JEUS Introduction Guide

JEUS 9

TMAXSOFT

Copyright

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

Trademarks

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. JEUS Introduction

This chapter describes the basics of JEUS and Jakarta EE specification. It also explains the concepts of a JEUS system, its components, and features of each edition.

1.1. Overview

Java Enterprise User Solution (JEUS) provides a platform for the development, operation, and execution of applications in a web environment. It also supports Java-based web application services and their management. JEUS provides a platform and the following components that are required to execute Jakarta EE applications.

- EJB containers
- Web containers (JSPs/Servlet engines)
- Security modules
- Naming Server
- Transaction managers
- JDBC Connection Pool
- Session managers

Jakarta EE

JEUS conforms to the Eclipse Foundation's Jakarta EE specification, and is Jakarta EE 9 certified.

Jakarta EE is briefly described in the following quotes from the official Jakarta EE website.

"Jakarta EE is a set of specifications that enables the world wide community of java developers to work on cloud native java enterprise applications. The specifications are developed by well known industry leaders that instills confidence in technology developers and consumers."

For more information about Jakarta EE, refer to the Jakarta EE website. (Jakarta EE Official Web Site)

The following table lists the features that are implemented in JEUS 9.

Spec	JEUS 9
Jakarta EE	Jakarta EE 9
Jakarta™ Activation	2.0
Jakarta™ Annotations	2.0
Jakarta™ Authentication	2.0
Jakarta™ Authorization	2.0

Spec	JEUS 9
Jakarta™ Batch	2.0
Jakarta™ Bean Validation	3.0
Jakarta™ Concurrency	2.0
Jakarta™ Connectors	2.0
Jakarta™ Contexts and Dependency Injection	3.0
Jakarta™ Debugging Support for Other Languages	2.0
Jakarta™ Dependency Injections	2.0
Jakarta™ Enterprise Beans	4.0
Jakarta™ Enterprise Web Services	2.0
Jakarta™ Expression Language	4.0
Jakarta™ Faces	3.0
Jakarta™ Interceptors	2.0
Jakarta™ JSON Binding	2.0
Jakarta™ JSON Processing	2.0
Jakarta™ Mail	2.0
Jakarta™ Managed Beans	2.0
Jakarta™ Messaging	3.0
Jakarta™ Persistence	3.0
Jakarta™ RESTful Web Services	3.0
Jakarta™ Security	2.0
Jakarta™ Server Pages	3.0
Jakarta™ Servlet	5.0
Jakarta™ SOAP with Attachments	2.0
Jakarta™ Standard Tag Libraries	2.0
Jakarta™ Transactions	2.0
Jakarta™ WebSocket	2.0
Jakarta™ Web Services Metadata	3.0
Jakarta™ XML Binding	3.0
Jakarta™ XML Web Services	3.0
Extensible Stylesheet Language Transformations(XSLT)	1.0
Java API for XML Processing(JAXP)	Supported

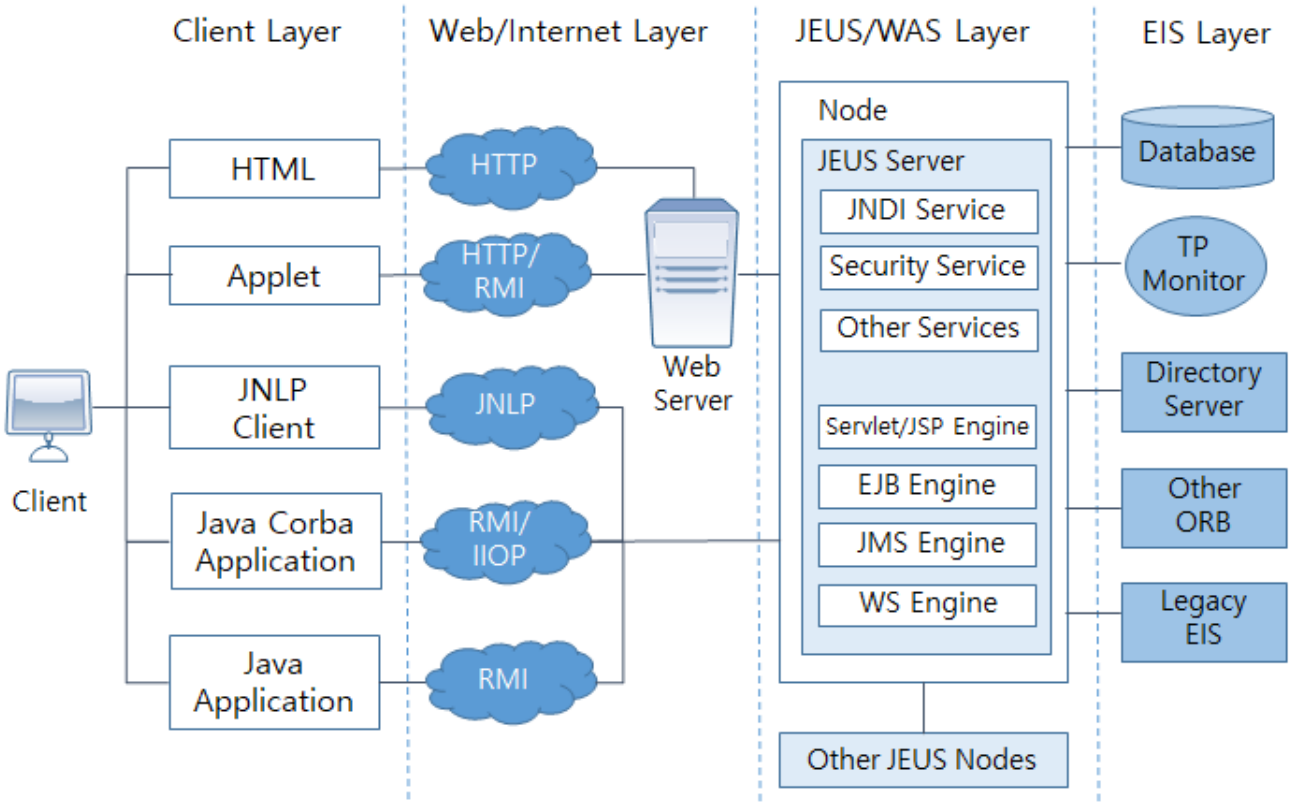
Spec	JEUS 9
Java Authentication and Authorization Service(JAAS)	1.0.1
Java Database Connectivity(JDBC) API	4.2
Java Naming and Directory Interface(JNDI) API	1.2.1
Java Transaction Service(JTS)	1.0
Simple Object Access Protocol(SOAP)	1.1/1.2
Streaming API for XML(StAX)	Supported
Universal Description,Discovery Intergration(UDDI)	2.0/3.0
Web Services Description Language(WSDL)	1.1/2.0
WebServer	WebtoB 5.0
HTTP	1.0/1.1/2.0
CGI	1.1
PHP	3.x/4.x/5.x
RMI-IIOP	Supported
EJB to CORBA Mapping	1.1
IBM MQ	Supported
Sonic MQ	Supported
WS-I Basic Profile	1.1
WS-Policy	1.5
WS-Policy Attachment	1.5
WS-Addressing	1.0
WS-Security	1.1
WS-Security Policy	1.2
WS-Trust	1.4
WS-Secure Conversation	1.4
WS-Reliable Messaging	1.2
WS-AtomicTransaction	1.2
WS-Coordination	1.2
OTS	Supported
Java IDL API	Supported

Spec	JEUS 9
IDE Tool	Not supported
GUI Tool	Not supported
Monitoring Tool	Console Tool
JDK	8, 11, 17

1. For more information about the specifications, refer to Jakarta EE Specifications.
2. Some editions of JEUS may not implement all of the features. For more information, refer to the documents for each JEUS edition.

1.2. System Concepts and Roles

The following figure shows how JEUS integrates with other Web servers or DBMSs to provide enterprise application solutions.



JEUS Concept and Component Layers

The following are the four layers shown in the previous figure.

- **Client Layer**

  Consists of Java applications and native applications. End users use various clients to access WAS services, and the clients use various protocols to access WAS services.

- **Web/Internet Layer**

  Consists of the web server and protocols used by clients and WAS. This layer deals with static contents and load balancing.

- **JEUS/WAS Layer**

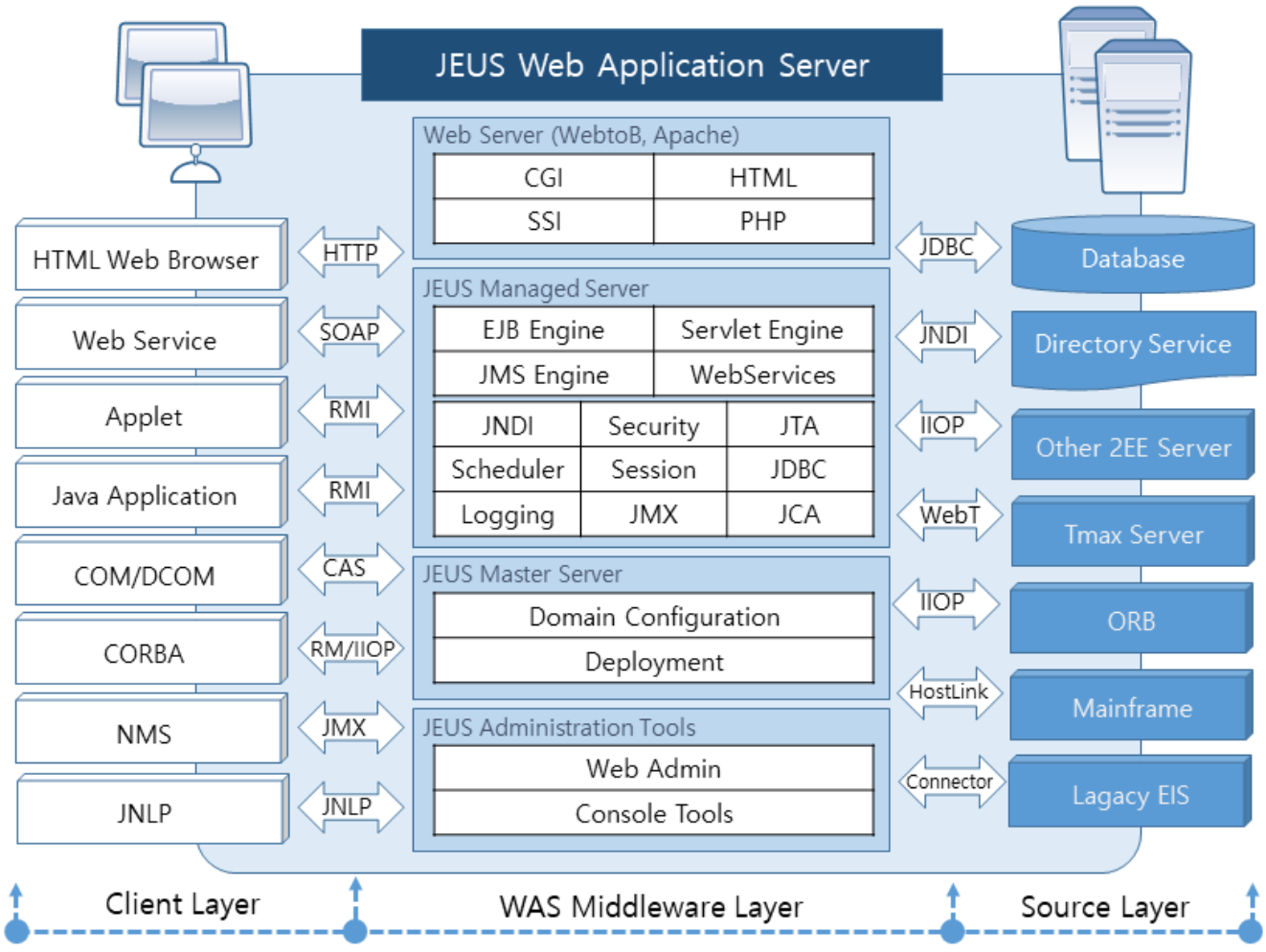
  Consists of Java-based middleware and deals with requests from the web and client layers.

- **EIS Layer**

  Consists of data and existing legacy services. WAS interoperates with legacy services through various mechanisms such as JDBC, directory services, and Jakarta EE connector.

1.3. Architecture and Components

JEUS is made up of various components. These components operate differently depending on which communication technologies they use to communicate with clients, data storage devices, and JEUS, as shown in the following figure (JEUS Web Application Architecture).



JEUS Web Application Architecture

The client layer in the previous figure shows a diverse set of client applications and communication

protocols. The source layer shows various types of data storage devices. The WAS middleware layer contains JEUS and the web server. The web server connects with client applications and is tightly coupled with the web application server. The web gateway (WebT) connects the web application server to the TP-Monitor (Tmax server), and the mainframe gateway (Host-Link) provides connection between a mainframe system and TP-Monitor.

Each layer (Client Layer, WAS Middleware Layer, Source Layer) is explained in more detail in the following sections.

1.3.1. Client Layer

The client layer consists of remote and local applications that use JEUS.

It contains the following components.

Client Layer	Description
Web browser	The most common client application is a standard web browser that sends requests to a JEUS Servlet Engine and WebtoB Light Web server to retrieve HTML contents. The communication protocol used is HTTP.
Web Services	Provides web service implementations.
Applet	Provides a special applet container that can access JEUS's components.
Java Application	Standalone Java applications are executed by using RMIs within client containers provided by JEUS. Clients are called *application clients* in the Jakarta EE specification.
COM/DCOM	EJBs can be invoked as COM objects in Microsoft Windows.
CORBA	CORBA applications can use JEUS via RMI/IIOP.
NMS	Network Management System manages and uses JEUS via JMX.
JNLP	Java Network Launching Protocol (JNLP) clients are supported.

1.3.2. WAS Middleware Layer

The JEUS WAS layer in JEUS Web Application Architecture corresponds to JEUS 9. Its components are as follows:

- **JEUS Master Server**

  Each domain must have a special server called Master Server. The Master Server centrally manage the configurations that define the relationship between servers of a domain, and applications and resources in the domain. It controls and monitors servers by communicating with the management tools, such as jeusadmin.

Service	Description
Domain Configuration	Manages domain configurations.
Application Management	Manages domain applications.
Administration	Centrally monitors and controls all domain servers, services, applications, and resources by using jeusadmin.

- **JEUS Managed Server(MS)**

  An MS contains several types of engines configured within a JEUS system. There are the following four engine types.

Engine / Service	Description
EJB Engine	Runs EJB applications.
Servlet Engine	Web container that handles both JSP/Servlet applications and static contents like HTML.
JMS Engine	Supports JMS structures.
Web Services Engine	JEUS Web server that acts as the front end of the servlet engine.
JNDI Service	Naming system.
Security Service	Handles authentication and authority.
JTA	Provides complete transactions for various applications running on Web application servers.
Scheduler	Supports timers that trigger an event at a predefined time.
Session Manager	Reliably stores client session information when clustering is needed.
JDBC	Configurable database connection pool.
Logging	Stores and records jobs that have been executed on JEUS.
JMX	Allows NMS/JMX clients to manage JEUS.
JCA	JCA stands for Jakarta™ Connector Architecture. JCA allows any legacy Enterprise Application Integration (EAI) solution to integrate with a legacy EIS if the EAI solution supports JCA.

- **Web Server (WebtoB, Apache)**

  The web server handles the transfer of static content, such as HTML, and dynamic content, such as CGI. It serves as the front end for the servlet engine. TmaxSoft offers two versions of its web server product, WebtoB: the full version, which fully supports all web server features, and a limited version, JEUS Web Server, which supports only a subset of WebtoB's features.

JEUS Web server comes with JEUS but not with WebtoB. As an alternative, open-source web server Apache may be used with JEUS.

- **JEUS Administration Tools**

  The following management tool is available for use.

Tool	Description
Console tool (jeusadmin)	Command-line based console tool for controlling the JEUS server.

1.3.3. Source Layer

The source layer shown in JEUS Web Application Architecture shows the back-end resources and data storage used by JEUS system. The following are the components.

Source Layer	Description
Database	Can be accessed by JEUS through JDBC.
Directory Service	JNDI is used to access services like LDAP.
Other Jakarta EE Server	JEUS interoperates with Jakarta EE Servers from other vendors.
Tmax Server	TmaxSoft's TP-monitor. WebT API library is used to access JEUS or Tmax.
ORB	Can be referenced by using IIOP (Internet Inter-ORB Protocol).
Mainframe	Accessed by using Host-Link (connector) products.
Legacy EIS	Supports JCA and interoperates with JEUS.

1.4. Interoperable Modules

Interoperability refers to the ability of two or more systems, such as computers, communication devices, networks, software, and other information technology components, to interact with one another and exchange data according to a prescribed method in order to achieve predictable results (ISO ITC-215). JEUS supports different communication protocols and technologies such as web services, JNLP, and RMI-IIOP.

The following are the modules provided for JEUS interoperability.

Module	Description
RMI-IIOP	RMI technology that uses Internet Inter-ORB Protocol (IIOP) and enables distributed CORBA computing jobs in Java.
JEUS	Interoperates with other vendors' web applications and services.
WebT	Provides a connection between TP-Monitor and JEUS.
Host-Link	Adapter module that enables clients to use services that are running on an EIS.

Module	Description
JCA	Enables JEUS and JEUS clients to virtually interoperate with any legacy EIS.

1.5. Edition

The following table shows the main features of each JEUS 9 edition.

Edition	Main Features
JEUS Standard Edition	◦ Jakarta™ EE 9 support ◦ JEUS Server ◦ JEUS Web Server ◦ Java Database Connectivity (JDBC) support ◦ Java Naming and Directory Interface (JNDI) support ◦ Java Management Extensions (JMX) support ◦ JEUS JDBC Connection Pool ◦ JEUS Security ◦ JEUS Server Clustering ◦ JEUS Transaction Manager ◦ JEUS Administration Tools
JEUS Enterprise Edition	◦ JEUS Standard Edition ◦ JEUS Session Clustering ◦ Jakarta™ Messaging(JMS) Server Clustering

1. All features of the previous table are covered in the JEUS manuals. Refer to this table to check whether the features described in each guide are available for use. JEUS includes a Trial license by default.
2. A cloud license is provided for Standard Edition and Enterprise Edition respectively.

2. JEUS Environment

This chapter describes JEUS management tools, directory structure, environment variables, and XML configuration files.

2.1. Management Tools

The following tools are used to access JEUS.

Tool	Description
startMasterServer	Basic tool for starting JEUS Master Server. For more information, refer to "JEUS Reference Guide".
startManagedServer	Basic tool for starting JEUS Managed Server. For more information, refer to "JEUS Reference Guide".
jeusadmin	Used to control JEUS from a command prompt. For more information, refer to "jeusadmin" in JEUS Reference Guide.

Other than those previously mentioned tools, you can use additional commands to shut down the Master Server and Managed Server. For more information, see "JEUS Reference Guide"

2.2. Directory Structure

The following is the directory structre of the installed JEUS.

```
{JEUS_HOME}
    |-- bin
    |    |--[01]startMasterServer
    |    |--[01]startManagedServer
    |    |--[01]stopServer
    |    |--[01]jeusadmin
    |--derby
    |--docs
    |--lib
    |   |--shared
    |        |--[X]libraries.xml
    |--license
    |--setup
    |--templates
    |--samples
    |--webserver
    |--domains
          |--<domain_name>
                 |--.applications
                 |--.deploymentplans
```

```
                  |--.libraries
                  |--bin
                  |--config
                  |--lib
                  |   |--application
                  |--servers
                      |--<server_name>
                             |--.workspace
                             |     |--deployed
                             |     |--tmp
                             |     |--web-nio
                             |     |--tmlog
                             |--bin
                             |--lib
                             |   |--application
                             |--logs

* Legend
- [01]: binary or executable file
- [X] : XML document
- [J] : JAR file
- [T] : Text file
- [C] : Class file
- [V] : java source file
- [DD] : deployment descriptor
```

The following describes each directory and file.

{JEUS_HOME}

The root directory of JEUS. The actual directory name and location are selected during installation.

bin

Contains the scripts to start and stop servers. The scripts are startMasterServer, startManagedServer, and stopServer. This directory also contains the executable files including the JEUS console tool 'jeusadmin'.

derby

Contains Apache Derby, which helps users build sample applications or perform tests.

docs

Contains Javadoc documentations for the APIs provided by JEUS.

lib

Contains the libraries used to start JEUS. Users only need to access the shared directory.

Directory	Description
shared	Contains the libraries used by applications. Library information must be added to libraries.xml in order to use the libraries in the shared directory. Reference information about the library must also be specified in the JEUS deployment descriptor of the application that will use the library. For more information about shared libraries, see "Shared Libraries" in JEUS Applications & Deployment Guide.

license

Contains JEUS license files that are needed to execute JEUS.

setup

Contains the files needed to set up the environment to use JEUS after JEUS has been installed.

templates

Contains configuration file templates.

samples

Contains example files for JEUS.

webserver

Directory where the JEUS Web server is installed during the JEUS installation. For details, refer to "JEUS Web Engine Guide".

domains

Each domain contains a file named nodes.xml that contains the information about the nodes used from DOMAIN_HOME and JEUS_HOME.

DOMAIN_HOME contains the following files and directories:

- .applications

  Contains application files managed by the domain.

  These can only be added or deleted by using the **install-application** and **uninstall-application** commands. This is a JEUS system directory with restricted user access. For detailed information about each command, refer to "install-application" and "uninstall-application" in JEUS Reference Guide.

- .deploymentplans

  Contains deployment plan files managed by the domain.

  These can only be added or deleted by using the **install-deployment-plan** and **uninstall-deployment-plan** commands. This is a JEUS system directory with restricted user access. For detailed information about each command, refer to "install-deployment-plan" and "uninstall-deployment-plan" in JEUS Reference Guide.

- .libraries

  Contains library files managed by the domain.

  These can only be added or deleted by using the **install-library** and **uninstall-library** commands. This is a JEUS system directory with restricted user access. For detailed information about each command, refer to "install-library" and "uninstall-library" in JEUS Reference Guide.

- bin

  Contains the scripts to start or stop the Master Server and Managed Server of the domain. The functions of the scripts are the same as startMasterServer, startManagedServer, and stopServer scripts in the 'JEUS_HOME/bin' directory, except that the users don’t need to specify the file names.

- config

  Contains backup files that store changes to domain.xml, a domain configuration file. For detailed information about domain configurations, refer to "Changing Domain Settings" in JEUS Domain Guide.

Directory	Description
security	◦ SYSTEM_DOMAIN: Contains security domain files, including accounts.xml and policies.xml. Each XML file can be dynamically modified by using jeusadmin. For detailed information about security domain configurations, refer to "Configuring the Security System Domain" in JEUS Security Guide. ◦ security-domains.xml: Contains security domain configuration for JEUS. ◦ security.key: Contains the keys for symmetric key encryption. They are created when 'JEUS_HOME/bin/encryption' is executed. For detailed information about the security.key file, refer to "Configuring Password Security" in JEUS Security Guide. ◦ policy: Contains the Java permissions configuration file. This is used by Java SE Security Manager, separate from the JEUS security system.
servlet	◦ web.xml: This file is used when a web module does not have a separate web.xml file. By default, the file is empty. ◦ webcommon.xml: Settings that apply to all Web modules of the Web engines in the domain. For detailed information about the file, refer to "Directory Structure" in JEUS Web Engine Guide.

- lib/application

  Contains the shared application libraries for the domain.

  If a library conflicts with an application library in the SERVER_HOME directory, it is overridden

by 'SERVER_HOME/lib/application' and a warning message appears. For detailed information about the 'lib/application' directory, refer to "lib/application Directory" in JEUS Applications & Deployment Guide.

- servers

  Create the SERVER_HOME directory by using the server name in this directory. For detailed information about the directory structure, refer to "Server Directory Structure" in JEUS Server Guide.

Directory	Description
.workspace	Contains workspaces used by each JEUS server. Cannot be modified by the user.
bin	Contains scripts for starting and stopping the server. The scripts execute the same functions as those in 'JEUS_HOME/bin', but they do not require the domain and server names. ◦ Master Server: uses startMasterServer and stopServer. ◦ Managed Server: uses startManagedServer and stopserver.
lib/application	Contains application libraries for the server. This directory takes precedence over the domain-level library directory (DOMAIN_HOME/lib/application). If a library conflicts with an application library in the DOMAIN_HOME/lib/application, the file in this directory overrides that in DOMAIN_HOME and a warning message appears. For detailed information about lib/application, refer to "lib/application Directory" in JEUS Applications & Deployment Guide.
logs	Includes launcher logs, server logs, and access log files. For detailed information, refer to "Logging" in JEUS Server Guide.

2.3. Environment Variables

All JEUS environment variables are configured in 'JEUS_HOM/bin/jeus.properties' and are used by the scripts in 'JEUS_HOME/bin'.

The following table shows the environment variables used in JEUS.

Environment Variables	Description
JEUS_HOME	Directory where JEUS is installed. This variable is required. Example: JEUS_HOME=/home/jeus/jeus9
JAVA_HOME	Directory where JDK is installed. Example: JAVA_HOME=/usr/jdk1.8

The variables can be changed when necessary, but they cannot be used in an XML configuration file. All environment variables are set to default values during installation. In most cases, the default values can be used without any changes.

Changing the environment variables varies according to the OS. Refer to each OS manual for more information.

2.4. Environment Configuration Files

JEUS environment is configured by using XML. The configurations can be changed manually or with tools.

The following tables describe the XML configuration files and their locations.

- domain.xml (jeus-domain.xsd, ejb-engine.xsd, web-engine.xsd, jms-engine.xsd)

Location	JEUS_HOME/domains/<domain_name>/config/
Description	Basic configuration file that manages JEUS Manager and nodes.
Reference Manual	"JEUS Domain Guide" and "JEUS Server Guide"

- jeus-web-dd.xml (jeus-web-dd.xsd)

Location	WEB-INF directory in the web application archive.
Description	JEUS Web application (servlet app) deployment descriptor file.
Reference Manual	"JEUS Web Engine Guide"

- jeus-ejb-dd.xml (jeus-ejb-dd.xsd)

Location	META-INF directory in the EJB application archive.
Description	JEUS EJB module deployment descriptor file.
Reference Manual	"JEUS EJB Guide"

- jeus-client-dd.xml (jeus-client-dd.xsd)

Location	WEB-INF directory in the client application archive.
Description	Application client DD (Deployment Descriptors) file.
Reference Manual	"JEUS Application Client Guide"

- jeus-connector-dd.xml (jeus-connector-dd.xsd)

Location	META-INF directory in the resource adapter archive.
Description	Application client DD (Deployment Descriptors) file.
Reference Manual	"JEUS JCA Guide"

- policies.xml (policies.xsd)

Location	JEUS_HOME/domains/<*domainname*>/config/security
Description	JEUS security policies.
Reference Manual	"JEUS Security Guide"

- accounts.xml (accounts.xsd)

Location	JEUS_HOME/domains/<*domainname*>/config/security
Description	JEUS security accounts.
Reference Manual	"JEUS Security Guide"

- jeus-web-dd.xml (jeus-web-dd.xsd), jeus-ejb-dd.xml (jeus-ejb-dd.xsd), jeus-client-dd.xml (jeus-client-dd.xsd)

Location	META-INF directory in the Web service client archive
Description	Web service client information.
Reference Manual	"JEUS Web Service Guide"

- jeus-webservices-config.xml (jeus-webservices-config.xsd)

Location	META-INF directory in the Web service client archive
Description	Ant tasks for Web services
Reference Manual	"JEUS Web Service Guide"

1. The Jakarta EE standard descriptor files, web.xml and ejb-jar.xml, are also used. For information about each file, refer to the Jakarta EE specification.
2. All XML schema files are located in the 'JEUS_HOME/lib/schemas/jeus/supportLocale/ko' directory.

3. JEUS Manual Organization

This chapter describes the JEUS manuals and abbreviations used in the manual. Read this chapter carefully because it contains important information about the JEUS manual.

3.1. Overview

JEUS manuals can be accessed in the following three ways.

- HTML documents provided with the software.
- PDF files on the JEUS installer CD.

  Viewing a PDF require Adobe Acrobat Reader or another application that can read PDFs. Visit Download Adobe Reader to download the Adobe Acrobat Reader.

- Updated software or guides in the TechNet site of TmaxSoft (TechNet).

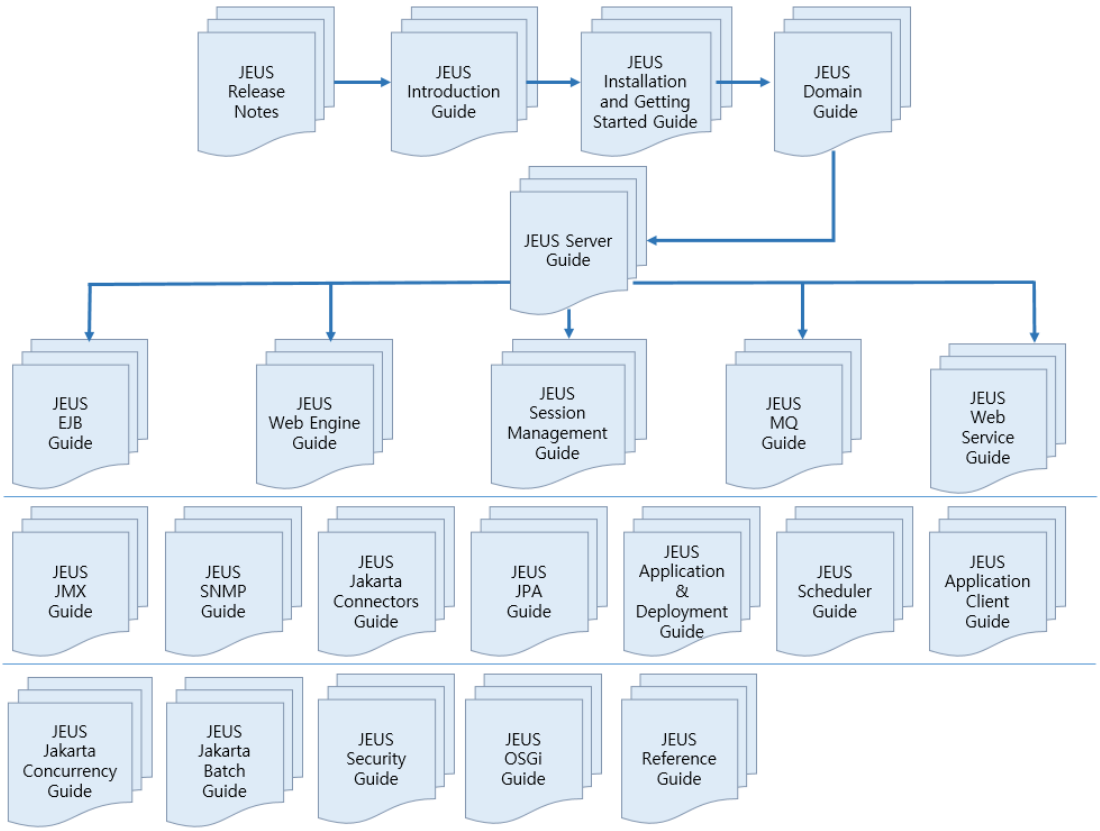
JEUS manuals:

- Are intended for Jakarta EE professionals who understand Java and Jakarta EE technology.
- Are organized into individual and loosely related documents.
- Generally follow a standard document format with a few exceptions.
- Are structured in a wide variety of ways.

Knowledge of Java and Jakarta EE technologies is required to understand this document. Useful online documents and specifications are available at the Java website, Oracle Technical Documentation. JEUS manuals do not cover all the information described in the Jakarta EE specifications, but only include JEUS related information.

3.2. Manual Organization

The following figure shows the complete document organization of the JEUS manuals. The arrows indicate the preferred order of reading for the first-time users. The JEUS manuals consist of 22 documents.



JEUS Manual Organization

The contents of each document are outlined in the following list. Refer to this list to quickly locate a specific topic.

- **"JEUS Release Notes"**

  Describes new features of released JEUS, and how to upgrade the previous version.

  - New JEUS features
  - Upgrading from a previous version

- **"JEUS Introduction Guide"**

  Describes JEUS and its architecture and components.

  - Introduction to JEUS
  - JEUS environment
  - JEUS documentation

- **"Installation and Getting Started Guide"**

  Describes how to install and start JEUS with examples for each platform.

  - Installing Java on UNIX
  - Installing JEUS on UNIX
  - Descriptions for sample applications (Getting Started)

- JEUS system tutorial
- EJB tutorial
- Servlet/JSP tutorial
- Web service tutorial

- **"JEUS Domain Guide"**

  Outlines the structure of JEUS domains and their configurations. Those who need to understand JEUS must read this guide.

  - Concepts and components of a domain
  - Structure and creation of a domain
  - Changing and applying domain configurations
  - Controlling and changing the state of servers with domains
  - Concepts of a cluster with domains
  - Handling abnormal termination of servers that manage domains
  - Domain securities

- **"JEUS Server Guide"**

  This is the most important guide for JEUS administrators who manage JEUS.

  - JEUS components and services
  - Configuring JEUS components
  - JEUS server controls and monitoring
  - Basic JEUS JNDI concepts and application development methods
  - Integrating and configuring external resources with JEUS
  - Using connection pools and other features provided by JEUS
  - JEUS transaction manager and related components
  - JEUS logging system
  - Connection pool settings for main JDBC drivers

"JEUS Server Guide" contains very broad contents such as JEUS security, naming, and transaction managers. It contains many disparate topics such as JEUS security, Naming, and Transaction Manager. Although the topics vary, documents are organized as they are because the components are configured using the same configuration file and processed on the same JVM.

- **"JEUS EJB Guide"**

  Describes how to deploy JEUS EJB engines and modules.

- Overview of JEUS EJB
- Configuring, controlling, monitoring, and tuning the JEUS EJB engine
- Managing, deploying, controlling, and monitoring EJB modules
- General EJB features and configurations
- JEUS EJB security
- JEUS EJB security interoperability
- JEUS EJB clustering
- Session EJB
- Entity EJB
- Message Driven Beans
- EJB clients
- Using Ant with JEUS EJBs

- **"JEUS Web Engine Guide"**

  Describes how to manage JEUS Web containers. It discusses about the Jakarta EE WAR archive and how to manage and deploy Servlets/JSPs.

  - Overview of JEUS Web engine and environment configurations
  - JEUS Web connection management
  - JEUS Web context (Web applications/WAR files) structure and deployment, controlling, monitoring, and tuning of Web contexts
  - JEUS Web engine features and settings
  - Web server connection and clustering (WebtoB, Apache and built-in HTTP server connections and clusters)
  - Virtual hosting
  - Concepts and use of JEUS WebCache
  - Concepts and use of reverse proxies
  - Default settings and operations of dynamic class loading

- **"JEUS Session Management Guide"**

  Describes the concepts and settings of the session manager and session server. This guide is intended for system administrators and developers who maintain and share sessions in a clustered environment or on a single server.

  - The structure, operations, configurations, and tuning of sessions
  - The structure, operations, and configurations of distributed session servers that enable session tracking in clustered environments

- **"JEUS MQ Guide"**

  Describes the JEUS messaging system (JMS).

- Overview of JEUS JMS
- Configuring, monitoring, and controlling the JMS engine environment
- JMS programming in JEUS
- JEUS MQ clustering methods
- JEUS MQ special functions

- **"JEUS Web Service Guide"**

  Describes Web services in JEUS.

  - Overview of JEUS Web services
  - Generating Web service endpoints
  - Calling Web services
  - Implementing client applications that use Web services
  - Data types and JEUS Web services
  - Using Ant with Web services
  - Standard binding declarations and customizations
  - Handler framework
  - Provider and dispatch interfaces
  - Asynchronous Web services
  - Using MIME attachment message transmission and Fast Infoset with Web services
  - WS-Policy, WS-Addressing, WS-ReliableMessaging, and WS-Transaction
  - Using UDDI
  - Web service security
  - XML and Web services

- **"JEUS JMX Guide"**

  Describes how to manage JEUS using JMX.

  - Configuring JEUS JMX Manager
  - Developing JMX applications
  - JMX API reference

- **"JEUS SNMP Guide"**

  Describes how to monitor JEUS using the SNMP protocol.

  - Overview of JEUS SNMP agent
  - SNMP agent organization
  - Programming with JEUS SNMP
  - JEUS SNMP MIB

- **"JEUS Jakarta Connectors Guide"**

  Describes how to integrate JEUS and legacy systems.

  - Overview of connectors
  - Connector packaging
  - Using and tuning connectors
- **"JEUS JPA Guide"**

  Describes the necessary configurations for using TopLink Essential with JEUS.

  - Introduction to Jakarta Persistence API
  - Configuring providers
  - Configuring JEUS
- **"JEUS Scheduler Guide"**

  Describes the features of JEUS scheduler.

  - Configuring scheduler services
  - Programming with scheduler services
- **"JEUS Applications & Deployment Guide"**

  Describes how to use tools to deploy Jakarta EE applications to JEUS.

  - Managing applications in a domain
  - Graceful undeployment and redeployment
  - Overview of modules, applications, and shared libraries
  - Creating and deploying Jakarta EE application files
- **"JEUS Application Client Guide"**

  Describes interoperability between Jakarta EE clients and JEUS.

  - Jakarta EE application clients
  - Applet clients
  - JNLP clients
- **"JEUS Security Guide"**

  Describes how to configure and operate the security system in JEUS, and explains security-related programming.

  - Overview and configurations of the security system
  - Configuring security for applications and modules
  - Programming with the security API
  - Developing custom security services

    - Using JACC Provider
    - Using JAAS
- **"JEUS Jakarta Concurrency Guide"**

  Describes Jakarta concurrency programming in JEUS.

    - Overview and configurations of concurrency utilities for Jakarta EE
    - Programming with the APIs of concurrency utilities for Jakarta EE
- **"JEUS Jakarta Batch Guide"**

  Describes Jakarta Batch programming.

    - Overview and configurations of Jakarta Batch
    - Programming with Jakarta Batch
- **"JEUS OSGi Guide"**

  Describes the features of OSGi provided in JEUS.

    - General OSGi features and configurations
    - Usage of OSGi features in web applications
- **"JEUS Reference Guide"**

  Contains references that are helpful when using JEUS.

    - Information about system properties
    - Using console commands
    - Using Ant tasks
    - APIs used in JEUS