<MB_HOME>/repository/conf/datasources	<EI_HOME>/wso2/broker/conf/datasources
<MB_HOME>/repository/components/dropins	<EI_HOME>/dropins
<MB_HOME>/repository/components/extensions	<EI_HOME>/extensions
<MB_HOME>/repository/components/lib	<EI_HOME>/lib
<MB_HOME>/repository/components/patches	<EI_HOME>/patches

See the following topics for instructions:

- [Preparing to upgrade](#)
- [Upgrading the databases](#)
 - [Upgrading the Carbon database](#)
 - [Upgrading the broker-specific database](#)
- [Migrating the configurations](#)
- [Migrating tenant artifacts](#)
- [Configuring the JMS client](#)
- [Starting the server](#)

Preparing to upgrade

The following prerequisites must be completed before upgrading:

- **Be sure** to stop the publishers that are connected to WSO2 MB 3.1.0 before commencing the migration process.
 - Create a backup of the databases in your WSO2 MB 3.1.0 instance.
 - Copy the <MB_HOME> directory to back up the product configurations.
 - Download the product installer from [here](#) and install WSO2 EI 6.6.0.
- Let's call the installation location of your product the <EI_HOME> directory. This is located in a place specific to your OS as shown below:

OS	Home directory
Mac OS	/Library/WSO2/EnterpriseIntegrator/6.6.0
Windows	C:\Program Files\WSO2\EnterpriseIntegrator\6.6.0\
Ubuntu	/usr/lib/wso2/EnterpriseIntegrator/6.6.0
CentOS	/usr/lib64/EnterpriseIntegrator/6.6.0



The downtime is limited to the time taken for switching databases in the production environment.

Upgrading the databases

See the following topics for information on upgrading the databases of WSO2 MB 3.1.0.

Upgrading the Carbon database

In WSO2 EI 6.6.0, you can use the same [Carbon database](#) that you used for MB 3.1.0. Therefore, you can simply restore the backup of the existing Carbon database to use with WSO2 EI 6.6.0.

Upgrading the broker-specific database

The Message Broker profile of WSO2 EI comes with several changes to the [broker-specific database](#), and therefore you must upgrade this database as explained below. We are providing a simple tool that you can easily download and run to carry out this upgrade. Follow the steps given below.

1. Disconnect all the subscribers and publishers of WSO2 MB 3.1.0.
2. Shut down the server.
3. Run the migration script to update the database:
 - a. Open a terminal and navigate to the `<EI_HOME>/wso2/broker/dbscripts/mb-store/migration-3.1.0_to_3.2.0` directory, which contains scripts for each database type.
 - b. Run the migration script relevant to your database type. For example, if you are using Oracle, use the following script: `oracle-mb.sql`.
4. Download and run the migration tool:
 - a. Download the [migration tool](#).
 - b. Unzip the `org.wso2.mb.migration.tool-2.0.zip` file. The directory structure of the unzipped folder is as follows:

```
TOOL_HOME
|-- lib <folder>
|-- config.properties <file>
|-- tool.sh <file>
|-- README.txt <file>
|-- org.wso2.carbon.mb.migration.tool.jar
```

- c. Download the relevant database connector and copy it to the `lib` directory in the above folder structure. For example, if you are upgrading your MySQL databases, you can download the MySQL connector JAR from <http://dev.mysql.com/downloads/connector/j/5.1.html> and copy it to the `lib` directory.
- d. Open the `config.properties` file from the folder structure shown above and update the database connection details shown below.

```
#Configurations for the database
dburl=
driverclassname=
dbuser=
dbpassword=
```

The parameters in the above file are as follows:

dburl	The URL for your broker-specific database. For example, <code>jdbc:mysql://localhost/wso2_mb</code> .
driverclassname	The database driver class. For example, <code>com.mysql.jdbc.Driver</code> for MySQL.
dbuser	The user name for connecting to the database.
dbpassword	The password for connecting to the database.

- e. Run the migration tool:
 - i. If you are on a Linux environment, open a command prompt and execute the following command: `tool.sh`.
 - ii. If you are on a non-Linux environment, execute the `org.wso2.carbon.mb.migration.tool.jar` manually.

The database is now upgraded with the changes relevant to the Message Broker profile of WSO2 EI 6.6.0.

Migrating the configurations

To migrate all the required folders, files, libraries, etc. from WSO2 MB 3.1.0 to WSO2 EI 6.6.0:



Do not copy **configuration files** directly between servers. Instead, [update the files manually](#).

1. Copy the database connector JAR files stored in the `<MB_HOME>/repository/components/lib` directory to the `<EI_HOME>/lib` directory of WSO2 EI 6.6.0.
2. Copy the keystores and truststores from the `<MB_HOME>/repository/resources/security` directory to the `<EI_HOME>/wso2/broker/repository/resources/security` directory in WSO2 EI 6.6.0.
3. If you have secondary user stores created for WSO2 MB 3.1.0, you need to copy the 'userstore' folder in the `<MB_HOME>/repository/deployment/server/` directory to the `<EI_HOME>/wso2/broker/repository/deployment/server` directory in WSO2 EI 6.6.0.

To migrate the configurations from WSO2 MB 3.1.0 to WSO2 EI 6.6.0:

1. Update the configuration files with information of the migrated keystores and truststores. See [Configuring Keystores in WSO2 products](#) for more information.

2. Go to the `<EI_HOME>/wso2/broker/conf/datasources` directory and update the Carbon datasource configuration in the `master-datasources.xml` file. See [Changing the Carbon Database](#) for instructions.
3. Update the configurations related to the broker-specific database in the `master-datasources.xml` file and other related configurations files. See [Changing the Default Broker Database](#) for instructions.
4. Go to the `<EI_HOME>/wso2/broker/conf` directory and update the datasource references in the `user--mgt.xml` and `registry.xml` files to match the updated configurations in the `master--datasources.xml` file. The instructions are available in [Changing the Carbon Database](#).
5. Check for any further configurations that were done for WSO2 MB 3.1.0 based on your solution. For example, update the following configurations in the Message Broker profile of WSO2 EI 6.6.0:

- a. `broker.xml`
- b. `metrics.xml`
- c. `metrics-properties.xml`
- d. `messaging-event-broker.xml`
- e. Check the configurations related to external user stores, caching, mounting, transports etc.

WSO2 EI 6.6.0 is based on Carbon Kernel 4.5.0, which introduces **log4j2**. Also, the **carbon.logging** jar is not packed with the EI 6.6.0 distribution and the **pax-logging-api** is used instead. Follow the instructions given below to migrate from **log4j** (in WSO2 MB 3.1.0) to **log4j2** (in WSO2 EI 6.6.0).

1. If you have used a custom log4j component in WSO2 MB 3.1.0, apply the following changes to your component:
 - a. Replace carbon logging or commons.logging dependencies with pax-logging dependency as shown below.

```
<!-- Pax Logging -->
<dependency>
  <groupId>org.ops4j.pax.logging</groupId>
  <artifactId>pax-logging-api</artifactId>
  <version>${pax.logging.api.version}</version>
</dependency>

<!-- Pax Logging Version -->
<pax.logging.api.version>1.10.1</pax.logging.api.version>
```

- b. If log4j dependency is directly used, apply one of the options given below.

Option 1

Replace the log4j dependency (shown below) with log4j2 and rewrite the loggers accordingly.

```
<dependency>
  <groupId>org.ops4j.pax.logging</groupId>
  <artifactId>pax-logging-log4j2</artifactId>
  <version>${pax.logging.log4j2.version}</version>
</dependency>
```

Option 2

Replace the log4j dependency with pax-logging dependency and rewrite the loggers using `commons.logging` accordingly.

- c. If commons.logging is imported using Import-Package add the version range.

```
org.apache.commons.logging;
version="${commons.logging.version.range}"
<commons.logging.version.range>[1.2.0,2.0.0)</commons.logging.version.range>
```

2. Follow the [instructions on configuring log4j2](#) to register the Appenders and Loggers.

Migrating tenant artifacts

If multitenancy is used, copy the tenant artifacts from the `<MB_HOME>/repository/tenants` directory to the `<EI_HOME>/wso2/broker/repository/tenants` directory of WSO2 EI 6.6.0.

Configuring the JMS client

To be able to connect the queues, topics, and durable topic subscribers to the Message Broker profile, change the AMQP transport port of the JMS client to 5675.

Starting the server

Once you have completed the migration, you can start the Message Broker profile of WSO2 EI. For details see [Running the Product](#).