REMOVE-Upgrading from WSO2 Enterprise Service Bus

This page walks you through the process of upgrading to WSO2 Enterprise Integrator (WSO2 EI) 6.6.0 from WSO2 Enterprise Service Bus (WSO2 ESB) 5.0.0.

Before you begin, note the following:

- If you are upgrading from a version older than ESB 5.0.0, you should first upgrade to ESB 5.0.0 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 ESB 5.0.0 to WSO2 EI 6.6.0:

Enterprise Service Bus 5.0.0	Enterprise Integrator 6.6.0
`<ESB_HOME>/repository/conf`	`<EI_HOME>/conf`
`<ESB_HOME>/repository/conf/axis2`	`<EI_HOME>/conf/axis2`
`<ESB_HOME>/repository/conf/datasources`	`<EI_HOME>/conf/datasources`
`<ESB_HOME>/repository/components/dropins`	`<EI_HOME>/dropins`
`<ESB_HOME>/repository/components/extensions`	`<EI_HOME>/extensions`
`<ESB_HOME>/repository/components/lib`	`<EI_HOME>/lib`
`<ESB_HOME>/repository/component/patches`	`<EI_HOME>/patches`
`<ESB_HOME>/repository/components/plugins`	`<EI_HOME>/wso2/components/plugins`

See the following topics for instructions:

Preparing to upgrade

The following prerequisites must be completed before upgrading:

- Create a backup of the databases in your WSO2 ESB 5.0.0 instance.
- Copy the `<ESB_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

You can use the same databases that you used for WSO2 ESB 5.0.0 with WSO2 EI 6.6.0. However, you need to apply the changes discussed below.

This database update is required for the following reasons:

- As per the **Security Advisory (WSO2-2017-0345)**, WSO2 EI 6.6.0 uses OAEP for data encryption in addition to the RSA algorithm (which is used in WSO2 ESB 5.0.0). Therefore, the internally-encrypted data in your current databases (such as datasource configurations, syslog passwords, user store configurations, keystore registry entries, service security policies, event publisher configurations, event receiver configurations, and event sink configurations), as well as data encrypted using secure vault (such as plain text passwords in configuration files and synapse configurations) should be re-encrypted using OAEP.
- WSO2 EI 6.6.0 requires Message Processor Tasks to be stored in the registry using a **new** naming convention. The naming convention has changed from `TASK_PREFIX + messageProcessorName + taskNumber` (in WSO2 ESB 5.0.0) to `TASK_PREFIX + messageProcessorName + SYMBOL_UNDERSCORE + taskNumber` (in WSO2 EI 6.6.0).
  When you run the following migration client, the existing **Task** references in the registry will be removed. Later in this migration guide, when you migrate the integration artifacts from your ESB 5.0.0 instance to EI 6.6.0, new records will be created in the registry with the new naming convention.

- **To re-encrypt all internally-encrypted data using OAEP:**

1. Get the latest WUM updates (later than the update level released on 18/04/2018) for your WSO2 ESB 5.0.0. This will give you a new WSO2 ESB 5.0.0 distribution with the latest updates.
2. Connect the WUM-updated ESB distribution to your existing databases (which are used for registry data, and user management data):

   a. Open the `master-datasources.xml` file (stored in the `<WUM_UPDATED_ESB_HOME>/repository/conf/datasources/` directory) and update the parameters given below.

      By default, registry and user management data are stored in one database and is configured in the `master-datasources.xml` file. If you have separate databases for registry and user management data, you may have separate datasource configurations.

Element	Description
url	The URL of the database.
username and **password**	The name and password of the database user.
driverClassName	The class name of the database driver.

   b. Open the `registry.xml` file (stored in the `<WUM_UPDATED_ESB_HOME>/repository/conf` directory) and specify the datasource name of the registry database.

      ```
      <dbConfig name="wso2registry">
          <dataSource>jdbc/MY_DATASOURCE_NAME</dataSource>
      </dbConfig>
      ```

   c. If a JDBC user store is used in your ESB, open the `user-mgt.xml` file (stored in the `<WUM_UPDATED_ESB_HOME>/repository/conf/` directory), and update the following database connection parameters under the `<UserStoreManager class="org.wso2.carbon.user.core.jdbc.JDBCUserStoreManager">` section.

Element	Description
url	The URL of the database.
username and **password**	The name and password of the database user.
driverClassName	The class name of the database driver.

      Further, update the system administrator configurations and the datasource name in the `user-mgt.xml` file.

   d. Encrypt the plain text passwords that you added to the configuration files (`master-datasources.xml`, `user-mgt.xml`, etc.).

3. Be sure that the `carbon.properties` file is included in the `<WUM_UPDATED_ESB_HOME>/repository/conf/` directory with the following parameter:

   ```
   org.wso2.CipherTransformation=RSA/ECB/OAEPwithSHA1andMGF1Padding
   ```

4. Start the WUM-updated ESB server. This will re-encrypt the data in the databases.

- **To re-encrypt plain text strings using OAEP:**

1. First of all you need to execute the cipher tool script from the command prompt using the command relevant to your OS:

- On Windows: `./ciphertool.bat -Dconfigure`
- On Linux: `./ciphertool.sh -Dconfigure`

2. Connect the ESB profile of WSO2 EI to your existing databases (which are used for registry data and user management data):

    a. Open the `master-datasources.xml` file (stored in the `<EI_HOME>/conf/datasources/` directory) and update the parameters given below.

    > By default, registry and user management data are stored in one database and is configured in the `master-datasources.xml` file. If you have separate databases for registry and user management data, you may need separate datasource configurations.

Element	Description
url	The URL of the database.
username and **password**	The name and password of the database user.
driverClassName	The class name of the database driver.

    b. Open the `registry.xml` file (stored in the `<EI_HOME>/conf` directory) and specify the datasource name.

    ```
    <dbConfig name="wso2registry">
        <dataSource>jdbc/MY_DATASOURCE_NAME</dataSource>
    </dbConfig>
    ```

    c. If a JDBC user store is used, open the `user-mgt.xml` file (stored in the `<EI_HOME>/conf/` directory), and update the following database connection parameters under the `<UserStoreManager class="org.wso2.carbon.user.core.jdbc.JDBCUserStoreManager">` section.

Element	Description
url	The URL of the database.
username and **password**	The name and password of the database user.
driverClassName	The class name of the database driver.

    Further, update the system administrator configurations and the datasource name in the `user-mgt.xml` file.

3. The keystores for WSO2 EI 6.6.0 need to be setup and configured in order to perform this data re-encryption.
    a. Migrate the keystores and truststores from WSO2 ESB 5.0.0 to WSO2 EI 6.6.0 by copying the files from the `<ESB_HOME>/repository/resources/security` directory to the same directory in WSO2 EI 6.6.0.
    b. Open the `carbon.xml` file (stored in the `<EI_HOME>/conf/` directory) and update the details of the keystore used for data encryption.

Update the following configuration element:

```
<KeyStore>
    <Location>${carbon.home}/resources/security/wso2carbon.jks</Location>
    <Type>JKS</Type>
    <Password>wso2carbon</Password>
    <KeyAlias>wso2carbon</KeyAlias>
    <KeyPassword>wso2carbon</KeyPassword>
</KeyStore>

<TrustStore>
    <!-- trust-store file location -->
    <Location>${carbon.home}/repository/resources/security/client-truststore.jks<
/Location>
    <!-- trust-store type (JKS/PKCS12 etc.) -->
    <Type>JKS</Type>
    <!-- trust-store password -->
    <Password>wso2carbon</Password>
</TrustStore>
```

Add the following configuration element under <Security> in the carbon.xml file, and update the values:

```
<InternalKeyStore>  <Location>${carbon.home}/repository/resources/security/internal.
jks</Location>
  <Type>JKS</Type>
  <Password>wso2carbon</Password>
  <KeyAlias>wso2carbon</KeyAlias>
  <KeyPassword>wso2carbon</KeyPassword>
</InternalKeyStore>
```

See Configuring Keystores in WSO2 Products for more information.

4. Create the <EI_HOME>/migration/ directory, copy the migration-conf.properties file, and update the following values:

keystore.identity. location	The location of the keystore that is used for data encryption in WSO2 EI 6.6.0. By default, this is <EI_HOME> /repository/resources/security/wso2carbon.jks.
keystore.identity. key.password	The key password of the keystore. By default, this is wso2carbon.
admin.user.name	The user name of the system administrator.

5. Copy the migration JAR file to the <EI_HOME>/dropins/ directory.
6. Start the ESB profile of WSO2 EI:

    a. Open a terminal and navigate to the <EI_HOME>/bin/ directory.
    b. Execute the product start up script with the '-Dmigrate.from.product.version=esb' command as shown below.

On MacOS/Linux/CentOS
Open a terminal and execute the following command:

```
sh integrator.sh -Dmigrate.from.product.version=esb
```

On Windows
Open a terminal and execute the following command:

```
integrator.bat -Dmigrate.from.product.version=esb
```

The relevant data is now re-encrypted and records of any scheduled tasks are deleted from the registry.

7. Once the migration is successful, stop the server and delete the migration JAR (org.wso2.carbon.ei.migration-6.6.0.jar) fro m the <EI_HOME>/dropins/ directory.
8. You can now connect your WSO2 EI 6.6.0 instance to the old database. This is discussed under Migrating ESB configurations.

Migrating the ESB

Follow the instructions given below to upgrade from WSO2 ESB 5.0.0 to WSO2 EI 6.6.0.

Migrating ESB configurations

Do not copy **configuration files** directly between servers. Instead, update the files manually.

To migrate all the required folders, files, libraries, etc. from WSO2 ESB 5.0.0 to the ESB profile of WSO2 EI 6.6.0:

1. Copy the database connector JAR files stored in the `<ESB_HOME>/repository/components/lib` directory to the `<EI_HOME>/lib` directory.
2. You need to migrate the keystores and truststores used in WSO2 ESB 5.0.0 from the `<ESB_HOME>/repository/resources/security` directory to the same directory in WSO2 EI 6.6.0.

   

   Note that you have already completed this keystore migration before upgrading the ESB databases.

3. If you have secondary user stores created for WSO2 ESB 5.0.0, you need to copy the 'userstore' folder in the `<ESB_HOME>/repository/deployment/server/` directory to the same directory in WSO2 EI 6.6.0.
4. If there are any third-party libraries used with WSO2 ESB 5.0.0 that you want to migrate, copy the relevant libraries to WSO2 EI 6.6.0:
   - If you have used JMS libraries, JDBC libraries, etc., copy the files from the `<ESB_HOME>/repository/components/lib` directory to the `<EI_HOME>/lib` directory.
   - If you have used OSGi bundles such as SVN kit etc., copy the contents of the `<ESB_HOME>/repository/components/dropins` directory to the `<EI_HOME>/dropins` directory.

To migrate the configurations from WSO2 ESB 5.0.0 to the ESB profile of WSO2 EI 6.6.0:

Before you begin, note the following changes that affect your configuration files:

- HTTP content negotiation is enabled in the ESB profile since WSO2 EI 6.0.0. This setting transforms the message (at the time of building the message) according to the content type specified using the 'Accept' header of the request message. You can disable this behavior by setting the following parameter to 'false' in the `axis2.xml` file (stored in the `<EI_HOME>/conf/axis2/` directory). Read more about Working with Message Builders and Formatters .

  ```
  <parameter name="httpContentNegotiation">false</parameter>
  ```

- When an XML payload converts to JSON in WSO2 EI 6.6.0, null and empty values are handled as explained in Working with JSON Message Payloads.
- The following configuration files have changed from WSO2 ESB 5.0.0 to WSO2 EI 6.6.0.
  - `axis2.xml`
  - `axis2_nhttp.xml`
  - `axis2_pt.xml`
  - `tenant-axis2.xml`
  - `cache.xml`
  - `config-validation.xml`
  - `logging-bridge.properties`
  - `osgi-debug.option`
  - `cloud-services-desc.xml`
  - `authenticators.xml`
  - `ciper-tool.properties`
  - `catalina-server.xml`
  - `carbon.xml`
  - `identity.xml`
  - `nhttp.properties`
  - `passthru-http.properties`
  - `synapse.properties`
  - `synapse-handlers.xml`
  - `user-mgt.xml`
  - `jndi.properties`

1. Update the configuration files with information of the migrated keystores and truststores. See Configuring Keystores in WSO2 Products for more information. **Note** that some keystore configurations were already updated before upgrading the ESB databases.
2. Go to the `<EI_HOME>/conf/datasources` directory and update the Carbon datasource configuration in the `master-datasources.xml` file. **Note** that some configurations in this file were already updated before upgrading the ESB databases. For instructions, see Changing the Carbon Database and select your database type.
3. Go to the `<EI_HOME>/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. **Note** that some configurations in these files were already updated before upgrading the ESB databases. The instructions are available in Changing the Carbon Database .

4. Check for any other configurations that were done for WSO2 ESB 5.0.0 based on your solution, and update the configurations in WSO2 EI 6.6.0 accordingly. For example, 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 ESB 5.0.0) to **log4j2** (in WSO2 EI 6.6.0).

1. If you have used a custom log4j component in WSO2 ESB 5.0.0, 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 ESB artifacts

Before you begin, note the following changes that may affect your ESB artifacts.

- If you have created a messages store in your configuration and configured the provider path as `repository/conf/jndi.properties` in your ESB 5.0.0 instance, this needs to be updated with the `jndi.properties` file path in EI 6.6.0 (which is `conf/jndi.properties`).
- If you have used the $ctx function inline (in the **Payload Factory** mediator) to get property values, you need to change this to the full XPath. The $ctx function or the get-property() function can be used inside the arguments (args) tags to get property values. Inline $ctx support (in the payload factory) has been deprecated since WSO2 EI 6.4.0.
- From WSO2 EI 6.4.0 onwards, the **XSLT** mediator writes response messages to the JSON stream. Note that in WSO2 ESB 5.0.0, the XSLT mediator was not doing any changes to the JSON stream after message transformation.
- From WSO2 EI 6.4.0 onwads, there are validations affecting the **Enrich** mediator, which prevents the source and target in the message body.
- From WSO2 EI 6.4.0 onwards, if you have specified an XPath value in your mediation sequence, the response message generated by the ESB will include the element tags of your XPath value. For example, if your XPath value is "//faultdescription", the response message will be `<faultdescription>DESCRIPTION</faultdescription>`. If you want the response message to contain only the DESCRIPTION, you need to specify the XPath value as "//faultdescription/text()".
- From WSO2 EI 6.4.0 onwards, if you are using the MailTo transport to send emails through a mediation sequence, note that the email sender specified in the mediation sequence overrides the email sender configured in the axis2.xml file.

You should manually deploy the Composite Application Archive (CAR) files that you have in WSO2 ESB 5.0.0 to WSO2 EI 6.6.0.

- To migrate deployment artifacts including message flow configurations, copy the required Synapse artifacts from the `<ESB_HOME>/repository/deployment/server/synapse-configs/default` directory to the same directory in WSO2 EI 6.6.0.
- To migrate connector artifacts:

    - Create a folder named `synapse-libs` in the `<EI_HOME>/repository/deployment/server/synapse-configs/default/` directory and copy the JARs from the directory by the same name in WSO2 ESB 5.0.0. Note that this directory will not exist in your WSO2 ESB 5.0.0 distribution if no connectors are used.
    - Copy the JARs from the `<ESB_HOME>/repository/deployment/server/synapse-configs/default/imports` directory to the same directory in WSO2 EI 6.6.0.
- If you have custom artifacts created in the `<ESB_HOME>/repository/deployment/server/` directory, copy them to the same directory in WSO2 EI 6.6.0.

Migrating tenant artifacts (ESB)

If multitenancy is used, copy the tenant artifacts of the ESB from the `<ESB_HOME>/repository/tenants` directory to the same directory in WSO2 EI 6.6.0.

You may not be able to access the tenant domain after migrating to EI 6.6.0. To fix this issue, change the tenant's main sequence located in `<EI_HOME>/repository/tenants/1/synapse-configs/default/sequences/main.xml` as shown below:

```
<sequence xmlns="http://ws.apache.org/ns/synapse" name="main">
   <in>
      <log/>
   </in>
   <out>
      <send/>
   </out>
</sequence>
```

Migrating the ESB Analytics

Follow the instructions given below to upgrade ESB Analytics to the Analytics profile of WSO2 EI 6.6.0. This migration is necessary only if Analytics was used with your ESB 5.0.0 installation.

Migrating Analytics configurations

Do not copy **configuration files** directly between servers. Instead, update the files manually.

To migrate all the required folders, files, libraries, etc.:

1. Copy the database connector JAR files stored in the `<ESB_5.0.0_ANALYTICS_HOME>/repository/components/lib` directory to the `<EI_HOME>/wso2/analytics/repository/components/lib` directory in WSO2 EI 6.6.0.
2. Copy the keystores and truststores from the `<ESB_5.0.0_ANALYTICS_HOME>/repository/resources/security` directory to the `<EI_HOME>/wso2/analytics/repository/resources/security` directory in WSO2 EI 6.6.0.
3. If you have secondary user stores created for WSO2 ESB Analytics 5.0.0, you need to copy the 'userstore' folder in the `<ESB_5.0.0_ANALYTICS_HOME>/repository/deployment/server/` directory to the `<EI_HOME>/wso2/analytics/repository/deployment/server/` directory in WSO2 EI 6.6.0.

To migrate the configurations:

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/analytics/conf/datasources` directory and update the Carbon datasource configuration in the `master-datasources.xml` file with the details of the Carbon database. For instructions, see Changing the Carbon Database and select your database type.
3. Go to the `<EI_HOME>/wso2/analytics/conf/datasources` directory and update the datasource configuration in the `analytics-datasources.xml` file with the details of the Analytics-specific databases.
4. Go to the `<EI_HOME>/wso2/analytics/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. Go to the `<EI_HOME>/wso2/analytics/conf/analytics/` directory and update the `rdbms-config.xml` file according to the configurations in the same file of your previous Analytics installation.
6. Go to the `<EI_HOME>/wso2/analytics/conf/data-bridge` directory in WSO2 EI 6.6.0 and update the configuration files according to the configurations in the previous installation.
7. Go to the `<EI_HOME>/wso2/analytics/conf` directory and update the `event-processor.xml` file according to the configurations in the previous installation.

If you enable the HA mode for ESB Analytics by setting the `<mode name="HA" enable="true">` property in the `event-processor.xml` file, state persistence is enabled by default. If there is no real-time use case that requires any state information after starting the cluster, you should disable event persistence by setting the `persistence` attribute to `false` in the same file as shown below.

```
<persistence enable="false">
    <persistenceIntervalInMinutes>15</persistenceIntervalInMinutes>
```

```
    <persisterSchedulerPoolSize>10</persisterSchedulerPoolSize>
    <persister class="org.wso2.carbon.event.processor.core.internal.persistence.
FileSystemPersistenceStore">
        <property key="persistenceLocation">cep_persistence</property>
    </persister>
</persistence>
```

8. Enable WSO2 EI 6.6.0 to publish ESB statistics to the Analytics profile by following the instructions in Publishing ESB Data to the Analytics Profile.

   Note that in EI 6.6.0, you need to configure the <EI_HOME>/conf/carbon.xml file to enable the ESB server to publish statistics, whereas in WSO2 ESB, you need to configure the `<ESB_HOME>/repository/deployment/server/eventpublishers/MessageFlowConfigurationPublisher.xml` and `<ESB_HOME>/repository/deployment/server/eventpublishers/MessageFlowStatisticsPublisher.xml` file instead.

9. Check for any other configurations that were done for WSO2 ESB 5.0.0 Analytics based on your solution and update the configurations in WSO2 EI 6.6.0 accordingly. For example, check the configurations related to external user stores, caching, mounting, transports etc.

Migrating tenant artifacts (Analytics)

If you are using multitenancy, copy the `<ESB_5.0.0_ANALYTICS_HOME>/repository/tenants` directory to the `<EI_HOME>/wso2/analytics/repository/tenants` directory of WSO2 EI 6.6.0.