

# OSGi Guide

JEUS 9

**TMAXSOFT**

# Copyright

Copyright 2025. TmaxSoft Co., Ltd. All Rights Reserved.

## Company Information

TmaxSoft Co., Ltd.

TmaxSoft Tower 10F, 45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, South Korea

Website: <https://www.tmaxsoft.com/en/>

## Restricted Rights Legend

All TmaxSoft Software (JEUS®) and documents are protected by copyright laws and international convention. TmaxSoft software and documents are made available under the terms of the TmaxSoft License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxSoft Co., Ltd. Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxSoft trademarks, logos, or any other brand features.

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

## Trademarks

JEUS® is registered trademark of TmaxSoft Co., Ltd.

Java, Solaris are registered trademarks of Oracle Corporation and its subsidiaries and affiliates.

Microsoft, Windows, Windows NT are registered trademarks or trademarks of Microsoft Corporation.

HP-UX is a registered trademark of Hewlett Packard Enterprise Company.

AIX is a registered trademark of International Business Machines Corporation.

UNIX is a registered trademark of X/Open Company, Ltd.

Linux is a registered trademark of Linus Torvalds.

Noto is a trademark of Google Inc. Noto fonts are open source. All Noto fonts are published under the SIL Open Font License, Version 1.1. (<https://www.google.com/get/noto/>)

Other products and company names are trademarks or registered trademarks of their respective

owners.

The names of companies, systems, and products mentioned in this manual may not necessarily be indicated with a trademark symbol (™, ®).

## Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses: APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

Detailed Information related to the license can be found in the following directory:  
\${INSTALL\_PATH}/license/oss\_licenses

## Document History

Product Version	Guide Version	Date	Remarks
JEUS 9	3.1.2	2025-03-24	-
JEUS 9	3.1.1	2024-12-24	-

# Contents

1. Introduction .....	1
1.1. Overview .....	1
1.2. OSGi Framework .....	1
1.3. Overview of JEUS OSGi .....	1
2. OSGi Framework Configuration and Management .....	3
2.1. Overview .....	3
2.2. OSGi Framework Configuration Items .....	3
2.3. Adding OSGi Framework Configuration .....	5
2.4. Modifying OSGi Framework Configuration .....	6
2.5. Deleting OSGi Framework Configuration .....	8
2.6. Access to OSGi Framework through JNDI .....	8
3. Bundle Management and Monitoring .....	10
3.1. Monitoring Bundles .....	10
3.2. Installing Bundles .....	10
3.3. Starting Bundles .....	11
3.4. Stopping Bundles .....	11
3.5. Updating Bundles .....	11
3.6. Deleting Bundles .....	11
4. Web Application Deployment .....	13

# 1. Introduction

This chapter briefly describes the OSGi technology.

## 1.1. Overview

OSGi, a technology defined by the OSGi Alliance, consists of a set of specifications for a dynamic component system designed for Java. An application compatible with the OSGi specifications is composed of small, reusable components that are dynamically composed. These components, referred to as bundles, communicate with each other through Java objects known as OSGi services, which are managed by the OSGi service registry.



For more information about the OSGi technology, refer to <http://www.osgi.org>. This website provides useful information about the specifications that constitute the OSGi technology, as well as Javadoc for API.

## 1.2. OSGi Framework

The OSGi framework is a fundamental element in the OSGi specifications. It provides scalability with bundles and enables stable management and deployment of downloadable application components.

Main functions provided by the OSGi framework are as follows.

- The framework manages the lifecycle of bundles. The bundle, a unit of managing the OSGi framework, is a JAR file that records property values defined by the OSGi specifications in the META-INF/Manifest.MF file. The OSGi framework itself is a bundle and the OSGi framework bundle is called System Bundle.
- The framework manages dependency between bundles, and wire them if necessary. Each bundle describes required and available packages and versions in Manifest.MF according to the OSGi specifications. The OSGi framework use the information to identify dependencies between bundles and wire them.
- The framework manages the services provided by bundles. In the context of OSGi, a service refers to a general Java object that offers specific functionalities. Each bundle registers its services in the service registry, which is managed by the OSGi framework, or utilizes the registered services by using the methods described in the OSGi specifications.

## 1.3. Overview of JEUS OSGi

The JEUS OSGi module provides an environment where JEUS, a Jakarta EE-based web application server, can create and use OSGi framework instances. The following describes the functions provided by the JEUS OSGi module.

- The OSGi framework instances can be created and started according to the configuration when starting the server.
- Application can approach the OSGI framework bundles through JNDI.
- The status of the OSGi framework can be checked, and OSGi bundles can be installed, started, stopped, resumed, or removed.
- Web applications are enabled to access services managed by the OSGi framework.

## 2. OSGi Framework Configuration and Management

This chapter describes how to configure and use the OSGi framework with the JEUS OSGi module.

### 2.1. Overview

To execute an application written according to the OSGi specifications and composed of bundles, the OSGi framework is necessary. The OSGi framework is a fundamental component of OSGi technology, responsible for deploying and managing bundles, handling dependencies between bundles, and facilitating the registration and management of services through the OSGi service registry.

The JEUS OSGi module facilitates the setup of an OSGi environment for the user, by automatically creating instances of the OSGi framework during the initialization process when the JEUS server starts, and by installing the pre-configured bundles. These tasks are performed through server configuration. The user can execute the related tasks by using jeusadmin, a console-based management tool.



JEUS uses Apache Felix to implement the OSGi framework. For more information about Apache Felix, refer to [Apache Felix Official Documentation](#).

### 2.2. OSGi Framework Configuration Items

OSGi framework configuration items are as follows.



The configuration items for OSGi on JEUS do not support dynamic configuration. To change the configuration of a running server, a server restart is necessary.

- Name

Specifies the unique name of the OSGi framework.

- JNDI Export Name

Specifies the name of JNDI to which OSGi framework objects are registered. If not specified, the framework name specified in 'Name' is used as the default JNDI name. When a user application directly accesses the OSGi framework, it can use the framework objects registered in JNDI by performing a JNDI lookup. The OSGi framework objects registered in JNDI are only valid within the server environment, and remote JNDI lookup is not supported.

- Target Servers

Specifies the server to use the OSGi framework.

- Framework Configuration Properties

Specifies the configuration values to be applied when creating and initializing the OSGi framework. Each property is set as a key-value pair. You can use both pre-defined properties specified in the OSGi specifications and pre-defined values in Apache Felix framework. Additionally you can define application-specific configuration properties. For more information about properties defined by the OSGi specifications, refer to [OSGi Core Specification](#). To learn about properties defined by Apache Felix used in JEUS, refer to [Apache Felix Framework Configuration Properties](#)

If the user does not specify the properties, default values are configured by the JEUS OSGi as follows.

Property	Value
felix.startlevel.bundle	30
org.osgi.framework.bundle.parent	framework
org.osgi.framework.startlevel.beginning	100
org.osgi.framework.storage	SERVER_HOME/.workspace/osgi/Framework_NAME

- Configuration items for Initial Bundles Directory

The JEUS OSGi module provides configuration items to specify the initial bundles directory and control its operation. If the initial bundles directory is specified for use, the JEUS OSGi module completes the initialization process of the framework and checks the designated location. It treats the JAR files in the directory as bundles and installs them in the framework.



JEUS provides the functionality of installing the bundles in the specified directory when the framework starts. However, it does not offer directory management or synchronization functions.

The configurable items for the initial bundles directory are as follows.

Item	Description
Install Initial Bundles	Decides whether to install bundles in the initial bundles directory when the OSGi framework starts.
Initial Bundles Directory Location	Specifies the location (file path) of the initial bundles directory. If not specified, the default path used is <code>JEUS_HOME/lib/osgi/Framework_NAME</code> directory.
Default Start Level	Specifies the default start level for the initial bundles directory.
Fail On Error	Decides whether to regard the failed installation of bundles as an error.

- Bundle Installation Descriptors

Specifies the location of bundle installation descriptor XML files. If the location is specified as a relative path, the file location is decided based on the initial bundles directory.

The bundle installation descriptor used in the JEUS OSGi module is an XML file that contains the information about the bundles to be installed. If the user creates the bundle installation descriptor that includes information of the bundles, and specifies the XML file location in the OSGi framework configuration, the JEUS OSGi module completes the initialization process of the OSGi framework and installation of bundles in the initial bundle directory, and then reads the XML file to install bundles based on the information in the file.

The following is an example of the bundle installation descriptor.

#### Bundle Installation Descriptor Example

```
<?xml version="1.0" encoding="UTF-8"?>
<bundles xmlns="http://www.tmaxsoft.com/xml/ns/jeus/osgi/bundles">
  <!-- Specifies the default start level to be applied when installing bundles specified in the
XML file. -->
  <default-start-level>15</default-start-level>
  <!-- Decides whether to regard the failed bundle installation as an error. -->
  <fail-on-error>true</fail-on-error>
  <bundle>
    <!-- Specifies the start level value for bundles. This overrides the value specified in
default-start-level. -->
    <start-level>5</start-level>
    <!-- Decides whether to install bundles -->
    <install>true</install>
    <!-- Decides whether to start the bundles after installation -->
    <start>true</start>
    <!-- Specifies the URL of bundles to be installed -->
    <location>http://central.maven.org/maven2/org/apache/logging/log4j/log4j-api/2.8.2/log4j-
api-2.8.2.jar</location>
  </bundle>
</bundles>
```

## 2.3. Adding OSGi Framework Configuration

This section describes how to add OSGi framework configuration.

### Using the console tool

The following is an example of adding OSGi framework configuration by using the console tool.

```
[MASTER]domain1.adminServer>list-osgi-frameworks
No OSGi Frameworks exists.
[MASTER]domain1.adminServer>add-osgi-framework -name framework-0 -servers server1
Successfully performed the ADD operation for OSGi Framework framework-0, but all changes were non-
dynamic. They will be applied after restarting.
Check the results using "list-osgi-frameworks".
[MASTER]domain1.adminServer>list-osgi-frameworks
List of OSGi Frameworks
=====
```

```

+-----+-----+-----+
| Framework Name | JNDI Export Name | Target Servers |
+-----+-----+-----+
| framework-0   |                   | server1        |
+-----+-----+-----+
=====
[MASTER]domain1.adminServer>

```



For detailed explanations of the commands used in the examples, refer to "OSGi Commands" in JEUS Reference Guide.

After the configuration is completed successfully, and the server restarts, the following message is displayed that OSGi framework is started successfully.

```

...
[2017.10.25 16:06:24][1] [server1-1] [OSGi-0003] OSGi Framework [framework-0] is started
successfully.
...

```

## 2.4. Modifying OSGi Framework Configuration

This section describes how to modify OSGi framework configuration.

### Using the console tool

The following is an example of changing OSGi framework configuration by using the console tool.

```

[MASTER]domain1.adminServer>list-osgi-frameworks -framework framework-0
OSGi Framework [framework-0]
=====
+-----+-----+-----+
| Attribute Name | Value |
+-----+-----+-----+
| Framework Name | framework-0 |
| JNDI Export Name | Not Specified |
| Target Servers | server1 |
| Install Initial Bundles | true |
| Initial Bundles Directory Location | Not Specified |
| Default Start Level | 30 |
| Fail On Error | true |
+-----+-----+-----+
=====

Framework Configuration Properties
=====
+-----+-----+
| Key | Value |
+-----+-----+
(No data available)

```

```

=====
Bundle Installation Descriptors
=====
+-----+
|                               Location                               |
+-----+
(No data available)
=====

[MASTER]domain1.adminServer>modify-osgi-framework -name framework-0 -dsl 45
Successfully performed the MODIFY operation for OSGi Framework framework-0, but some changes were
non-dynamic. They will be applied after restarting.
Check the results using "list-osgi-frameworks".
[MASTER]domain1.adminServer>list-osgi-frameworks -framework framework-0
OSGi Framework [framework-0]
=====
+-----+-----+
| Attribute Name | Value |
+-----+-----+
| Framework Name | framework-0 |
| JNDI Export Name | Not Specified |
| Target Servers | server1 |
| Install Initial Bundles | true |
| Initial Bundles Directory Location | Not Specified |
| Default Start Level | 45 |
| Fail On Error | true |
+-----+-----+
=====

Framework Configuration Properties
=====
+-----+-----+
| Key | Value |
+-----+-----+
(No data available)
=====

Bundle Installation Descriptors
=====
+-----+
|                               Location                               |
+-----+
(No data available)
=====
[MASTER]domain1.adminServer>

```



1. For detailed explanations of the commands used in the examples, refer to "OSGi Commands" in JEUS Reference Guide.
2. The configuration items for OSGi on JEUS do not support dynamic configuration. To change the configuration of a running server, a server restart is necessary.

## 2.5. Deleting OSGi Framework Configuration

This section describes how to delete OSGi framework configuration.

### Using the console tool

The following is an example of deleting OSGi framework configuration by using the console tool.

```
[MASTER]domain1.adminServer>list-osgi-frameworks
List of OSGi Frameworks
=====
+-----+-----+-----+
| Framework Name | JNDI Export Name | Target Servers |
+-----+-----+-----+
| framework-0   |                   | server1        |
+-----+-----+-----+
=====
[MASTER]domain1.adminServer>remove-osgi-framework -name framework-0
Successfully performed the REMOVE operation for OSGi Framework framework-0.
Check the results using "list-osgi-frameworks".
[MASTER]domain1.adminServer>list-osgi-frameworks
No OSGi Frameworks exists.
[MASTER]domain1.adminServer>
```



1. For detailed explanations of the commands used in the examples, refer to "OSGi Commands" in JEUS Reference Guide.
2. The configuration items for OSGi on JEUS do not support dynamic configuration. To change the configuration of a running server, a server restart is necessary.

## 2.6. Access to OSGi Framework through JNDI

Once the OSGi framework is created, the JEUS OSGi module registers the system bundle context (the bundle context of the OSGi framework) in JEUS JNDI. The type of the registered object is 'org.osgi.framework.launch.Framework', and the JNDI name is the name specified in the configuration. If no name is specified, the OSGi framework name is used as the JNDI name. Applications can access the OSGi framework object through JNDI and use it if needed.

The following is an example of how to get the service object by accessing the OSGi framework object registered in JNDI with the name 'framework-0'.

Example of accessing the service object through the OSGi framework registered in JNDI

```
...
    InitialContext ctx = new InitialContext();
    BundleContext bundleContext = ((Framework) initialContext.lookup("framework-
0")).getBundleContext();
```

```
ServiceReference sref = bctx.getServiceReference(ServiceA.class);  
...  
ServiceA service = (ServiceA)bctx.getService(sref);  
...
```

# 3. Bundle Management and Monitoring

This chapter describes how to manage and monitor the OSGi framework managed by JEUS OSGi by using the console tool.



For detailed explanations of the commands used in the examples, refer to "OSGi Commands" in JEUS Reference Guide.

## 3.1. Monitoring Bundles

The following is an example of checking the list of bundles installed in the OSGi framework by using the console tool.

```
[MASTER]domain1.adminServer>show-osgi-framework-info -server server1 -framework framework-0 -lb
OSGi Framework [framework-0] Information in the server[framework-0]
Framework [framework-0] Overview
=====
+-----+-----+-----+-----+-----+-----+
|          System Bundle Name          | Version |
+-----+-----+-----+-----+-----+
| org.apache.felix.framework          | 5.4.0   |
+-----+-----+-----+-----+-----+
=====

Installed Bundles in the framework [framework-0].
=====
+---+---+---+---+---+---+
| Id | Symbolic Name | Version | Location | State | Start-level |
+---+---+---+---+---+---+
| 0  | org.apache.felix.framework | 5.4.0 | System Bundle | ACTIVE | 0 |
+---+---+---+---+---+---+
| 2  | org.ops4j.pax.url.mvn | 1.3.7 | https://repo.maven.apache.org/maven2/org/ops4j/pax/url/pax-url-mvn-1.3.7.jar | ACTIVE | 30 |
+---+---+---+---+---+---+
=====
[MASTER]domain1.adminServer>
```

## 3.2. Installing Bundles

The following is an example of installing bundles in the OSGi framework by using the console tool.

```
[MASTER]domain1.adminServer>install-bundle -location
https://repo.maven.apache.org/maven2/org/ops4j/pax/url/pax-url-mvn-1.3.7.jar
-framework framework-0 -server server1
```

```
Installation completed successfully.  
[MASTER]domain1.adminServer>
```

### 3.3. Starting Bundles

The following is an example of starting bundles installed in the OSGi framework by using the console tool.

```
[MASTER]domain1.adminServer>start-bundle -name org.ops4j.pax.url.mvn -framework framework-0 -server  
server1  
The bundle has been started successfully.  
[MASTER]domain1.adminServer>
```

### 3.4. Stopping Bundles

The following is an example of stopping bundles installed in the OSGi framework by using the console tool.

```
[MASTER]domain1.adminServer>stop-bundle -name org.ops4j.pax.url.mvn -framework framework-0 -server  
server1  
The bundle has been stopped successfully.  
[MASTER]domain1.adminServer>
```

### 3.5. Updating Bundles

The following is an example of updating bundles installed in the OSGi framework by using the console tool.

```
[MASTER]domain1.adminServer>update-bundle -name org.ops4j.pax.url.mvn -framework framework-0 -server  
server1  
The bundle has been updated successfully.  
[MASTER]domain1.adminServer>
```

### 3.6. Deleting Bundles

The following is an example of deleting bundles installed in the OSGi framework by using the console tool.

```
[MASTER]domain1.adminServer>uninstall-bundle -name org.ops4j.pax.url.mvn -framework framework-0  
-server server1  
The bundle has been uninstalled successfully.
```

```
[MASTER]domain1.adminServer>
```

# 4. Web Application Deployment

This chapter describes how to deploy web applications with OSGi bundles included.

If a web application deployed by JEUS utilizes OSGi bundles as libraries, it is possible to package these bundles along with the application and install them in the designated OSGi framework during the deployment process. To accomplish this, the following tasks need to be performed when packaging the application.

1. Place the bundles to use in the application.
2. Specify the configuration related to OSGi in jeus-web-dd.xml.

Example of jeus-web-dd.xml configuration

```
<?xml version="1.0"?>
<jeus-web-dd xmlns="http://www.tmaxsoft.com/xml/ns/jeus">
  ....
  <osgi>
    <!-- Framework Name -->
    <framework-name>fwk-0</framework-name>
    <!-- Initial Bundles Directory Settings -->
    <install-initial-bundles>true</install-initial-bundles>
    <initial-bundles-directory-location>WEB-INF/osgi</initial-bundles-directory-location>
    <default-start-level>11</default-start-level>
    <fail-on-error>true</fail-on-error>
    <!-- Class Loading Delegation Settings -->
    <class-loading-delegation>
      <application-bundle>
        <bundle-symbolic-name>jeus.osgi.hello-api</bundle-symbolic-name>
        <version>1.0.0.Final</version>
      </application-bundle>
    </class-loading-delegation>
  </osgi>
  ....
</jeus-web-dd>
```

The configuration items are as follows.

- Framework Name

Specifies the name of OSGi framework to use in the web application.

- Configuration items for Initial Bundles Directory

In JEUS, the directory within a web application where bundles are located is referred to as the Initial Bundles Directory. You can control the location of this directory and its functionality by using configuration items of jeus-web-dd.xml. If the initial bundles directory is specified for use, the designated location is checked and the included bundles are installed in the specified OSGi framework. **Please note that when deploying the web application, the bundles included within it will be installed in the specified OSGi framework. Conversely, when undeploying the application, these bundles will be removed.**

The configurable items for the initial bundles directory are as follows.

Item	Description
Install Initial Bundles	Decides whether to install bundles in the initial bundles directory when deploying the web application.
Initial Bundles Directory Location	Specifies the location of the initial bundles directory.
Default Start Level	Specifies the default start level for the initial bundles directory.
Fail On Error	Decides whether to regard the failed installation of bundles as an error.

- Configuration items for Class Loading Delegation

These configuration items are set up when the web application needs to reference classes from the bundles installed in the OSGi framework. By specifying the bundle symbolic name and version of the desired bundles, the application can utilize the referenced classes. Multiple bundles can be specified for referencing, and in such cases, the search for the class is performed in the specified order.



1. The OSGi bundles are searched for the classes or resources only if they cannot be found in the hierarchical structure of JEUS class loader.
2. If a bundle containing classes referenced by the web application is updated, a class-related error may occur in the web application. In this case, you must redeploy or reload the web application. To mitigate the occurrence of this error, it is recommended to use API bundles for reference because they are relatively little changed.

If the application including the OSGi bundles is deployed, the related configuration can be checked by using the console commands.

## Using the console tool

By utilizing console commands to retrieve detailed information about the web application, you can also access the configuration details of the associated OSGi.

```
[MASTER]domain1.adminServer>application-info -server server1 -id hello-osgi.war -type WAR
There are no EJBs in this module.
General information about the web module [hello-osgi].
=====
+-----+-----+-----+-----+
| Module Name | Unique Module Name | Context Path | Target Session Cluster |
+-----+-----+-----+-----+
| hello-osgi | hello-osgi          | /hello-osgi |                          |
+-----+-----+-----+-----+
=====

Servlets
```

```
=====
```

Name	Class	State	Count	Attribute	RegType	URLPatterns
appServlet	org.springframework.web.servlet.DispatcherServlet	READY	0	SYNC	WEB_XML	/

```
=====
```

...

...

#### OSGi Application Information

```
=====
```

Attribute Name	Value
Framework Name	framework-0
Install Initial Bundles	true
Initial Bundles Directory Location	/home/user/workspace/jeus/jeus/target/jeus/domain1/osgi/servers/server1/.workspace/deployed/hello-osgi.war/hello-osgi_war___/WEB-INF/osgi
Default Start Level	11
Fail On Error	true

```
=====
```

#### Delegate Bundles

```
=====
```

Bundle Symbolic Name	Version
jeus.osgi.hello-api	1.0.0.Final

```
=====
```

#### EJBs

```
=====
```

Bean Name	Type	Local Export Name	Remote Export Name
(No data available)			

```
=====
```

[MASTER]domain1.adminServer>