# Red Hat JBoss Enterprise Application Platform 8.0

## Using the JBoss Server Migration Tool

For migrating server configurations to Red Hat JBoss Enterprise Application Platform 8.0

# Red Hat JBoss Enterprise Application Platform 8.0 Using the JBoss Server Migration Tool

For migrating server configurations to Red Hat JBoss Enterprise Application Platform 8.0

## Legal Notice

## Abstract

This guide describes how to use the JBoss Server Migration Tool to migrate your application server configuration from a previous release of JBoss EAP to JBoss EAP 8.0.

# Table of Contents

# PROVIDING FEEDBACK ON JBOSS EAP DOCUMENTATION

To report an error or to improve our documentation, log in to your Red Hat Jira account and submit an issue. If you do not have a Red Hat Jira account, then you will be prompted to create an account.

**Procedure**

1. Click the following link to **create a ticket**.

2. Enter a brief description of the issue in the **Summary**.

3. Provide a detailed description of the issue or enhancement in the **Description**. Include a URL to where the issue occurs in the documentation.

4. Clicking **Submit** creates and routes the issue to the appropriate documentation team.

# MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Due to the enormity of this endeavor, these changes will be gradually implemented over upcoming releases. For more details on making our language more inclusive, see our CTO Chris Wright's message .

# CHAPTER 1. INTRODUCTION TO THE JBOSS SERVER MIGRATION TOOL

## 1.1. ABOUT RED HAT JBOSS ENTERPRISE APPLICATION PLATFORM

Red Hat JBoss Enterprise Application Platform (JBoss EAP) is a middleware platform built on open standards that is compatible with the Jakarta Enterprise Edition 10 specification.

The 8.0 release of JBoss EAP is a Jakarta EE Jakarta Enterprise Edition compatible implementation for Web Profile, Core Profile, and Full Platform specifications. The 8.0 release is also a certified implementation of the Jakarta EE Jakarta Enterprise Edition Web Profile, Core Profile, and the Full Platform specifications.

JBoss EAP provides two operating modes for server instances.

**Standalone server**

The standalone server operating mode represents running JBoss EAP as a single server instance.

**Managed domain**

The managed domain operating mode allows for the management of multiple JBoss EAP instances from a single control point.

JBoss EAP includes APIs and development frameworks for quickly developing secure and scalable Jakarta EE applications. Many of the APIs and capabilities that are exposed to applications deployed to JBoss EAP servers are organized into subsystems that are configured in the server configuration files. For example, you configure database access information in the datasources subsystem so that it can be accessed by applications deployed to JBoss EAP standalone servers or managed domains. The introduction of new features and deprecation of other features can require modification of the server configurations from one release of JBoss EAP to another.

For more information about Red Hat JBoss Enterprise Application Platform, see the Product Documentation for JBoss EAP located on the Red Hat Customer Portal.

## 1.2. ABOUT THE USE OF EAP_PREVIOUS_HOME, EAP_NEW_HOME AND MIGRATION_TOOL_HOME IN THIS DOCUMENT

In this document, the variable **EAP_PREVIOUS_HOME** is used to denote the path to the **source** server installation. Replace this variable with the actual path to your **source** server installation. In addition, the variable **EAP_NEW_HOME** is used to denote the path to the **target** server installation. Replace this variable with the actual path to your **target** server installation. Also, the variable **MIGRATION_TOOL_HOME** states the directory where the migration tool is installed.

### 1.2.1. JBoss EAP Installation Path

- If you installed JBoss EAP using the **.zip** file install method, the install directory is the **jboss-eap-8.0** directory where you extracted the **.zip** file.

- If you installed JBoss EAP using the RPM install method, the install directory is **/opt/rh/eap8/root/usr/share/wildfly/**.

- If you used the installer to install JBoss EAP, the default path for **EAP_NEW_HOME** is **${user.home}/EAP-8.0.0**:

- For Red Hat Enterprise Linux, Solaris, and HP-UX: **/home/USER_NAME/EAP-8.0.0/**

- For Microsoft Windows: **C:\Users\USER_NAME\EAP-8.0.0\**

- If you used the JBoss Developer Studio installer to install and configure the JBoss EAP server, the default path for *EAP_NEW_HOME* is **${user.home}/jbdevstudio/runtimes/jboss-eap**:

  - For Red Hat Enterprise Linux: **/home/USER_NAME/jbdevstudio/runtimes/jboss-eap/**

  - For Microsoft Windows: **C:\Users\USER_NAME\jbdevstudio\runtimes\jboss-eap** or **C:\Documents and Settings\USER_NAME\jbdevstudio\runtimes\jboss-eap\**

# CHAPTER 2. BEFORE YOU BEGIN

## 2.1. RUN WITH CLEAN TARGET SERVER INSTALLATION

Because the JBoss Server Migration Tool creates the configuration files based on the configuration of a previous release, it is intended to be run on a clean and unconfigured target server installation. The JBoss Server Migration Tool creates a backup of the target server's configuration files by appending **.beforeMigration** to the file names. It then creates totally new configuration files for the target server using the source server's configuration files, and migrates the configuration to run in the target server configuration.

> **WARNING**
>
> When you run the JBoss Server Migration Tool, all changes on the target server made between installation and running the migration tool are lost.
>
> Also, be aware that if you run the tool against the target server directory more than once, the subsequent runs will overwrite the original target configuration files that were backed up on the first run of the tool. This is because each run of the tool backs up the configuration files by appending **.beforeMigration**, resulting in the loss of any existing backed up configuration files.

## 2.2. CUSTOMIZE THE MIGRATION

The JBoss Server Migration Tool provides the ability to configure logging, reporting, and the execution of migration tasks. By default, when you run the JBoss Server Migration Tool in non-interactive mode, it migrates the entire server configuration. You can configure the JBoss Server Migration Tool to customize logging and reporting output. You can also configure it to skip any part of the configuration that you do not want to migrate.

**Additional resources**

- For instructions on how to configure properties to control the migration process, see Configuring the JBoss Server Migration Tool.

# CHAPTER 3. RUNNING THE JBOSS SERVER MIGRATION TOOL

You can run the JBoss Server Migration Tool in either of the following ways.

- Interactive mode: This mode, which is the default, allows you to choose exactly which configurations you want to migrate.

- Non-interactive mode: This mode allows you to run the tool without prompts.

> **IMPORTANT**
>
> You must stop both the source and the target JBoss EAP servers before you run the JBoss Server Migration Tool.

## 3.1. RUN THE JBOSS SERVER MIGRATION TOOL IN INTERACTIVE MODE

By default, the JBoss Server Migration Tool runs interactively. This mode allows you to choose exactly which server configurations you want to migrate.

> **NOTE**
>
> Interactive mode does not allow you to choose which subsystems to migrate. For information on how to configure the tool at the subsystem or task level, see Configure the migration tasks performed by the JBoss Server Migration Tool.

The following are the basic steps that are performed for a minimal migration. If the server from which you are migrating includes custom configurations, for example deployments, or if it is missing default resources, the tool provides additional prompts.

**Procedure**

1. To run the tool in interactive mode, navigate to the target server installation directory and run the following command, providing the **source** argument as the path to the source server installation.

   ```
   $ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source
   EAP_PREVIOUS_HOME --target EAP_NEW_HOME
   ```

2. You are prompted to determine if you want to migrate the source server's standalone configurations, which are located in the *EAP_PREVIOUS_HOME***/standalone/configuration/** directory, to the target server's standalone configurations, which are located in the *EAP_NEW_HOME***/standalone/configuration/** directory.

   ```
   Migrate the source's standalone server?
   yes/no? yes
   ```

   If you respond with **no**, standalone server migration is skipped and no standalone server configuration files are migrated.

   If you respond with **yes**, you see the following prompt.

> Migrate all configurations?
> yes/no? yes

Respond with **yes** to migrate all of the source server's standalone server configuration files.

Respond with **no** to receive a prompt for each individual **standalone\*.xml** configuration file.

3. Next, you are prompted to determine if you want to migrate the source server's managed domain configurations, which are located in the ***EAP_PREVIOUS_HOME*/domain/configuration/** directory, to the target server's managed domain configurations, which are located in the ***EAP_NEW_HOME*/domain/configuration/** directory.

> Migrate the source's managed domain?
> yes/no? yes

If you respond with **no**, managed domain migration is skipped and no managed domain configuration files are migrated.

If you respond with **yes**, the tool begins migrating the managed domain content of the source server. A ciphered repository is used to store data, such as deployments and deployment overlays, that are referenced by the source server's managed domain and host configurations. Because the source and target servers use a similar content repository, the tool simply copies the data from the source server to the target server and prints the results to the console and the server log.

> INFO  [ServerMigrationTask#397] Migrating domain content found:
> [22/caa450a9ba3b84eaf5a15b6da418b92ce6c98e/content,
> 23/b62a37ba8a4830622bfcdb960280577cc6796e/content]
> INFO  [ServerMigrationTask#398] Resource with path
> /EAP_NEW_HOME/domain/data/content/22/caa450a9ba3b84eaf5a15b6da418b92ce6c98e/content migrated.
> INFO  [ServerMigrationTask#399] Resource with path
> /EAP_NEW_HOME/domain/data/content/23/b62a37ba8a4830622bfcdb960280577cc6796e/content migrated.

4. Next, the migration tool scans the source server for managed domain configuration files, prints the results to the console, and provides the following prompt.

> Migrate all configurations?
> yes/no? yes

Respond with **yes** to migrate all of the source server's managed domain configuration files.

Respond with **no** to receive a prompt for each individual managed domain configuration file.

5. Next, the migration tool scans the source server for host configurations files, prints the results to the console, and provides the following prompt.

> INFO  [ServerMigrationTask#457] Retrieving source's host configurations...
> INFO  [ServerMigrationTask#457] /jboss-eap-8.0/domain/configuration/host-master.xml
> INFO  [ServerMigrationTask#457] /jboss-eap-8.0/domain/configuration/host-slave.xml
> INFO  [ServerMigrationTask#457] /jboss-eap-8.0/domain/configuration/host.xml

> Migrate all configurations?
> yes/no? yes

Respond with **yes** to migrate all of the source server's host configuration files.

Respond with **no** to receive a prompt for each individual host configuration file.

6. Upon completion, you should see the following message in the server console.

> Migration Result: SUCCESS

## 3.2. RUN THE JBOSS SERVER MIGRATION TOOL IN NON-INTERACTIVE MODE

You can run the JBoss Server Migration Tool in non-interactive mode. This mode allows it to run without prompts.

> **NOTE**
>
> The JBoss Server Migration Tool automatically migrates all subsystem configurations for all server configuration files. For information on how to configure the tool at the subsystem or task level, see Configure the migration tasks performed by the JBoss Server Migration Tool.

**Procedure**

1. To run the tool in non-interactive mode, navigate to the target server installation directory and run the following command, providing the **source** argument as the path to the source server installation and setting the **--interactive** or **-i** argument to **false**.

   > $ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source
   > EAP_PREVIOUS_HOME --target EAP_NEW_HOME --interactive false

2. By default, the tool automatically migrates all of the source server's standalone and managed domain configuration files. However, you can configure the tool's properties to skip migration of specific configurations. Upon completion, you should see the following message in the server console.

   > Migration Result: SUCCESS

# CHAPTER 4. REVIEWING THE RESULTS OF JBOSS SERVER MIGRATION TOOL EXECUTION

## 4.1. REVIEW THE MIGRATED CONFIGURATION FILES

When the migration is complete, review the migrated server configuration files in the *EAP_NEW_HOME*/**standalone**/**configuration**/ and *EAP_NEW_HOME*/**domain**/**configuration**/ directories.

- Note that any original *EAP_NEW_HOME* target server configuration file names selected for migration are backed up and are now appended with **.beforeMigration**.

- The *EAP_NEW_HOME* target server configuration file names not appended with **.beforeMigration** are now updated with the content migrated from the *EAP_PREVIOUS_HOME* source server configuration.

- The original configuration files located in the *EAP_PREVIOUS_HOME* source server configuration directories remain untouched.

- The **logging.properties** and **standalone-load-balancer.xml** files in the target configuration directories remain untouched.

If you choose to migrate all of the available configurations, you should see the following configuration files in the target server directories.

**Example: List of configuration files on the target server**

```
$ ls EAP_NEW_HOME/standalone/configuration/
application-roles.properties
application-roles.properties.beforeMigration
application-users.properties
application-users.properties.beforeMigration
logging.properties
mgmt-groups.properties
mgmt-groups.properties.beforeMigration
mgmt-users.properties
mgmt-users.properties.beforeMigration
standalone-full-ha.xml
standalone-full-ha.xml.beforeMigration
standalone-full.xml
standalone-full.xml.beforeMigration
standalone-ha.xml
standalone-ha.xml.beforeMigration
standalone-load-balancer.xml
standalone-osgi.xml
standalone-osgi.xml.beforeMigration
standalone.xml
standalone.xml.beforeMigration

$ ls EAP_NEW_HOME/domain/configuration/
application-roles.properties
application-roles.properties.beforeMigration
application-users.properties
application-users.properties.beforeMigration
```

domain.xml
domain.xml.beforeMigration
host-master.xml
host-master.xml.beforeMigration
host-slave.xml
host-slave.xml.beforeMigration
host.xml
host.xml.beforeMigration
logging.properties
mgmt-groups.properties
mgmt-groups.properties.beforeMigration
mgmt-users.properties
mgmt-users.properties.beforeMigration

## 4.2. TRACKING MIGRATION TASK EXECUTION

The JBoss Server Migration Tool begins each target server migration by executing a root task, which can then execute subtasks. Those subtasks can then also execute additional tasks and subtasks. As it executes, the tool tracks each migration task, along with any subtasks, and saves the results in a tree structure that is later used to build the reports.

Each migration task is given a name, which consists of a task name concatenated with optional attributes using the following syntax.

*TASK_NAME(ATTRIBUTE_1_NAME=ATTRIBUTE_1_VALUE,ATTRIBUTE_2_NAME=ATTRIBUTE_ 2_VALUE, ...)*

The name defines the task subject or type, and the attributes are used to distinguish between subtasks and sibling tasks. For example, all of the following are names to distinguish Jakarta Enterprise Beans subsystem update tasks.

subsystem.ejb3.update
subsystem.ejb3.update.activate-ejb3-remoting-http-connector(resource=/subsystem=ejb3)
subsystem.ejb3.update.setup-default-sfsb-passivation-disabled-cache(resource=/subsystem=ejb3)
subsystem.ejb3.update.add-infinispan-passivation-store-and-distributable-
cache(resource=/subsystem=ejb3)

Since a migration task can be executed multiple times under different parent tasks, each task is stored in the tree using each of its parent task names, starting with root, separated by a **>** character. The task execution tree is used to build the migration reports.

A task execution can result in one of the following statuses.

Table 4.1. Server migration task execution statuses

| Status | Description |
| --- | --- |
| Success | The task executed successfully. |
| Skipped | The task skipped the execution, either because it was not needed or because it was configured to be skipped. |
| Fail | The task execution failed. |

## 4.3. REVIEW THE TASK SUMMARY LOG

The Task Summary is generated and printed to the migration console and to the JBoss Server Migration Tool log file. It provides a high-level overview of the migration results, by component and subtask, as a hierarchical list.

**Additional resources**

- See the appendix of this guide for an example Task Summary report .

- For more information about options to configure the task summary report, see Configuring the Task Summary log.

## 4.4. REVIEW THE JBOSS SERVER MIGRATION TOOL REPORTS

The JBoss Server Migration Tool generates nicely formatted HTML and XML reports in the **MIGRATION_TOOL_HOME/reports/** directory. These reports provide a detailed analysis of the migration process and how the target server was configured during the migration. The default names for these reports are **migration-report.html** and **migration-report.xml**. Each of these names is configurable.

This section provides a brief overview of the content of these reports.

- The JBoss Server Migration Tool HTML report  file.

- The JBoss Server Migration Tool XML report  file.

**Additional resources**

- For information about how to configure the reports, see Configuring reporting for JBoss Server Migration Tool.

### 4.4.1. JBoss Server Migration Tool HTML report

The HTML report consists of three sections.

**Summary**

This section provides the execution start time, information about the source and target servers, and the result of the migration.

**Environment**

This section lists the environment properties that were used for the migration.

**Tasks**

This section, which includes collapsible subsections, provides statistics and a map of the executed migration tasks. Each task is listed by its name and is color-coded according to the status of the completion of the task:

- Green if it was successful.

- Red if it failed.

- Gray if it was skipped.

**Additional resources**

- For an example HTML report , see the appendix of this guide.

- For configuration options for the HTML report, see Configuring the HTML Report.

## 4.4.2. JBoss Server Migration Tool XML report

The XML Report is a low level report that provides all of the migration data gathered by the tool. It is formatted in a way that it can be imported into and manipulated by third-party spreadsheet or other data manipulation tools.

**Additional resources**

- For an example XML report , see the appendix of this guide.

- For configuration options for the XML report, see Configuring the XML Report.

# CHAPTER 5. CONFIGURING THE JBOSS SERVER MIGRATION TOOL

## 5.1. CONFIGURING THE JBOSS SERVER MIGRATION TOOL PROPERTIES

You use properties to configure the JBoss Server Migration Tool logging and reporting output and to determine which components and configurations you want to migrate. You can configure these properties using a combination of the following methods.

- You can configure the properties file defined within the tool.

- You can pass user properties on the command line.

- You can configure system environment variables.

### 5.1.1. Configure the JBoss Server Migration Tool Using the tool properties

You can configure the JBoss Server Migration Tool using the **environment.properties** file located in the **MIGRATION_TOOL_HOME/configuration/** directory. This standard Java properties file provides the default values for all of the valid properties that can be configured when migrating to the target server. To change a default value, remove the **#** comment character that precedes the property and set it to the value you need.

### 5.1.2. Configure the JBoss Server Migration Tool using user properties

If you prefer, you can create a standard Java properties file that defines JBoss Server Migration Tool configuration properties and pass its path on the command line using the **--environment** or **-e** argument. This path can be an absolute path or a path relative to the current directory.

```
$ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source EAP_PREVIOUS_HOME --target EAP_NEW_HOME --environment path/to/my-server-migration.properties
```

Properties defined in files passed on the command line using the **--environment** or **-e** argument override the ones defined in the **MIGRATION_TOOL_HOME/migration/configuration/environment.properties** file.

### 5.1.3. Configure the JBoss Server Migration Tool Using system properties

You can configure the JBoss Server Migration Tool by passing system properties on the command line using the following syntax.

```
$ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source EAP_PREVIOUS_HOME --target EAP_NEW_HOME -Djboss.server.migration.PROPERTY_NAME=PROPERTY_VALUE
```

The system property name should be specified as **jboss.server.migration.** concatenated with the environment property name. The following example demonstrates how to specify the name of the XML report as **migration-report.xml** when starting the JBoss Server Migration Tool.

```
$ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source EAP_PREVIOUS_HOME --target EAP_NEW_HOME -Djboss.server.migration.report.xml.fileName=migration-report.xml
```

Environment properties specified on the command line override both user configuration properties and tool configuration properties.

## 5.2. CONFIGURING LOGGING FOR JBOSS SERVER MIGRATION TOOL

The JBoss Server Migration Tool uses the JBoss Logging framework to log the progress of the migration. Results are written to the console and also to a file named **migration.log**, which is located in the **MIGRATION_TOOL_HOME/migration/logs/** directory. This log file is created if it does not already exist, and its content is overwritten on each subsequent execution of the tool.

The logging configuration is provided by the **MIGRATION_TOOL_HOME/migration/logging.properties** file. You can modify this configuration file or you can specify an alternative logging configuration file by using the **logging.configuration** system property on the command line.

```
$ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source EAP_PREVIOUS_HOME --target EAP_NEW_HOME -Dlogging.configuration=file:EAP_PREVIOUS_HOME/migration/configuration/my-alternate-logging.properties
```

## 5.3. CONFIGURING MODULE MIGRATION

The JBoss Server Migration Tool can migrate any module installed in the source server as long as that module is not already installed on the target server. Module migration can be done explicitly by request, or implicitly because another module or migrated server configuration depends on it.

### 5.3.1. Module environment properties

You can control whether a module should be migrated or not by using the **modules.includes** and **modules.excludes** environment properties. The syntax for a module ID is **name:slot**. The **:slot** is optional and if it is not specified defaults to **main**.

- A module whose ID is referenced by the **modules.excludes** environment property is never migrated.

- A module whose ID is referenced by the **modules.includes** environment property is always migrated, unless it is referenced by the **modules.excludes** environment property.

### 5.3.2. Configuring module properties

The environment properties used to migrate modules can be configured in any of the following ways:

- You can configure the properties in the tool's **MIGRATION_TOOL_HOME/migration/configuration/environment.properties** file.

  ```
  modules.includes=com.example.moduleA,com.example.moduleB
  modules.excludes=com.example.moduleC
  ```

- You can include the above properties in your own custom properties file, and then pass the properties file name on the command line using the **--environment** argument.

  ```
  $ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source EAP_PREVIOUS_HOME --target EAP_NEW_HOME --environment PATH_TO_MY_PROPERTIES_FILE
  ```

- You can pass the information on the command line using a system property. The environment property names must be prefixed with **jboss.server.migration.**, for example:

  $ MIGRATION_TOOL_HOME/bin/jboss-server-migration.sh --source
  *EAP_PREVIOUS_HOME* --target EAP_NEW_HOME -
  Djboss.server.migration.modules.includes="com.example.moduleA" -
  Djboss.server.migration.modules.excludes="com.example.moduleC,com.example.moduleD"

> **WARNING**
>
> The JBoss Server Migration Tool does not verify that the source module is compatible with the target server. An incompatible migrated module can cause the target server to malfunction or not work at all. A module can be incompatible due to a dependency on a module that is installed on both the source and target servers, but includes or exposes different resources on each one.

## 5.4. CONFIGURING REPORTING FOR JBOSS SERVER MIGRATION TOOL

You can configure the following reports for the JBoss Server Migration Tool:

- Task Summary log

- HTML report

- XML report

### 5.4.1. Configuring the Task Summary log

You can customize the generation of the Task Summary using the following environment property.

Table 5.1. Task Summary environment property

| Property name | Type | Property description and default value |
| --- | --- | --- |
| report.summary.maxTaskPathSizeToDisplaySubtasks | Integer | Include migrated subtasks in the summary where the level is less than or equal to the specified integer. Defaults to **5**. |

### 5.4.2. Configuring the HTML report

You can customize the HTML report using the following environment properties.

Table 5.2. HTML report environment properties

| Property name | Type | Property description and default value |
|---|---|---|
| report.html.fileName | String | The name of the HTML report file. If not set, the report is not generated.<br><br>Defaults to **MIGRATION_TOOL_HOME/reports/migration-report.html**. |
| report.html.maxTaskPathSizeToDisplaySubtasks | Integer | Include migrated subtasks in the summary where the level is less than or equal to the specified integer.<br><br>Defaults to **4**. |
| report.html.templateFileName | String | The HTML report template file name.<br><br>Defaults to **migration-report-template.html**. |

### 5.4.3. Configuring the XML report

You can customize the XML report using the following environment properties.

**Table 5.3. XML report environment properties**

| Property name | Type | Property description and default value |
|---|---|---|
| report.xml.fileName | String | The name of the XML report file. If not set, the report is not generated.<br><br>Defaults to **MIGRATION_TOOL_HOME/reports/migration-report.xml**. |

## 5.5. CONFIGURING MIGRATION FOR A STANDALONE SERVER CONFIGURATION

You can configure the JBoss Server Migration Tool to skip the migration of a standalone server entirely, to provide the configuration file names that you want to migrate, or to provide alternate paths for the source or target server's base and configuration directories.

You can customize the migration of the standalone server configuration using the following environment properties.

**Table 5.4. Standalone server migration environment properties**

| Property name | Property description |
|---|---|
| standalone.skip | If set to **true**, the tool skips the entire standalone server migration. |

| Property name | Property description |
| --- | --- |
| server.source.standalone.serverDir | Defines an alternative path for the source server's standalone directory, which defaults to the source server's *EAP_NEW_HOME*/**standalone**/ directory. |
| server.source.standalone.configDir | Defines an alternative path for the source server's standalone configuration directory, which defaults to the source server's *EAP_NEW_HOME*/**standalone**/**configuration**/ directory. |
| server.source.standalone.configFiles | A comma-delimited list of the source server's standalone configurations to be migrated. |
| server.target.standalone.serverDir | Defines an alternative path for the target server's standalone directory, which defaults to the target server's *EAP_NEW_HOME*/**standalone**/ directory. |
| server.target.standalone.configDir | Defines an alternative path for the target server's standalone configuration directory, which defaults to the target server's *EAP_NEW_HOME*/**standalone**/**configuration**/ directory. |

**Additional resources**

- For information about how to configure the JBoss Server Migration Tool using these properties, see Configuring the JBoss Server Migration Tool.

## 5.6. CONFIGURING THE MIGRATION OF A MANAGED DOMAIN CONFIGURATION

You can configure the JBoss Server Migration Tool to skip the migration of a managed domain entirely, to provide the configuration file names that you want to migrate, or to provide alternate paths for the source or target server's base and configuration directories.

You can customize the migration of the managed domain configuration using the following environment properties.

Table 5.5. Managed domain migration environment properties

| Property name | Property description |
| --- | --- |
| domain.skip | If set to **true**, the tool skips the entire managed domain migration. |
| server.source.domain.domainDir | Defines an alternative path for the source server's managed domain directory, which defaults to the source server's *EAP_NEW_HOME*/**domain**/ directory. |

| Property name | Property description |
| --- | --- |
| server.source.domain.configDir | Defines an alternative path for the source server's managed domain configuration directory, which defaults to the source server's **EAP_NEW_HOME**/**domain**/**configuration**/ directory. |
| server.source.domain.domainConfigFiles | A comma-delimited list of the source server's managed domain configurations to be migrated. |
| server.source.domain.hostConfigFiles | A comma-delimited list of the source server's host configuration files that are to be migrated. |
| server.target.domain.domainDir | Defines an alternative path for the target server's managed domain directory, which defaults to the target server's **EAP_NEW_HOME**/**domain**/**configuration**/ directory. |
| server.target.domain.configDir | Defines an alternative path for the target server's managed domain configuration directory, which defaults to the target server's **EAP_NEW_HOME**/**domain**/**configuration**/ directory. |

**Additional resources**

- For information about how to configure the JBoss Server Migration Tool using these properties, see Configuring the JBoss Server Migration Tool.

## 5.7. CONFIGURE THE MIGRATION TASKS PERFORMED BY THE JBOSS SERVER MIGRATION TOOL

By default, the JBoss Server Migration Tool automatically migrates all components and subsystems for each standalone server, managed domain, and host configuration you choose to migrate. You can customize the execution of specific tasks and subtasks performed by the tool using environment properties. For example, you can configure the tool to skip the removal of unsupported subsystems or to skip the migration of deployments. The tasks performed by the tool are dependent upon the type of server configuration and the version of the source server from which you are migrating.

Information about how to configure environment properties to customize the tasks performed by the JBoss Server Migration Tool can be found in the following sections.

# CHAPTER 6. MIGRATING JBOSS EAP 6.4 CONFIGURATIONS TO JBOSS EAP 8.0

The procedure to migrate the JBoss EAP 6.4 configurations to JBoss EAP 8.0 has been updated. You must migrate first your JBoss EAP 6.4 configurations to the latest CP version of JBoss EAP 7.4 and then migrate it further to JBoss EAP 8.0.

- To migrate a JBoss EAP 6.4 configuration to JBoss EAP 7.4, see Use the JBoss Server Migration Tool to Migrate Server Configurations.

- To migrate a JBoss EAP 7.x configuration to JBoss EAP 8.0, see Migrating JBoss EAP 7.x configurations to JBoss EAP 8.0.

# CHAPTER 7. MIGRATING JBOSS EAP 7.X CONFIGURATIONS TO JBOSS EAP 8.0

## 7.1. MIGRATING A JBOSS EAP 7.X STANDALONE SERVER TO JBOSS EAP 8.0

By default, the JBoss Server Migration Tool performs the following tasks when migrating a standalone server configuration from JBoss EAP x to JBoss EAP 8.0.

### 7.1.1. Migrate JBoss Domain Properties

The words **master** and **slave** on Domain related property names were replaced with the words 'primary' and 'secondary', and the migration automatically fixes any usage of the old property names.

The console logs any property renamed by the migration.

```
INFO JBoss domain property jboss.domain.master.address migrated to
jboss.domain.primary.address
INFO JBoss domain property jboss.domain.master.port migrated to jboss.domain.primary.port
INFO JBoss domain property jboss.domain.master.protocol migrated to
jboss.domain.primary.protocol
```

If the properties are successfully renamed, the following message is displayed:

```
INFO JBoss domain properties migrated.
```

### 7.1.2. Remove unsupported subsystems

The JBoss Server Migration Tool removes all unsupported subsystem configurations and extensions from migrated server configurations. The tool logs each subsystem and extension to its log file and to the console as it is removed.

> **NOTE**
>
> Any subsystem that was not supported in JBoss EAP 7.x, but was added by an administrator to that server, is also not supported in JBoss EAP 8.0 and will be removed.

To skip removal of the unsupported subsystems, set the **subsystems.remove-unsupported-subsystems.skip** environment property to **true**.

You can override the default behavior of the JBoss Server Migration Tool and specify which subsystems and extensions should be included or excluded during the migration using the following environment properties.

Table 7.1. Server migration environment properties

| Property name | Property description |
| --- | --- |
| **extensions.excludes** | A list of module names of extensions that should never be migrated, for example, **com.example.extension1**,**com.example.extension3**. |

| Property name | Property description |
|---|---|
| **extensions.includes** | A list of module names of extensions that should always be migrated, for example, **com.example.extension2**,**com.example.extension4**. |
| **subsystems.excludes** | A list of subsystem namespaces, stripped of the version, that should never be migrated, for example, **urn:jboss:domain:logging**, **urn:jboss:domain:ejb3**. |
| **subsystems.includes** | A list of subsystem namespaces, stripped of the version, that should always be migrated, for example, **urn:jboss:domain:security**, **urn:jboss:domain:ee**. |

### 7.1.3. Migrate referenced modules for a standalone server

A configuration that is migrated from a source server to a target server might reference or depend on a module that is not installed on the target server. The JBoss Server Migration Tool detects this and automatically migrates the referenced modules, plus their dependent modules, from the source server to the target server.

A module referenced by a standalone server configuration is migrated using the following process.

- A module referenced by the **datasource** subsystem configuration is migrated as a datasource driver module.

- A module referenced by the **ee** subsystem configuration is migrated as a global module.

- A module referenced by the **naming** subsystem configuration is migrated as an object factory module.

- A module referenced by the **messaging** subsystem configuration is migrated as a Jakarta Messaging bridge module.

- Any extension that is not installed on the target configuration is migrated to the target server configuration.

The console logs a message noting the module ID for any module that is migrated. It is possible to exclude the migration of specific modules by specifying the module ID in the **modules.excludes** environment property.

### 7.1.4. Migrate referenced paths for a standalone server

A configuration that is migrated from a source server to a target server might reference or depend on file paths and directories that must also be migrated to the target server. The JBoss Server Migration Tool does not migrate absolute path references. It only migrates files or directories that are configured as relative to the source configuration. The console logs a message noting each path that is migrated.

## 7.1.5. Migrate legacy Security Realms

JBoss EAP 8 does not support the legacy Security Realms framework. The JBoss Server Migration Tool migrates the configuration to using the default JBoss EAP 8 Elytron replacements.

If the default legacy security realm was not used, you may need to manually configure Elytron.

The console logs configuration resources migrated to the default JBoss EAP 8 Elytron replacements.

> INFO  Legacy security XML configuration retrieved.
> WARN  Migrated Remoting subsystem's http connector resource /subsystem/remoting/http-connector/http-remoting-connector using a legacy security-realm, to Elytron's default application SASL Authentication Factory migration-defaultApplicationSaslAuthenticationFactory. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
> WARN  Migrated Undertow subsystem https-listener resource /subsystem/undertow/server/default-server/https-listener/https using a legacy security-realm, to Elytron's default TLS ServerSSLContext migration-defaultTLSServerSSLContext. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
> WARN  Migrated Undertow subsystem http-invoker resource /subsystem/undertow/server/default-server/host/default-host/setting/http-invoker using a legacy security-realm, to Elytron's default Application HTTP AuthenticationFactory migration-defaultApplicationHttpAuthenticationFactory. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
> INFO  Legacy security realms migrated to Elytron.

## 7.1.6. Migrate legacy Security Domains

JBoss EAP 8 does not support the legacy Security Domains framework. The the JBoss Server Migration Tool migrates the configuration to using the default JBoss EAP 8 Elytron replacements.

If the default legacy security domain was not used, you may need to manually configure Elytron.

The console logs all configuration resources migrated to the default JBoss EAP 8 Elytron replacements.

> WARN  Migrated ejb3 subsystem resource /subsystem/ejb3/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
> WARN  Migrated undertow subsystem resource /subsystem/undertow/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!

## 7.1.7. Migrate keycloak subsystem

The **keycloak** subsystem is not supported in JBoss EAP 8 and is replaced by the **elytron-oidc-client** subsystem.

By default, the JBoss Server Migration Tool automatically migrates any legacy subsystem configuration. To skip this migration task, set the **subsystem.keycloak.migrate.skip** environment property value to **true**.

The legacy subsystem migration is performed without any interaction from the user.

When the legacy **keycloak** subsystem migration is completed, the following message is displayed in the migration console:

> INFO  Subsystem keycloak migrated.

Any issues encountered during the migration are written to the log files and displayed in the migration console.

### 7.1.8. Migrate `picketlink-federation` subsystem

The **picketlink-federation** subsystem is deprecated in JBoss EAP 8 and is replaced by the **keycloak-saml** subsystem.

By default, the JBoss Server Migration Tool automatically migrates any legacy subsystem configuration. To skip this migration task, set the **subsystem.picketlink-federation.migrate.skip** environment property value to **true**.

The legacy subsystem migration is performed without any interaction from the user.

The legacy subsystem migration can fail due to the following reasons:

- The legacy **picketlink-federation** subsystem cannot be migrated to the **keycloak-saml** subsystem due to the target server missing the Keycloak client SAML adapter.

- Non-empty legacy **picketlink-federation** subsystem configurations that must be migrated manually.

When the legacy **picketlink-federation** subsystem migration is completed, the following message is displayed in the migration console:

> INFO  Subsystem picketlink-federation migrated.

Any issues encountered during the migration are written to the log files and displayed in the migration console.

For more information, see the Migration Guide.

### 7.1.9. Update jgroups subsystem configuration

The JBoss Server Migration Tool does not automate the migration of the **jgroups** subsystem configuration. The JBoss Server Migration Tool reverts the configuration to the default JBoss EAP 8 **jgroups** configuration.

If the default JBoss EAP 8 **jgroups** subsystem configuration was not used, you may need to manually configure the **jgroups** subsystem configuration.

The console logs a message when the **jgroups** subsystem configuration is updated:

> WARN  Configuration of JGroups protocols has been changed to match the default protocols of the target server. Please note that further manual configuration may be needed if the legacy configuration being used was not the source server's default configuration!

### 7.1.10. Add the health subsystem for a standalone server

The JBoss EAP 8.0 **health** subsystem provides support for a server's health functionality. The JBoss Server Migration Tool automatically adds the default **health** subsystem configuration to the migrated configuration file.

To skip the addition of the health subsystem configuration, set the **subsystem.health.add.skip** environment property to **true**.

After you add the **health** subsystem to JBoss EAP 8.0, you'll see the following message in your web console:

> INFO  Subsystem health added.

### 7.1.11. Add the metrics subsystem for a standalone server

The JBoss EAP 8.0 **metrics** subsystem provides support for a server's metric functionality. The JBoss Server Migration Tool automatically adds the default **metrics** subsystem configuration to the migrated configuration file.

To skip the addition of the **metrics** subsystem configuration, set the **subsystem.metrics.add.skip** environment property to **true**.

After you add the **metrics** subsystem to JBoss EAP 8.0, you'll see the following message in your web console:

> INFO  Subsystem metrics added.

### 7.1.12. Migrate deployments for a standalone server

The JBoss Server Migration Tool can migrate the following types of standalone server deployment configurations.

- Deployments it references, also known as **persistent deployments**.

- Deployments found in directories monitored by its **deployment scanners**.

- **Deployment overlays** it references.

The migration of a deployment consists of installing related file resources on the target server, and possibly updating the migrated configuration.

The JBoss Server Migration Tool is preconfigured to skip deployments by default when running in non-interactive mode. To enable migration of deployments, set the **deployments.migrate-deployments.skip** environment property to **false**.

### IMPORTANT

Be aware that when you run the JBoss Server Migration Tool in interactive mode and enter invalid input, the resulting behavior depends on the value of the **deployments.migrate-deployments** environment property.

- If **deployments.migrate-deployments.skip** is set to **false** and you enter invalid input, the tool will try to migrate the deployments.

- If **deployments.migrate-deployments.skip** is set to **true** and you enter invalid input, the tool will skip the deployments migration.

### WARNING

The JBoss Server Migration Tool does not determine whether deployed resources are compatible with the target server. This means that applications or resources might not deploy, might not work as expected, or might not work at all. Also be aware that artifacts such as JBoss EAP 7.3 **\*-jms.xml** configuration files are copied without modification and can cause the JBoss EAP server to boot with errors.

Red Hat recommends that you use the Migration Toolkit for Runtimes (MTR) to analyze deployments to determine compatibility among different JBoss EAP servers. For more information, see Product Documentation for Migration Toolkit for Runtimes.

### 7.1.12.1. Migrate persistent deployments for a standalone server

To enable migration of persistent deployments when running in non-interactive mode, set the **deployments.migrate-persistent-deployments.skip** environment property to **false**.

The JBoss Server Migration Tool searches for any persistent deployment references and lists them to the console.

> INFO [ServerMigrationTask#67] Persistent deployments found: [cmtool-helloworld3.war, cmtool-helloworld4.war, cmtool-helloworld2.war, cmtool-helloworld1.war]

The processing workflow then depends on whether you are running the tool in **interactive** mode or in **non-interactive** mode, as described below.

#### Migrating persistent deployments in non-interactive mode

If you run the tool in non-interactive mode, the tool uses the preconfigured properties to determine whether to migrate the persistent deployments. Persistent deployments are migrated only if both the **deployments.migrate-deployments.skip** and **deployments.migrate-persistent-deployments.skip** properties are set to **false**.

#### Migrating persistent deployments in interactive mode

If you run the tool in interactive mode, the JBoss Server Migration Tool prompts you for each deployment using the following workflow.

1. After printing the persistent deployments it finds to the console, you see the following prompt.

   > This tool is not able to assert if persistent deployments found are compatible with the target server, skip persistent deployments migration?
   > yes/no?

   - Respond with **yes** to skip migration of persistent deployments. All deployment references are removed from the migrated configuration and you end this part of the migration process.

   - Respond with **no** to continue with the migration.

2. If you choose to continue, you see the following prompt.

   > Migrate all persistent deployments found?
   > yes/no?

   - Respond with **yes** to automatically migrate all deployments and end this part of the migration process.

   - Respond with **no** to continue with the migration.

3. If you choose to continue, you receive a prompt asking to confirm the migration for each referenced deployment.

   > Migrate persistent deployment 'helloworld01.war'?
   > yes/no?

   - Respond with **yes** to migrate the deployment.

   - Respond with **no** to remove the deployment from the migrated configuration.

     > INFO  [ServerMigrationTask#68] Removed persistent deployment from configuration /deployment=helloworld01.war

### 7.1.12.2. Migrate Deployment scanner deployments for a standalone server

Deployment scanners, which are only used in standalone server configurations, monitor a directory for new files and manage their deployment automatically or through special deployment marker files.

To enable migration of deployments that are located in directories watched by a deployment scanner when running in non-interactive mode, set the **deployments.migrate-deployment-scanner-deployments.skip** environment property to **false**.

When migrating a standalone server configuration, the JBoss Server Migration Tool first searches for any configured deployment scanners. For each scanner found, it searches its monitored directories for deployments marked as deployed and prints the results to the console.

The processing workflow then depends on whether you are running the tool in interactive mode or in non-interactive mode, as described below.

**Migrating Deployment scanner deployments in non-interactive mode**

If you run the tool in non-interactive mode, the tool uses the preconfigured properties to determine whether to migrate the deployment scanner deployments. Deployment scanner deployments are

migrated only if both the **deployments.migrate-deployments.skip** and **deployments.migrate-deployment-scanner-deployments.skip** properties are set to **false**.

**Migrating Deployment scanner deployments in interactive mode**

If you run the tool in interactive mode, the JBoss Server Migration Tool prompts you for each deployment using the following workflow.

1. After printing the deployment scanner deployments it finds to the console, you see the following prompt.

   > This tool is not able to assert if the scanner's deployments found are compatible with the target server, skip scanner's deployments migration?
   > yes/no?

   - Respond with **yes** to skip migration of deployment scanner deployments. All deployment references are removed from the migrated configuration and you end this part of the migration process.

   - Respond with **no** to continue with the migration.

2. If you choose to continue, you see the following prompt.

   > Migrate all scanner's deployments found?
   > yes/no?

   - Respond with **yes** to automatically migrate all deployments and end this part of the migration process.

   - Respond with **no** to continue with the migration.

3. If you choose to continue, you receive a prompt asking to confirm the migration for each referenced deployment.

   > Migrate scanner's deployment 'helloworld02.war'?
   > yes/no?

   - Respond with **yes** to migrate the deployment.

   - Respond with **no** to remove the deployment from the migrated configuration.

     > INFO  [ServerMigrationTask#69] Resource with path
     > EAP_NEW_HOME/standalone/deployments/helloworld02.war migrated.

### 7.1.12.3. Migrate deployment overlays for a standalone server

The migration of deployment overlays is a fully automated process. If you have enabled migration of deployments by setting the **deployments.migrate-deployments.skip** environment property to **false**, the JBoss Server Migration Tool searches for deployment overlays referenced in the standalone server configuration that are linked to migrated deployments. It automatically migrates those that are found, removes those that are not referenced, and logs the results to its log file and to the console.

## 7.2. MIGRATING A JBOSS EAP 7.X MANAGED DOMAIN TO JBOSS EAP 8.0

> **WARNING**
>
> When you use the JBoss Server Migration Tool, migrate your domain controller before you migrate your hosts to ensure your domain controller must use the later version of JBoss EAP when compared to the version used by hosts. For example, a domain controller running on JBoss EAP 7 cannot handle a host running on JBoss EAP 8.0.

By default, the JBoss Server Migration Tool performs the following tasks when migrating a managed domain configuration from JBoss EAP 7 to JBoss EAP 8.0.

## 7.2.1. Migrate JBoss Domain Properties

The words **master** and **slave** on Domain related property names were replaced with the words 'primary' and 'secondary', and the migration automatically fixes any usage of the old property names.

The console logs any property renamed by the migration.

```
INFO JBoss domain property jboss.domain.master.address migrated to
jboss.domain.primary.address
INFO JBoss domain property jboss.domain.master.port migrated to jboss.domain.primary.port
INFO JBoss domain property jboss.domain.master.protocol migrated to
jboss.domain.primary.protocol
```

If the properties are successfully renamed, the following message is displayed:

```
INFO JBoss domain properties migrated.
```

## 7.2.2. Remove unsupported subsystems

The JBoss Server Migration Tool removes all unsupported subsystem configurations and extensions from migrated server configurations. The tool logs each subsystem and extension to its log file and to the console as it is removed.

> **NOTE**
>
> Any subsystem that was not supported in JBoss EAP 7.x, but was added by an administrator to that server, is also not supported in JBoss EAP 8.0 and will be removed.

To skip removal of the unsupported subsystems, set the **subsystems.remove-unsupported-subsystems.skip** environment property to **true**.

You can override the default behavior of the JBoss Server Migration Tool and specify which subsystems and extensions should be included or excluded during the migration using the following environment properties.

Table 7.2. Server migration environment properties

| Property name | Property description |
| --- | --- |
| **extensions.excludes** | A list of module names of extensions that should never be migrated, for example, **com.example.extension1**,**com.example.extension3**. |
| **extensions.includes** | A list of module names of extensions that should always be migrated, for example, **com.example.extension2**,**com.example.extension4**. |
| **subsystems.excludes** | A list of subsystem namespaces, stripped of the version, that should never be migrated, for example, **urn:jboss:domain:logging**, **urn:jboss:domain:ejb3**. |
| **subsystems.includes** | A list of subsystem namespaces, stripped of the version, that should always be migrated, for example, **urn:jboss:domain:security**, **urn:jboss:domain:ee**. |

### 7.2.3. Migrate referenced modules for a managed domain

A configuration that is migrated from a source server to a target server might reference or depend on a module that is not installed on the target server. The JBoss Server Migration Tool detects this and automatically migrates the referenced modules, plus their dependent modules, from the source server to the target server.

A module referenced by a managed domain configuration is migrated using the following process.

- A module referenced by the **datasource** subsystem configuration is migrated as a datasource driver module.

- A module referenced by the **ee** subsystem configuration is migrated as a global module.

- A module referenced by the **naming** subsystem configuration is migrated as an object factory module.

- A module referenced by the **messaging** subsystem configuration is migrated as a Jakarta Messaging bridge module.

- Any extension that is not installed on the target configuration is migrated to the target server configuration.

The console logs a message noting the module ID for any module that is migrated. It is possible to exclude the migration of specific modules by specifying the module ID in the **modules.excludes** environment property.

### 7.2.4. Migrate referenced paths for a managed domain

A configuration that is migrated from a source server to a target server might reference or depend on file paths and directories that must also be migrated to the target server. JBoss Server Migration Tool does not migrate absolute path references. It only migrates files or directories that are configured as relative to the source configuration. The console logs a message noting each path that is migrated.

### 7.2.5. Migrate legacy Security Realms

JBoss EAP 8 does not support the legacy Security Realms framework. The JBoss Server Migration Tool migrates the configuration to using the default JBoss EAP 8 Elytron replacements.

If the default legacy security domain is not used, you may need to manually configure Elytron.

The console logs configuration resources migrated to the default JBoss EAP 8 Elytron replacements.

```
 INFO  Legacy security XML configuration retrieved.
 WARN  Migrated Remoting subsystem's http connector resource /profile/full-
ha/subsystem/remoting/http-connector/http-remoting-connector using a legacy security-realm, to
Elytron's default application SASL Authentication Factory migration-
defaultApplicationSaslAuthenticationFactory. Please note that further manual Elytron configuration
may be needed if the legacy security realm being used was not the source server's default
Application Realm configuration!
 WARN  Migrated Undertow subsystem https-listener resource /profile/full-
ha/subsystem/undertow/server/default-server/https-listener/https using a legacy security-realm, to
Elytron's default TLS ServerSSLContext migration-defaultTLSServerSSLContext. Please note that
further manual Elytron configuration may be needed if the legacy security realm being used was not
the source server's default Application Realm configuration!
 WARN  Migrated Undertow subsystem http-invoker resource /profile/full-
ha/subsystem/undertow/server/default-server/host/default-host/setting/http-invoker using a legacy
security-realm, to Elytron's default Application HTTP AuthenticationFactory migration-
defaultApplicationHttpAuthenticationFactory. Please note that further manual Elytron configuration
may be needed if the legacy security realm being used was not the source server's default
Application Realm configuration!
 INFO  Legacy security realms migrated to Elytron.
 WARN  Migrated Remoting subsystem's http connector resource
/profile/full/subsystem/remoting/http-connector/http-remoting-connector using a legacy security-realm,
to Elytron's default application SASL Authentication Factory migration-
defaultApplicationSaslAuthenticationFactory. Please note that further manual Elytron configuration
may be needed if the legacy security realm being used was not the source server's default
Application Realm configuration!
 WARN  Migrated Undertow subsystem https-listener resource
/profile/full/subsystem/undertow/server/default-server/https-listener/https using a legacy security-
realm, to Elytron's default TLS ServerSSLContext migration-defaultTLSServerSSLContext. Please
note that further manual Elytron configuration may be needed if the legacy security realm being used
was not the source server's default Application Realm configuration!
 WARN  Migrated Undertow subsystem http-invoker resource
/profile/full/subsystem/undertow/server/default-server/host/default-host/setting/http-invoker using a
legacy security-realm, to Elytron's default Application HTTP AuthenticationFactory migration-
defaultApplicationHttpAuthenticationFactory. Please note that further manual Elytron configuration
may be needed if the legacy security realm being used was not the source server's default
Application Realm configuration!
 INFO  Legacy security realms migrated to Elytron.
 WARN  Migrated Remoting subsystem's http connector resource
/profile/ha/subsystem/remoting/http-connector/http-remoting-connector using a legacy security-realm,
to Elytron's default application SASL Authentication Factory migration-
defaultApplicationSaslAuthenticationFactory. Please note that further manual Elytron configuration
may be needed if the legacy security realm being used was not the source server's default
Application Realm configuration!
 WARN  Migrated Undertow subsystem https-listener resource
/profile/ha/subsystem/undertow/server/default-server/https-listener/https using a legacy security-
realm, to Elytron's default TLS ServerSSLContext migration-defaultTLSServerSSLContext. Please
note that further manual Elytron configuration may be needed if the legacy security realm being used
was not the source server's default Application Realm configuration!
 WARN  Migrated Undertow subsystem http-invoker resource
```

/profile/ha/subsystem/undertow/server/default-server/host/default-host/setting/http-invoker using a legacy security-realm, to Elytron's default Application HTTP AuthenticationFactory migration-defaultApplicationHttpAuthenticationFactory. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
 INFO  Legacy security realms migrated to Elytron.
 WARN  Migrated Remoting subsystem's http connector resource /profile/default/subsystem/remoting/http-connector/http-remoting-connector using a legacy security-realm, to Elytron's default application SASL Authentication Factory migration-defaultApplicationSaslAuthenticationFactory. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
 WARN  Migrated Undertow subsystem https-listener resource /profile/default/subsystem/undertow/server/default-server/https-listener/https using a legacy security-realm, to Elytron's default TLS ServerSSLContext migration-defaultTLSServerSSLContext. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
 WARN  Migrated Undertow subsystem http-invoker resource /profile/default/subsystem/undertow/server/default-server/host/default-host/setting/http-invoker using a legacy security-realm, to Elytron's default Application HTTP AuthenticationFactory migration-defaultApplicationHttpAuthenticationFactory. Please note that further manual Elytron configuration may be needed if the legacy security realm being used was not the source server's default Application Realm configuration!
 INFO  Legacy security realms migrated to Elytron.

## 7.2.6. Migrate legacy Security Domains

JBoss EAP 8 does not support the legacy Security Domains framework. The JBoss Server Migration Tool migrates the configuration to using the default JBoss EAP 8 Elytron replacements.

If the default legacy security domain was not used, you may need to manually configure Elytron.

The console logs all configuration resources migrated to the default JBoss EAP 8 Elytron replacements.

 WARN  Migrated ejb3 subsystem resource /profile/default/subsystem/ejb3/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
 WARN  Migrated undertow subsystem resource /profile/default/subsystem/undertow/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
 WARN  Migrated ejb3 subsystem resource /profile/full/subsystem/ejb3/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
 WARN  Migrated undertow subsystem resource /profile/full/subsystem/undertow/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
 WARN  Migrated messaging-activemq subsystem server resource /profile/full/subsystem/messaging-activemq/server/default, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
 WARN  Migrated iiop-openjdk subsystem resource using legacy security domain to Elytron defaults.

> Please note that further manual Elytron configuration should be needed!
>  WARN  Migrated ejb3 subsystem resource /profile/ha/subsystem/ejb3/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
>  WARN  Migrated undertow subsystem resource /profile/ha/subsystem/undertow/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
>  WARN  Migrated ejb3 subsystem resource /profile/full-ha/subsystem/ejb3/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
>  WARN  Migrated undertow subsystem resource /profile/full-ha/subsystem/undertow/application-security-domain/other using legacy security domain other, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
>  WARN  Migrated messaging-activemq subsystem server resource /profile/full-ha/subsystem/messaging-activemq/server/default, to Elytron's default application Security Domain. Please note that further manual Elytron configuration may be needed if the legacy security domain being used was not the source server's default Application Domain configuration!
>  WARN  Migrated iiop-openjdk subsystem resource using legacy security domain to Elytron defaults. Please note that further manual Elytron configuration should be needed!

## 7.2.7. Migrate `keycloak` subsystem

The **keycloak** subsystem is not supported in JBoss EAP 8 and is replaced by the   **elytron-oidc-client** subsystem.

By default, the JBoss Server Migration Tool automatically migrates any legacy subsystem configuration. To skip this migration task, set the **subsystem.keycloak.migrate.skip** environment property value to **true**.

The legacy subsystem migration is performed without any interaction from the user.

When the legacy **keycloak** subsystem migration is completed, the following message is displayed in the migration console:

> INFO  Subsystem keycloak migrated.

Any issues encountered during the migration are written to the log files and displayed in the migration console.

## 7.2.8. Migrate `picketlink-federation` subsystem

The **picketlink-federation** subsystem is deprecated in JBoss EAP 8 and is replaced by the   **keycloak-saml** subsystem.

By default, the JBoss Server Migration Tool automatically migrates any legacy subsystem configuration. To skip this migration task, set the **subsystem.picketlink-federation.migrate.skip** environment property value to **true**.

The legacy subsystem migration is performed without any interaction from the user.

The legacy subsystem migration can fail due to the following reasons:

- The legacy **picketlink-federation** subsystem cannot be migrated to the **keycloak-saml** subsystem due to the target server missing the Keycloak client SAML adapter.

- Non-empty legacy **picketlink-federation** subsystem configurations that must be migrated manually.

When the legacy **picketlink-federation** subsystem migration is completed, the following message is displayed in the migration console:

> INFO  Subsystem picketlink-federation migrated.

Any issues encountered during the migration are written to the log files and displayed in the migration console.

For more information, see the Migration Guide.

### 7.2.9. Update jgroups subsystem configuration

The JBoss Server Migration Tool does not automate the migration of the **jgroups** subsystem configuration. The JBoss Server Migration Tool reverts the configuration to the default JBoss EAP 8 **jgroups** configuration.

If the default JBoss EAP 8 **jgroups** subsystem configuration was not used, you may need to manually configure the **jgroups** subsystem configuration.

The console logs a message when the **jgroups** subsystem configuration is updated:

> WARN  Configuration of JGroups protocols has been changed to match the default protocols of the target server. Please note that further manual configuration may be needed if the legacy configuration being used was not the source server's default configuration!

### 7.2.10. Add host excludes for managed domain migration

The JBoss EAP 8.0 domain controller can potentially include functionality that is not supported by hosts running on older versions of the server. The **host-exclude** configuration specifies the resources that should be hidden from those older versions.

When migrating a domain controller configuration, the JBoss Server Migration Tool adds to or replaces the source server's **host-exclude** configuration with the configuration of the target JBoss EAP 8.0 server.

The JBoss Server Migration Tool automatically updates the **host-exclude** configuration and logs the results to its log file and to the console.

> INFO  Host-excludes configuration added.

### 7.2.11. Migrate deployments for a managed domain

The JBoss Server Migration Tool can migrate the following types of managed domain deployment configurations.

- Deployments it references, also known as **persistent deployments**.

- **Deployment overlays** it references.

The migration of a deployment consists of installing related file resources on the target server, and possibly updating the migrated configuration.

The JBoss Server Migration Tool is preconfigured to skip deployments by default when running in non-interactive mode. To enable migration of deployments, set the **deployments.migrate-deployments.skip** environment property to **false**.

> ### IMPORTANT
>
> Be aware that when you run the JBoss Server Migration Tool in interactive mode and enter invalid input, the resulting behavior depends on the value of the **deployments.migrate-deployments** environment property.
>
> - If **deployments.migrate-deployments.skip** is set to **false** and you enter invalid input, the tool will try to migrate the deployments.
>
> - If **deployments.migrate-deployments.skip** is set to **true** and you enter invalid input, the tool will skip the deployments migration.

> ### WARNING
>
> The JBoss Server Migration Tool does not determine whether deployed resources are compatible with the target server. This means that applications or resources might not deploy, might not work as expected, or might not work at all. Also be aware that artifacts such as JBoss EAP 7.3 **\*-jms.xml** configuration files are copied without modification and can cause the JBoss EAP server to boot with errors.
>
> Red Hat recommends that you use the Migration Toolkit for Runtimes (MTR) to analyze deployments to determine compatibility among different JBoss EAP servers. For more information, see Product Documentation for Migration Toolkit for Runtimes.

### 7.2.11.1. Migrate persistent deployments for a managed domain

To enable migration of persistent deployments when running in non-interactive mode, set the **deployments.migrate-persistent-deployments.skip** environment property to **false**.

The JBoss Server Migration Tool searches for any persistent deployment references and lists them to the console.

> INFO [ServerMigrationTask#67] Persistent deployments found: [cmtool-helloworld3.war, cmtool-helloworld4.war, cmtool-helloworld2.war, cmtool-helloworld1.war]

The processing workflow then depends on whether you are running the tool in **interactive** mode or in **non-interactive** mode, as described below.

#### Migrating persistent deployments in non-interactive mode

If you run the tool in non-interactive mode, the tool uses the preconfigured properties to determine whether to migrate the persistent deployments. Persistent deployments are migrated only if both the **deployments.migrate-deployments.skip** and **deployments.migrate-persistent-deployments.skip**

properties are set to **false**.

**Migrating persistent deployments in interactive mode**

If you run the tool in interactive mode, the JBoss Server Migration Tool prompts you for each deployment using the following workflow.

1. After printing the persistent deployments it finds to the console, you see the following prompt.

   > This tool is not able to assert if persistent deployments found are compatible with the target server, skip persistent deployments migration?
   > yes/no?

   - Respond with **yes** to skip migration of persistent deployments. All deployment references are removed from the migrated configuration and you end this part of the migration process.

   - Respond with **no** to continue with the migration.

2. If you choose to continue, you see the following prompt.

   > Migrate all persistent deployments found?
   > yes/no?

   - Respond with **yes** to automatically migrate all deployments and end this part of the migration process.

   - Respond with **no** to continue with the migration.

3. If you choose to continue, you receive a prompt asking to confirm the migration for each referenced deployment.

   > Migrate persistent deployment 'helloworld01.war'?
   > yes/no?

   - Respond with **yes** to migrate the deployment.

   - Respond with **no** to remove the deployment from the migrated configuration.

     > INFO  [ServerMigrationTask#68] Removed persistent deployment from configuration
     > /deployment=helloworld01.war

### 7.2.11.2. Migrate deployment overlays for a managed domain

The migration of deployment overlays is a fully automated process. If you have enabled migration of deployments by setting the **deployments.migrate-deployments.skip** environment property to **false**, the JBoss Server Migration Tool searches for deployment overlays referenced in the standalone server configuration that are linked to migrated deployments. It automatically migrates those that are found, removes those that are not referenced, and logs the results to its log file and to the console.

## 7.3. MIGRATING A JBOSS EAP 7.X HOST CONFIGURATION TO JBOSS EAP 8.0

By default, the JBoss Server Migration Tool performs the following tasks when migrating a host server configuration from JBoss EAP 7.x to JBoss EAP 8.0.

## 7.3.1. Migrate JBoss Domain Properties

The words **master** and **slave** on Domain related property names were replaced with the words 'primary' and 'secondary', and the migration automatically fixes any usage of the old property names.

The console logs any property renamed by the migration.

> INFO JBoss domain property jboss.domain.master.address migrated to jboss.domain.primary.address
> INFO JBoss domain property jboss.domain.master.port migrated to jboss.domain.primary.port
> INFO JBoss domain property jboss.domain.master.protocol migrated to jboss.domain.primary.protocol

If the properties are successfully renamed, the following message is displayed:

> INFO JBoss domain properties migrated.

## 7.3.2. Migrate referenced modules for a host configuration

A configuration that is migrated from a source server to a target server might reference or depend on a module that is not installed on the target server. The JBoss Server Migration Tool detects this and automatically migrates the referenced modules, plus their dependent modules, from the source server to the target server.

A module referenced by a host server configuration is migrated using the following process.

- A module referenced by a security realm configuration is migrated as a plug-in module.

The console logs a message noting the module ID for any module that is migrated. It is possible to exclude the migration of specific modules by specifying the module ID in the **modules.excludes** environment property.

## 7.3.3. Migrate referenced paths for a host configuration

A configuration that is migrated from a source server to a target server might reference or depend on file paths and directories that must also be migrated to the target server. The JBoss Server Migration Tool does not migrate absolute path references. It only migrates files or directories that are configured as relative to the source configuration. The console logs a message noting each path that is migrated.

# APPENDIX A. REFERENCE MATERIAL

## A.1. EXAMPLE TASK SUMMARY REPORT

The following is an example of the Task Summary report.

```
-------------
Task Summary
-------------

server ................................................................................. SUCCESS
  standalone ........................................................................... SUCCESS
    standalone-configurations ....................................................... SUCCESS
    standalone-configuration(source=/home/username/jboss-eap-
8.0/standalone/configuration/standalone-full-ha.xml) .. SUCCESS
    standalone-configuration(source=/home/username/jboss-eap-
8.0/standalone/configuration/standalone-full.xml) ..... SUCCESS
    standalone-configuration(source=/home/username/jboss-eap-
8.0/standalone/configuration/standalone-ha.xml) ....... SUCCESS
    standalone-configuration(source=/home/username/jboss-eap-
8.0/standalone/configuration/standalone-osgi.xml) ..... SUCCESS
    standalone-configuration(source=/home/username/jboss-eap-
8.0/standalone/configuration/standalone.xml) .......... SUCCESS
  domain ................................................................................. SUCCESS
    domain-configurations ........................................................... SUCCESS
    domain-configuration(source=/home/username/jboss-eap-8.0/domain/configuration/domain.xml)
..................... SUCCESS
    host-configurations ............................................................... SUCCESS
    host-configuration(source=/home/username/jboss-eap-8.0/domain/configuration/host-master.xml)
................... SUCCESS
    host-configuration(source=/home/username/jboss-eap-8.0/domain/configuration/host-slave.xml)
................... SUCCESS
    host-configuration(source=/home/username/jboss-eap-8.0/domain/configuration/host.xml)
......................... SUCCESS

-------------------------
Migration Result: SUCCESS
-------------------------
```

## A.2. EXAMPLE HTML REPORT

The following is an example of the JBoss Server Migration HTML report.

## Figure A.1. Example: Overview of sections

### JBoss Server Migration Report

#### Summary

| | |
|---|---|
| Start Time: | Mon, 30 Oct 2017 16:13:30 UTC |
| Source Server: | EAP 6.4.0.GA |
| Source Path: | /home/username/jboss-eap-6.4 |
| Target Server: | JBoss EAP 7.1.0.GA |
| Target Path: | /home/username/jboss-eap-7.1 |
| Result: | SUCCESS |

#### Environment

| | |
|---|---|
| baseDir: | /home/username/jboss-eap-7.1/migration |
| deployments.migrate-deployment-scanner-deployments.processedDeploymentScannerDirs: | /home/username/jboss-eap-6.4/standalone/deployments |
| report.html.fileName: | migration-report.html |
| report.html.templateFileName: | migration-report-template.html |
| report.summary.maxTaskPathSizeToDisplaySubtasks: | 3 |
| report.xml.fileName: | migration-report.xml |
| subsystem.ee.update.setup-javaee7-default-bindings.defaultDataSourceName: | ExampleDS |
| subsystem.ee.update.setup-javaee7-default-bindings.defaultJmsConnectionFactoryName: | hornetq-ra |
| subsystem.logging.update.remove-console-handler.skip: | true |

#### Tasks

| | |
|---|---|
| Executed: | 664 |
| Successful: | 564 |
| Skipped: | 100 |
| Failed: | 0 |

server -

| | |
|---|---|
| Task Number: | 1 |
| Task Name: | server |
| Task Path: | server |
| Logger Name: | org.jboss.migration.core.task.ServerMigrationTask#1 |
| Result Status: | SUCCESS |
| Subtasks: | modules.migrate-modules-requested-by-user<br>standalone<br>domain |

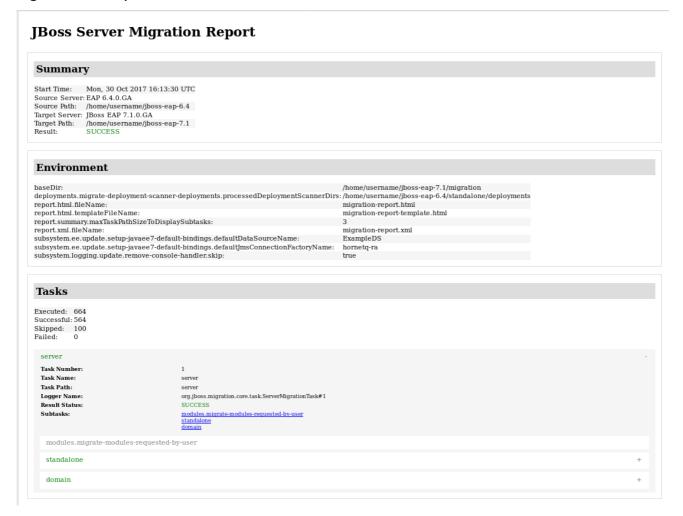modules.migrate-modules-requested-by-user

standalone +

domain +

Figure A.2. Example: Tasks detail page



## A.3. EXAMPLE XML REPORT

The following is an example of the JBoss Server Migration XML report.

```xml
<?xml version="1.0" ?>

<server-migration-report xmlns="urn:jboss:server-migration:1.0" start-time="Mon, 30 Oct 2023
16:13:30 UTC">
  <servers>
    <source name="EAP" version="7.4.0.GA" base-dir="/home/username/tools/jboss-eap-7.4"/>
    <target name="JBoss EAP" version="8.0.0.GA" base-dir="/home/username/tools/jboss-eap-8.0"/>
  </servers>
  <environment>
    <property name="baseDir" value="/home/username/tools/jboss-eap-8.0/migration"/>
    <property name="deployments.migrate-deployment-scanner-
deployments.processedDeploymentScannerDirs" value="/home/username/tools/jboss-eap-
7.4/standalone/deployments"/>
    <property name="report.html.fileName" value="migration-report.html"/>
    <property name="report.html.maxTaskPathSizeToDisplaySubtasks" value="4"/>
    <property name="report.html.templateFileName" value="migration-report-template.html"/>
```

```xml
    <property name="report.summary.maxTaskPathSizeToDisplaySubtasks" value="3"/>
    <property name="report.xml.fileName" value="migration-report.xml"/>
    <property name="subsystem.ee.update.setup-javaee7-default-bindings.defaultDataSourceName" value="ExampleDS"/>
    <property name="subsystem.ee.update.setup-javaee7-default-bindings.defaultJmsConnectionFactoryName" value="hornetq-ra"/>
    <property name="subsystem.logging.update.remove-console-handler.skip" value="true"/>
  </environment>
  <task number="1" name="server">
   <logger logger="org.jboss.migration.core.task.ServerMigrationTask#1"/>
   <result status="SUCCESS"/>
   <subtasks>
    <task number="2" name="modules.migrate-modules-requested-by-user">
     <logger logger="org.jboss.migration.core.task.ServerMigrationTask#2"/>
     <result status="SKIPPED"/>
    </task>
    <task number="3" name="standalone">
     <logger logger="org.jboss.migration.core.task.ServerMigrationTask#3"/>
     <result status="SUCCESS"/>
     <subtasks>
      <task number="4" name="contents.standalone.migrate-content-dir">
       <logger logger="org.jboss.migration.core.task.ServerMigrationTask#4"/>
       <result status="SKIPPED"/>
      </task>
      <task number="5" name="standalone-configurations">
       <logger logger="org.jboss.migration.core.task.ServerMigrationTask#5"/>
       <result status="SUCCESS"/>
       <subtasks>
        <task number="6" name="standalone-configuration(source=/home/username/tools/jboss-eap-6.4/standalone/configuration/standalone-full-ha.xml)">
         <logger logger="org.jboss.migration.core.task.ServerMigrationTask#6"/>
         <result status="SUCCESS"/>
         <subtasks>
          <task number="7" name="subsystems.remove-unsupported-subsystems">
           <logger logger="org.jboss.migration.core.task.ServerMigrationTask#7"/>
           <result status="SUCCESS"/>
           <subtasks>
            <task number="8" name="subsystems.remove-unsupported-subsystems.remove-unsupported-extension(module=org.jboss.as.cmp)">
             <logger logger="org.jboss.migration.core.task.ServerMigrationTask#8"/>
             <result status="SUCCESS"/>
            </task>
           </subtasks>
          </task>
          ...
          <task number="644" name="hosts">
           <logger logger="org.jboss.migration.core.task.ServerMigrationTask#644"/>
           <result status="SUCCESS"/>
           ...
           <subtasks>
            ...
            <task number="645" name="host(name=master)">
             <logger logger="org.jboss.migration.core.task.ServerMigrationTask#645"/>
             <result status="SUCCESS"/>
             <subtasks>
              ...
```

```xml
                    <task number="661" name="security-realms.migrate-properties">
                      <logger logger="org.jboss.migration.core.task.ServerMigrationTask#661"/>
                      <result status="SUCCESS"/>
                      <subtasks>
                        <task number="662" name="security-realm.ManagementRealm.migrate-
properties">

                          <logger logger="org.jboss.migration.core.task.ServerMigrationTask#662"/>
                          <result status="SUCCESS"/>
                        </task>
                        <task number="663" name="security-realm.ApplicationRealm.migrate-properties">
                          <logger logger="org.jboss.migration.core.task.ServerMigrationTask#663"/>
                          <result status="SUCCESS"/>
                        </task>
                      </subtasks>
                    </task>
                    <task number="664" name="security-realm.ApplicationRealm.add-ssl-server-
identity">

                      <logger logger="org.jboss.migration.core.task.ServerMigrationTask#664"/>
                      <result status="SUCCESS"/>
                    </task>
                  </subtasks>
                </task>
              </subtasks>
            </task>
          </subtasks>
        </task>
      </subtasks>
    </task>
  </subtasks>
</task>
</server-migration-report>
```

*Revised on 2024-02-21 14:04:13 UTC*