JEUS 소개

JEUS 9.1

TMAXSOFT

저작권 공지

회사 정보

(주)티맥스소프트

주소 : 경기도 성남시 분당구 정자일로 45, 티맥스소프트타워

기술 서비스 센터: 1544-8629

홈페이지: https://www.tmaxsoft.com

제한된 권리

상표 공지

본 사용설명서에 기재된 회사, 시스템, 제품 이름 등에 반드시 상표 표시(™, ®)를 하지는 않습니다.

오픈 소스 소프트웨어 공지

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다.: APACHE2.0, CDDL1.0, EDL1.0, OPEN SYMPHONY SOFTWARE1.1, TRILEAD-SSH2, Bouncy Castle, BSD, MIT, SIL OPEN FONT1.1

관련 상세 정보는 제품의 다음 디렉터리에 기재된 사항을 참고하시기 바랍니다. : ${INSTALL_PATH}/license/oss_licenses

유지 보수

구분	지원항목	서비스 내용
제품지원	패치 & 업그레이드	무상 패치 서비스 제공 메이저 버전 업그레이드 시 할인 혜택 웹 지원을 통한 패치 내역 제공
기술 지원 - 기본 서비스	장애 지원	장애 발생 시 원인 분석 및 조치 Service Desk팀 → 기술팀 → R&D의 3단계 장애 분석 및 조치
	일상 지원(온라인 지원)	E-mail, 전화, 원격, 웹 사이트 등 온라인 자원을 통한 질의 응답 서비스
	고객 맞춤 지원(방문 지원)	고객의 요청으로 수행하는 방문 지원 서비스
기술 지원 - 옵션 서비스	예방 지원	정기 점검을 통한 시스템 운영현황 보고 및 장애 예방 ◦ 관리자 또는 운영자의 요구사항 수렴 ◦ 운영 현황(시스템, 엔진 운영) 보고서 제공 ◦ 필요 시 시스템 개선 권장 사항 보고
유지 보수 비용 및 기간	계약 시 별도 협의	계약 시 EOL/EOS 문서 제공

안내서 이력

제품 버전	안내서 버전	발행일	비고
JEUS 9.1	3.3.1	2025-09-30	GA 버전
JEUS 9 Fix#1	3.2.2	2025-07-10	JDK 21 지원 내용 추가
JEUS 9 Fix#1	3.2.1	2025-06-30	RC 버전
JEUS 9	3.1.2	2025-01-03	-
JEUS 9	3.1.1	2024-09-27	-

목차

1. JEUS 소개

본 장에서는 JEUS에 대한 기본적인 이해와 Jakarta EE 스펙에 대해서 설명하고, JEUS 시스템의 개념 및 구성 요소와 에디션별 기능에 대해 기술한다.

1.1. 개요

JEUS(Java Enterprise User Solution)는 웹 환경에서 애플리케이션을 개발, 운용, 실행할 수 있는 플랫폼 역할을 하면서, 포괄적인 Java 기반의 웹 애플리케이션 서비스와 관리를 제공한다. JEUS는 Jakarta EE 애플리케이션을 구동할 때 필요한 플랫폼과 다음의 같은 구성 요소를 제공한다.

- EJB 컨테이너
- 웹 컨테이너(JSP/서블릿 엔진)
- 보안 모듈(Security Module)
- Naming Server
- 트랜잭션 매니저
- JDBC Connection Pool
- 세션 매니저

Jakarta EE

JEUS 9는 Eclipse Foundation의 Jakarta EE 스펙을 준수하며, Jakarta EE 9 인증을 획득했다.

다음은 Jakarta EE 공식 홈페이지에 있는 문구로, Jakarta EE에 대해 간략하게 소개를 하고 있다.

"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."

Jakarta EE에 대한 보다 자세한 정보를 알아보기 위해서는 Jakarta EE 공식 사이트를 참고한다.

다음은 JEUS 9에서 지원하는 기술에 대한 목록이다.

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)	지원

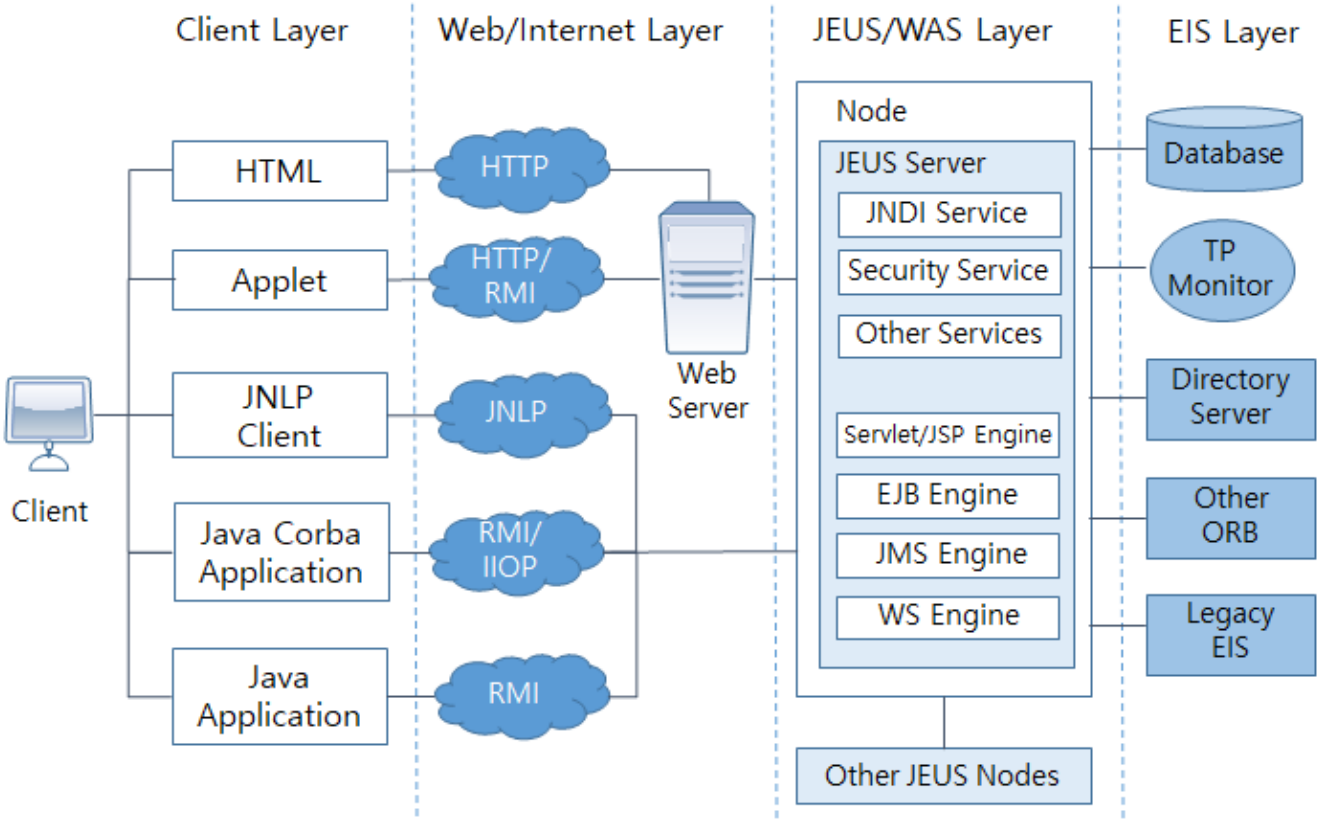
Spec	JEUS 9
Java Authentication and Authorization Service(JAAS)	1.0.1
Java Database Connectivity(JDBC) API	4.2
Java Naming and Directory Interface(JNDI) API	1.2.1
Java Transaction Service(JTS)	1.0
Simple Object Access Protocol(SOAP)	1.1/1.2
Streaming API for XML(StAX)	지원
Universal Description,Discovery 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	지원
EJB to CORBA Mapping	1.1
IBM MQ	지원
Sonic MQ	지원
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	지원
Java IDL API	지원

Spec	JEUS 9
IDE Tool	미지원
GUI Tool	미지원
Monitoring Tool	Console Tool
JDK	8, 11, 17, 21

1. 각 스펙에 대한 내용은 Jakarta EE 사양 문서에서 확인할 수 있다.
2. 사용하는 JEUS 에디션에 따라서 위 표에 나열된 모든 기능이 구현되지 않을 수 있다. 자세한 내용은 각 JEUS 에디션 소개 자료를 참고한다.

1.2. 시스템 개념과 역할

다음은 엔터프라이즈 애플리케이션 솔루션을 제공하기 위해서 JEUS가 다른 웹 서버나 DBMS 등과 어떻게 통합되는지 보여준다.



JEUS의 기능과 역할

위 그림에서 나타난 4가지 Layer는 다음과 같다.

- **Client Layer**

  웹 서버나 Java 애플리케이션 또는 Native 애플리케이션으로 구성된다. 최종 사용자는 WAS의 서비스를 사용하기 위해서 다양한 클라이언트를 사용하며, 이 클라이언트는 다양한 프로토콜 중에 하나를 사용해서

WAS의 서비스에 접근한다.

- **Web/Internet Layer**

  클라이언트와 WAS 사이에서 작동하는 웹 서버나 프로토콜로 정의된다. 이 Layer에서는 정적인 콘텐츠와 부하 분산을 처리한다.

- **JEUS/WAS Layer**

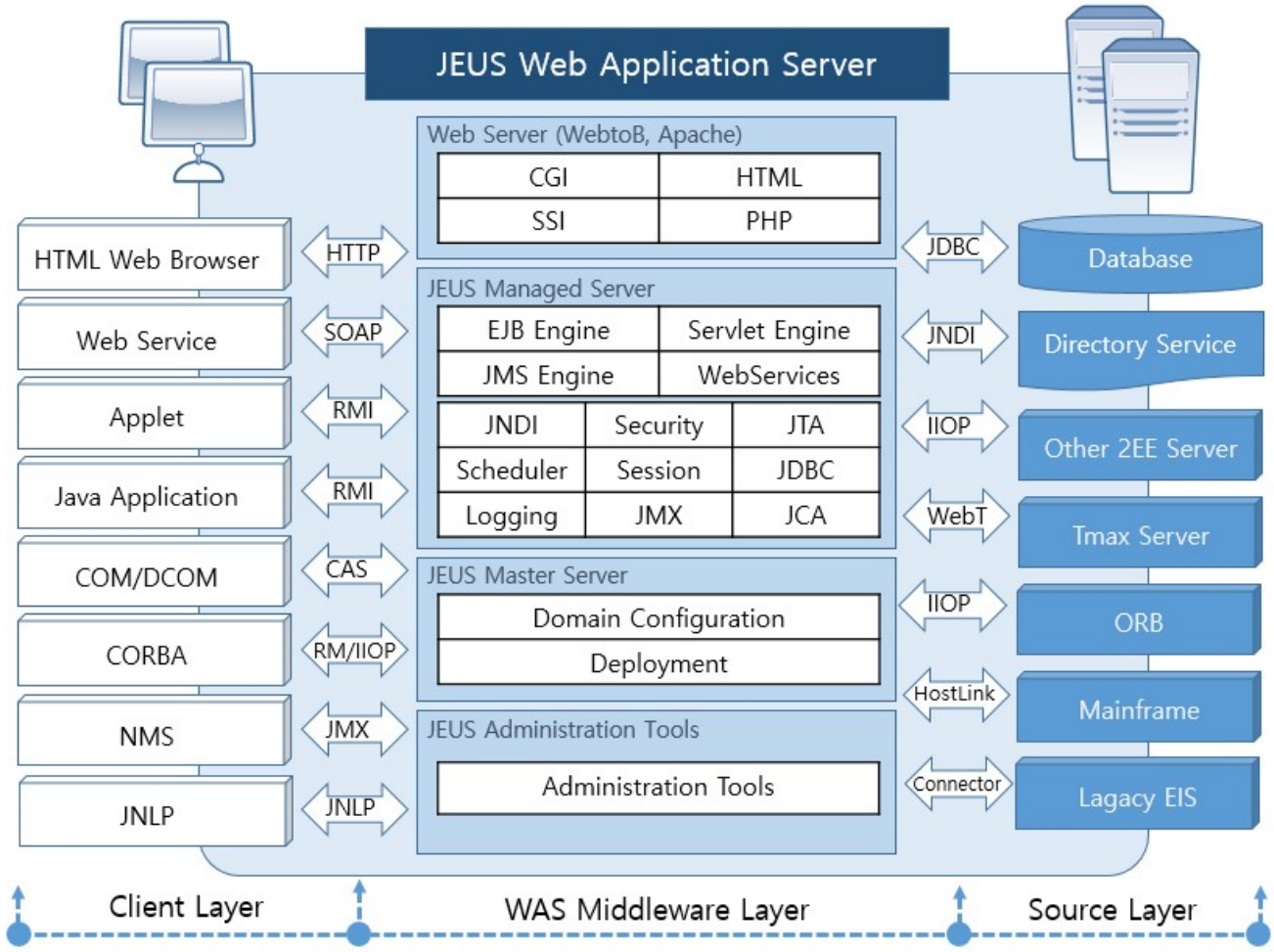
  Java 기반의 미들웨어로 구성되며, Web Layer나 Client Layer로부터 오는 요청을 받아서 처리한다.

- **EIS Layer**

  비즈니스 데이터나 기존의 Legacy 서비스를 나타낸다. WAS는 JDBC나 디렉터리 서비스, Jakarta Connector 등의 다양한 메커니즘을 통해서 Legacy 서비스와 상호 작용한다.

1.3. 구성 요소와 아키텍처

JEUS는 많은 서로 다른 모듈들로 구성되어 있다. 이러한 모듈들은 다음의 JEUS 웹 애플리케이션 아키텍처 다이어그램에서 클라이언트 애플리케이션과 데이터 저장장치, JEUS 사이의 통신 기술에 따라서 사용된다.



JEUS 웹 애플리케이션 아키텍처 다이어그램

Client Layer는 다양한 형태의 클라이언트 애플리케이션과 통신 프로토콜을 보여주고 있다. Source Layer는

다양한 형태의 back-end 데이터 저장장치들이 나열되어 있다. TmaxSoft의 JEUS 제품은 WebtoB와 더불어 그림의 중간 부분인 WAS Middleware Layer를 구성한다. 웹 서버는 클라이언트 애플리케이션과 연결되어 있으며, 웹 서버는 웹 애플리케이션 서버와 밀접하게 통합되어 있다. 또한 웹 게이트웨이(WebT)는 WAS와 TP-Monitor(Tmax Server)를 연계시키며, 마지막으로 MainFrame 게이트웨이(Host-Link)는 MainFrame과 TP-Monitor 사이의 연결을 제공한다.

다음 절에서는 그림에서 보이는 3가지 Layer(Client Layer, WAS Middleware Layer, Source Layer)의 구성 요소에 대해 알아본다.

1.3.1. Client Layer

Client Layer(클라이언트 계층)는 JEUS를 사용할 수 있는 원격 혹은 지역 애플리케이션을 나타낸다.

다음은 Client Layer를 구성하는 유형에 대한 설명이다.

Client Layer	설명
Web browser	가장 일반적인 클라이언트 애플리케이션은 HTML 콘텐츠를 얻기 위해 JEUS 서블릿 엔진과 WebtoB Light 웹 서버에 요청을 하는 표준 웹 브라우저이다. 통신 프로토콜은 HTTP이다.
Web Services	웹 서비스의 구현을 제공한다.
Applet	JEUS 자신의 구성 요소를 참조할 수 있는 애플릿 컨테이너를 제공한다.
Java Application	일반적인 독립된 Java 애플리케이션들은 JEUS에 의해 제공되는 클라이언트 컨테이너 내에서 RMI를 사용하여 실행된다. 이러한 클라이언트들을 Jakarta EE 스펙에서는 애플리케이션 클라이언트라고 한다.
COM/DCOM	Microsoft Windows 환경에서 EJB를 COM 형태로 호출할 수 있다.
CORBA	CORBA 기술을 사용한 애플리케이션도 RMI/IIOP를 통해 JEUS를 사용할 수 있다.
NMS	네트워크 관리 시스템은 JMX를 통해 JEUS를 관리하고 사용할 수 있다.
JNLP	JNLP(Java Network Lanuching Protocol) 클라이언트를 사용할 수 있다.

1.3.2. WAS Middleware Layer

JEUS 웹 애플리케이션 아키텍처 다이어그램에서 JEUS WAS Layer는 JEUS 9 제품을 나타내며 구성 요소는 다음과 같다.

- **JEUS Master Server**

  도메인 내에는 Master Server라는 특별한 하나의 서버가 항상 존재한다. Master Server는 도메인 내 서버들 간의 설정과 도메인 내의 모든 애플리케이션 및 리소스를 중앙에서 관리하고, 서버들을 제어 및 모니터링하는 관리 툴과 통신한다.

서비스	설명
Domain Configuration	도메인 전체의 설정을 관리한다.

서비스	설명
Application Management	도메인 전체의 애플리케이션을 관리한다.
Administration	관리 도구를 통해 도메인 내의 모든 서버, 서비스, 애플리케이션 및 리소스를 제어하거나 상태를 모니터링할 수 있다.

- **JEUS Managed Server(MS)**

  Managed Server(이하 MS)는 JEUS 시스템에서 구성될 수 있는 다양한 형태의 엔진과 서비스에 대한 기반을 제공하고 있다.

엔진 / 서비스	설명
EJB Engine	EJB 비즈니스 애플리케이션을 구동한다.
Servlet Engine	웹 컨테이너로 정적인 콘텐츠(HTML)뿐만 아니라 JSP/Servlet 애플리케이션을 구동한다.
JMS Engine	JMS 기반 구조를 제공한다.
Web Services Engine	JEUS 웹 서버의 인스턴스로서 서블릿 엔진의 front-end로 구동된다.
JNDI Service	Naming 시스템이다.
Security Service	인증과 권한 서비스이다.
JTA	웹 애플리케이션 서버에서 구동되는 다양한 애플리케이션들에 대한 완전한 트랜잭션을 제공한다.
Scheduler	미리 정해진 시간에 이벤트를 발생시키는 타이머 기능을 제공한다.
Session Manager	클러스터링이 필요한 경우 신뢰성있는 방식으로 클라이언트의 세션 정보를 저장한다.
JDBC	데이터베이스 Connection Pool이 설정될 수 있다.
Logging	JEUS 실행 중에 시스템에서 수행되었던 일련의 작업들에 대한 내용을 순서대로 보관 및 기록한다.
JMX	NMS/JMX 클라이언트가 JEUS 시스템을 관리할 수 있도록 한다.
JCA	JCA(Jakarta™ Connector Architecture)는 JCA를 지원하는 EAI(Enterprise Application Integration) 솔루션들에게 JEUS의 JCA를 통한 Legacy EIS 연결을 지원한다.

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

  웹 서버는 HTML과 같은 정적인 콘텐츠와 CGI와 같은 동적인 콘텐츠를 전송한다. 또한, 서블릿 엔진의 front-end로서 상호 작용한다. WebtoB는 TmaxSoft의 WebtoB 웹 서버로서 모든 기능을 지원하는 버전과 일부 축소된 기능만을 가진 JEUS 웹 서버의 2가지 버전이 있다.

  JEUS 웹 서버는 JEUS에 포함되어 있으며, WebtoB에는 포함되어 있지 않다. 또한, 오픈 소스 웹 서버인 Apache를 JEUS에서 사용할 수 있다.

- **JEUS Administration Tools**

JEUS는 다음과 같은 관리 툴을 제공한다.

툴	설명
콘솔 툴(jeusadmin)	명령행 기반의 콘솔 툴을 사용하여 JEUS 서버를 제어할 수 있다.

1.3.3. Source Layer

JEUS 웹 애플리케이션 아키텍처 다이어그램의 오른쪽의 Source Layer는 back-end의 리소스와 JEUS 시스템에 의해 사용될 수 있는 데이터 저장소를 나타내며, 종류는 다음과 같다.

Source Layer	설명
Database	JEUS에서 JDBC를 통해서 접속할 수 있다.
Directory Service	LDAP와 같은 것들이 있으며, JNDI를 통해서 사용된다.
Other Jakarta EE Server	JEUS는 타 벤더의 Jakarta EE 서버와 상호 작용이 가능하다.
Tmax Server	TmaxSoft에서 개발한 TP-Monitor로, WebT API 라이브러리는 JEUS와 Tmax를 통합하는 데 사용된다.
ORB	IIOP(Internet Inter-ORB Protocol)를 통해 참조될 수 있다.
Mainframe	IBM MainFrame들은 특별히 Host-Link(Connector) 제품을 통해 사용된다.
Legacy EIS	JCA를 지원하는 Legacy EIS로 JEUS와 상호 작용이 가능하다.

1.4. 상호 운용 모듈

상호 운용성이란 예상되는 결과를 얻기 위해 2개 또는 그 이상의 시스템(컴퓨터, 통신장치, 네트워크, 소프트웨어 혹은 다른 정보 기술 요소)에서 정의된 방법을 통한 데이터 교환이나 상호 작용을 의미한다(ISO ITC-215). JEUS는 서로 다른 프로토콜과 웹 서비스, JNLP, RMI-IIOP와 같은 기술들을 지원한다.

다음은 JEUS의 상호 운용을 위해 제공되는 모듈이다.

모듈	설명
RMI-IIOP	IIOP(Internet Inter-ORB Protocol) 프로토콜에서 수행되는 RMI 기술로서 JAVA 플랫폼에서 CORBA의 분산 컴퓨팅 작업을 가능하게 한다.
JEUS	다른 웹 애플리케이션의 사용을 가능하게 하며, 웹 서비스 또한 지원한다.
WebT	TP-Monitor와 JEUS를 연계하는 게이트웨이이다.
Host-Link	Legacy EIS에 있는 서비스를 클라이언트가 사용할 수 있도록 하는 어댑터 모듈이다.
JCA	JEUS와 JEUS 클라이언트가 가상적으로 어떠한 Legacy의 EIS에 대한 상호 작용도 가능하게 한다.

1.5. Edition

다음은 JEUS 9의 Edition에 대한 설명이다.

Edition	주요 특징
JEUS Standard Edition	◦ Jakarta™ EE 9 지원 ◦ JEUS Server ◦ JEUS Web Server ◦ Java Database Connectivity(JDBC) 지원 ◦ Java Naming and Directory Interface(JNDI) 지원 ◦ Java Management Extensions(JMX) 지원 ◦ 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. 제공되는 JEUS 안내서는 위의 표에서 언급된 모든 주제들을 포함하고 있다. 안내서에 기술된 기능들이 실제로 사용 가능한지 여부를 이 표에서 확인하기 바란다. JEUS를 설치하면 Trial 라이선스가 기본적으로 내장되어 있다.
2. 각각의 Standard Edition, Enterprise Edition별로 Cloud 라이선스를 제공한다.

2. JEUS 환경

본 장에서는 JEUS에서 사용하는 관리 툴과 디렉터리 구조와 환경변수에 대해서 설명한다. 그리고 JEUS와 관련 XML 설정 파일의 전반적인 내용에 대해서 소개한다.

2.1. 관리 툴

다음은 JEUS에 접속해서 사용하는 툴에 대한 설명이다.

툴	설명
startMasterServer	JEUS Master Server를 실행하는 가장 기본적인 툴이다. 자세한 내용은 "JEUS Reference 안내서"를 참고한다.
startManagedServer	JEUS Managed Server를 실행하는 가장 기본적인 툴이다. 자세한 내용은 "JEUS Reference 안내서"를 참고한다.
jeusadmin	콘솔 툴은 command prompt에서 JEUS를 관리하는 데 사용된다. 자세한 내용은 JEUS Reference 안내서의 "jeusadmin"을 참고한다.

위에서 언급한 것 이외에 Master Server와 Managed Server를 다운시키는 명령어가 존재한다. 자세한 내용은 "JEUS Reference 안내서"를 참고한다.

2.2. 디렉터리 구조

다음은 JEUS를 설치했을 때의 전체 디렉터리 구조이다.

```
{JEUS_HOME}
    |-- bin
    |    |--[01]startMasterServer
    |    |--[01]startManagedServer
    |    |--[01]stopServer
    |    |--[01]jeusadmin
    |--derby
    |--docs
    |--lib
    |   |--shared
    |        |--[X]libraries.xml
    |--license
    |--nodemanager
    |--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
                             |--nodemanager

* 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
```

다음은 디렉터리와 파일의 설명이다.

{JEUS_HOME}

JEUS의 최상위 디렉터리로 실제 디렉터리 이름과 위치는 설치할 때 결정된다.

bin

서버의 시작 및 종료 스크립트인 startMasterServer, startManagedServer, stopServer와 JEUS 콘솔 툴(jeusadmin)과 같은 실행 파일들이 위치한다.

derby

샘플 애플리케이션이나 테스트에서 쉽게 사용할 수 있도록 Apache Derby를 포함시킨다.

docs

JEUS에서 제공하는 API에 대한 Javadoc이 존재한다.

lib

JEUS가 기동하는 데 필요한 라이브러리가 존재한다. shared 디렉터리를 제외한 나머지 디렉터리들은 사용자가 접근할 필요가 없다.

디렉터리	설명
shared	shared 디렉터리에는 애플리케이션에서 사용하는 라이브러리가 존재한다. shared 디렉터리의 라이브러리를 사용하려면 libraries.xml에 라이브러리의 정보를 추가해야 한다. 그리고 해당 라이브러리를 사용할 애플리케이션의 JEUS Deployment Descriptor(DD)에서 해당 라이브러리에 대한 레퍼런스 정보를 지정해야 한다. shared 라이브러리에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 "공유 라이브러리"를 참고한다.

license

JEUS 라이선스 파일이 위치한다. 라이선스 파일은 JEUS가 실행되기 위해서 반드시 필요한 파일이다.

nodemanager

JEUS 노드 매니저를 위한 설정 파일인 jeusnm.xml 파일이 위치한다.

setup

JEUS 설치 후 사용할 수 있도록 환경을 구축하기 위해 필요한 파일들이 위치한다.

templates

각종 설정과 환경 등의 template 파일이 위치한다.

samples

JEUS의 예제 파일들이 위치한다.

webserver

JEUS가 설치될 때 JEUS 웹 서버가 설치되는 디렉터리이다. 자세한 내용은 "JEUS Web Engine 안내서"를 참조한다.

domains

하위에 도메인별로 DOMAIN_HOME과 JEUS_HOME에서 사용하는 노드 정보가 포함된 nodes.xml이 존재한다.

다음의 디렉터리 및 파일들은 DOMAIN_HOME 아래에 위치한다.

- .applications

  해당 도메인에서 관리하는 애플리케이션 파일이 존재한다.

  install-application, **uninstall-application** 명령어를 통해 추가 및 삭제가 가능하다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다. 각 명령어에 대한 설명은 JEUS Reference 안내서의 "install-application"과 "uninstall-application"을 참고한다.

- .deploymentplans

  해당 도메인에서 관리하는 Deployment Plan 파일이 존재한다.

  install-deployment-plan, **uninstall-deployment-plan** 명령어를 통해 추가 및 삭제가 가능하다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다. 각 명령어에 대한 설명은 JEUS Reference 안내서의 "install-deployment-plan"과 "uninstall-deployment-plan"을 참고한다.

◦ .libraries

해당 도메인에서 관리하는 라이브러리 파일이 존재한다.

install-library, **uninstall-library** 명령어를 통해 추가 및 삭제가 가능하다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다. 각 명령어에 대한 설명은 JEUS Reference 안내서의 "install-library"와 "uninstall-library"를 참고한다.

◦ bin

해당 도메인에 속한 Master Server와 Managed Server의 시작 및 종료 스크립트가 위치한다. JEUS_HOME/bin의 startMasterServer, startManagedServer, stopServer와 동일한 기능을 수행하지만 도메인 이름을 설정할 필요가 없다.

◦ config

도메인의 설정 파일인 domain.xml이 변경된 경우 이전 이력을 위해 존재하는 백업 파일들이 위치한다. 도메인 설정에 대한 자세한 설명은 JEUS Domain 안내서의 "도메인 설정변경"을 참고한다.

구분	설명
security	◦ SYSTEM_DOMAIN : 도메인 단위로 적용되는 보안 도메인 파일인 accounts.xml, policies.xml이 존재하며, 각 XML 파일은 jeusadmin을 통해 동적 설정 변경이 가능하다. 보안 도메인 설정에 대한 자세한 설명은 JEUS Security 안내서의 "보안 도메인 설정"을 참고한다. ◦ security-domains.xml : JEUS의 보안 도메인에 대한 설정을 저장하는 파일이다. ◦ security.key : 대칭키 암호화 알고리듬에 대한 Key를 저장하는 파일로 JEUS_HOME/bin/encryption을 수행하면 생성된다. security.key 파일에 대한 자세한 설명은 JEUS Security 안내서의 "패스워드 보안 설정"을 참고한다. ◦ policy : Java permission 설정 파일이다. JEUS의 보안 시스템과는 별도로 Java SE Security Manager에서 사용된다.
servlet	◦ web.xml : web.xml을 개별적으로 가지고 있지 않은 경우 웹 엔진이 사용할 웹 모듈의 web.xml이다. 기본값은 빈 XML 파일이다. ◦ webcommon.xml : 도메인 내 서버의 웹 엔진의 모든 웹 모듈에 적용되는 공통 설정 파일이다. 설정에 대한 자세한 설명은 JEUS Web Engine 안내서의 "디렉터리 구조"를 참고한다.
webadmin	◦ webadmin 폴더는 webadmin 설정을 저장하는 폴더이다. ◦ rolewithusers.json : 사용자와 webadmin 페이지 간 권한 관계를 저장하는 파일이다.

◦ lib/application

도메인 전체에 적용하고 싶은 애플리케이션 라이브러리를 위치시키는 디렉터리다.

SERVER_HOME에 존재하는 애플리케이션 라이브러리와 충돌이 발생할 경우 SERVER_HOME/lib/application이 우선되고 경고 메시지가 남는다. lib/application 디렉터리에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 "lib/application 디렉터리"를 참고한다.

◦ servers

이 디렉터리 하위에 SERVER_HOME 디렉터리가 서버 이름으로 생성된다. SERVER_HOME 디렉터리 구조에 대한 자세한 설명은 JEUS Server 안내서의 "서버 디렉터리 구조"를 참고한다.

디렉터리	설명
.workspace	JEUS가 사용하는 서버별 공간으로 사용자가 변경해서는 안 된다.
bin	서버의 시작/종료 스크립트를 포함하고 있다. JEUS_HOME/bin의 스크립트와 동일한 기능을 수행하지만 도메인 이름과 서버 이름을 설정할 필요가 없다. ◦ Master Server일 경우 : startMasterServer/stopServer가 존재한다. ◦ Managed Server일 경우 : startManagedServer/stopserver가 존재한다.
lib/application	서버에 적용하고 싶은 애플리케이션 라이브러리가 존재한다. 도메인 범위의 라이브러리(DOMAIN_HOME/lib/application)보다 우선순위가 높다. 라이브러리가 충돌할 경우 이 디렉터리에 존재하는 파일이 적용되며 경고 메시지가 남는다. lib/application에 대한 자세한 설명은 JEUS Applications & Deployment 안내서의 "lib/application 디렉터리"를 참고한다.
logs	서버의 Launcher 로그, 서버 로그, 엑세스 로그 파일이 남는다. 자세한 내용은 JEUS Server 안내서의 "Logging"을 참고한다.
nodemanager	노드 매니저가 재기동할 때 관리하고 있던 서버였는지를 판단하기 위한 정보를 저장하는 디렉터리이다. JEUS가 사용하는 디렉터리로 사용자의 접근을 제한한다.

2.3. 환경변수

환경변수는 모두 JEUS_HOM\bin\jeus.properties에 설정되어 있으며, JEUS_HOME\bin 디렉터리의 모든 스크립트에서 사용된다.

다음은 JEUS에서 사용하는 환경변수이다.

환경변수	내용
JEUS_HOME	JEUS가 설치된 홈 디렉터리로 필수 사항이다. (예: JEUS_HOME=/home/jeus/jeus9)
JAVA_HOME	JDK의 홈 디렉터리이다. (예: JAVA_HOME=/usr/jdk17)

해당 변수는 필요한 경우 수정해서 사용한다. 단, XML 설정 파일에서는 이들 환경변수를 사용할 수 없다. 모든 환경변수는 설치할 때 기본값으로 설정된다. 대부분의 경우 설정된 값을 그대로 사용하면 된다.

환경변수를 변경하는 방법은 OS에 따라 다르므로, 이에 대해서는 각 OS 안내서를 참고한다.

2.4. 환경설정 파일

JEUS는 환경설정을 위해서 각각 고유의 XML 포맷을 사용하며, 직접 수정하거나 툴을 사용해서 수정할 수 있다.

다음은 JEUS의 XML 설정 파일과 내용, 위치를 정리한 내용이다.

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

위치	JEUS_HOME/domains/<domain_name>/config/
목적	JEUS Manager와 노드를 관리하는 기본 설정 파일이다.
참고 안내서	"JEUS Domain 안내서", "JEUS Server 안내서"

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

위치	웹 애플리케이션 Archive의 WEB-INF
설명	JEUS 웹 애플리케이션(Servlet app) DD(Deployment Descriptors) 파일이다.
참고 안내서	"JEUS Web Engine 안내서"

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

위치	EJB 애플리케이션 Archive의 META-INF
설명	JEUS EJB Module DD(Deployment Descriptors) 파일이다.
참고 안내서	"JEUS EJB 안내서"

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

위치	클라이언트 애플리케이션 Archive의 WEB-INF
설명	애플리케이션 클라이언트 DD(Deployment Descriptors) 파일이다.
참고 안내서	"JEUS Application Client 안내서"

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

위치	리소스 어댑터 Archive의 META-INF
설명	리소스 어댑터 DD(Deployment Descriptors) 파일이다.
참고 안내서	"JEUS JCA 안내서"

- policies.xml (policies.xsd)

위치	JEUS_HOME/domains/<*domainname*>/config/security
설명	JEUS Security 정책을 설정한 파일이다.
참고 안내서	"JEUS Security 안내서"

- accounts.xml (accounts.xsd)

위치	JEUS_HOME/domains/<*domainname*>/config/security
설명	JEUS Security 계정을 설정한 파일이다.
참고 안내서	"JEUS Security 안내서"

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

위치	Webservice client archive의 META-INF
설명	웹 서비스 클라이언트 정보를 설정한 파일이다.
참고 안내서	"JEUS Web Service 안내서"

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

위치	Webservice client archive의 META-INF
설명	웹 서비스 클라이언트 Ant Task에서 사용하는 설정 파일이다.
참고 안내서	"JEUS Web Service 안내서"

1. Jakarta EE의 표준 Descriptor 파일인 web.xml이나 ejb-jar.xml 파일도 사용된다. 각 파일은 해당 Jakarta EE 스펙을 참고한다.
2. 모든 XML Schema 파일은 JEUS_HOME/lib/schemas/jeus/supportLocale/ko에 위치한다.

3. JEUS 안내서 구성

본 장에서는 안내서를 어떻게 활용하면 좋은지와 안내서에서 사용된 약자들을 소개한다. JEUS 안내서에서 필요한 정보를 얻기 위해서는 본 장을 주의 깊게 살펴보길 바란다.

3.1. 개요

JEUS 안내서는 다음과 같은 3가지 방법으로 구할 수 있다.

- 소프트웨어와 함께 제공되는 HTML 문서
- JEUS Installer CD에서 제공되는 PDF 파일

  PDF 파일을 보려면 Adobe Acrobat Reader 또는 다른 PDF를 지원하는 소프트웨어가 필요하다. Adobe Reader 다운로드 페이지에서 Adobe Acrobat Reader를 다운로드 받을 수 있다.

- TmaxSoft의 TechNet 사이트에서 업데이트된 소프트웨어나 안내서

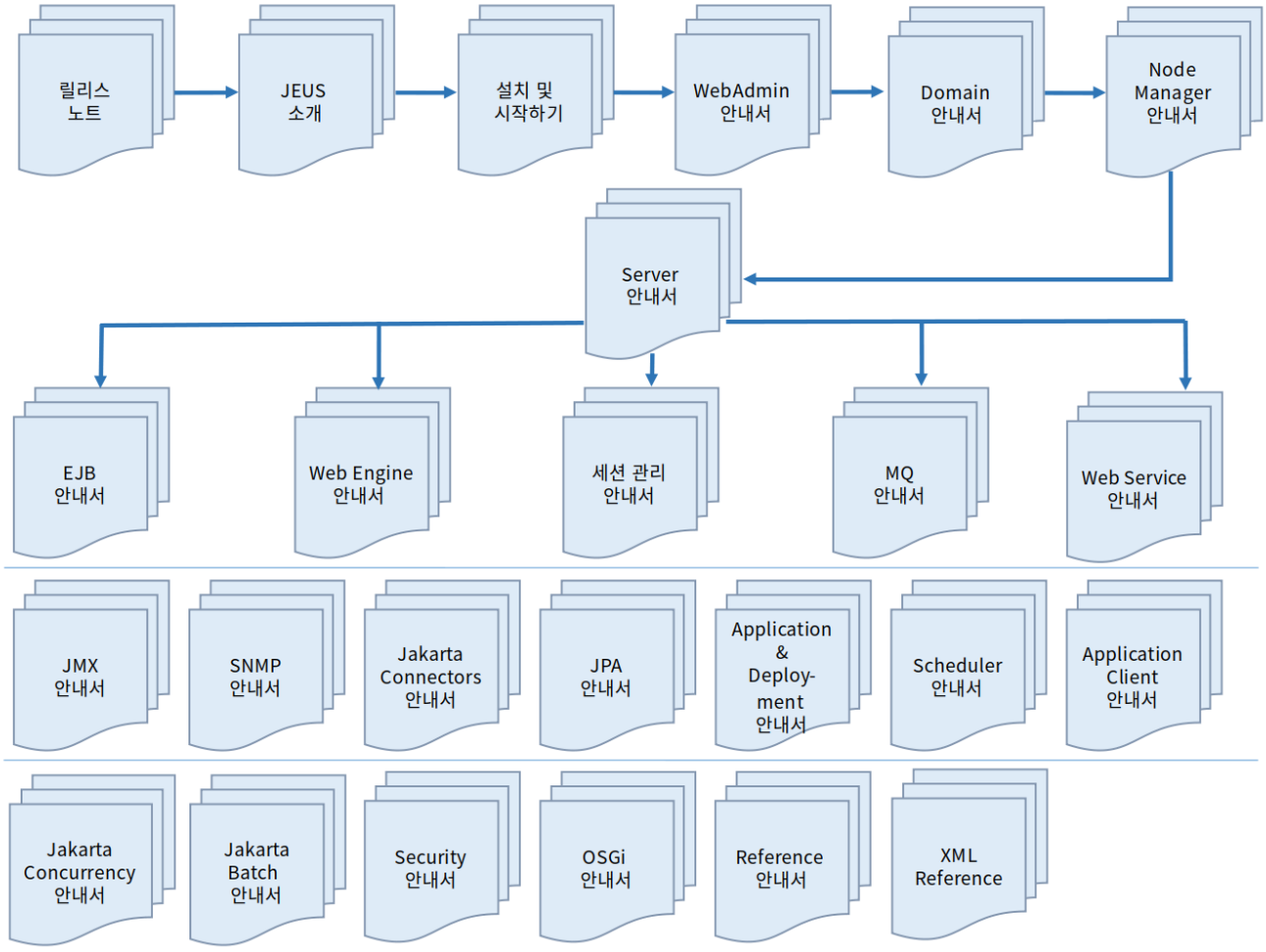
다음은 JEUS 안내서에 대해 기본적으로 알아야 하는 4가지 사항이다.

- JEUS 안내서는 Java와 Jakarta EE 기술에 대한 경험을 가진 Jakarta EE 전문가를 위해 작성되었다.
- 안내서는 개별적이고 연관성을 배제한 문서들로 구성되어 있다.
- JEUS 안내서 문서는 몇 가지의 예외를 제외하고는 기본적으로 표준적인 문서 양식에 따라 구성되어 있다.
- 안내서는 매우 다양한 방식으로 구성되어 있다.

안내서의 내용을 이해하기 위해서는 Java와 Jakarta EE 기술에 대한 지식이 필요하다. 그러한 지식은 서점에 있는 다양한 서적들로부터 얻길 바란다. 또한, Java 관련 웹 사이트인 Oracle Java 기술 문서에서 유용한 온라인 문서와 스펙, 자습서 등을 얻을 수 있다. JEUS 안내서에서 Jakarta EE 스펙에 서술되어 있는 모든 내용에 대해 다루는 것은 어렵다. 현재는 단지 JEUS에 특화된 정보들에 대해서만 다룬다.

3.2. 안내서의 구성

다음은 JEUS 안내서의 구성을 나타낸다. 화살표는 JEUS에 익숙하지 못한 사용자를 위해 읽어야 할 우선순위를 나타낸 것이다. JEUS 안내서는 총 25권으로 구성되어 있다.



JEUS 안내서 구성

각각의 안내서에 대한 내용들은 다음의 목록에 명시되어 있다. 특정한 주제에 대해서 위치를 빨리 찾고 싶다면 다음의 목록을 참고한다.

- **"릴리스 노트"**

  릴리즈 되는 제품의 새로운 기능이나 이전 버전의 업그레이드하는 방법을 설명한다.

  - JEUS의 새로운 기능
  - 이전 버전의 업그레이드 방법

- **"JEUS 소개"**

  JEUS에 대한 전반적인 소개와 JEUS의 아키텍처 및 각 구성 요소들에 대해 설명한다.

  - JEUS 서버에 대한 소개
  - JEUS 환경
  - JEUS 안내서의 구성

- **"설치 및 시작하기"**

  JEUS의 설치하는 방법과 각 유형별 사용하는 예제를 설명한다.

  - UNIX에서 Java 설치

- UNIX에서 JEUS 설치
- 예제 애플리케이션에 대한 설명 포함(Getting Started)
- JEUS 시스템 입문서
- EJB 입문서
- Servlet/JSP 입문서
- 웹 서비스 입문서

- **"WebAdmin 안내서"**

  JEUS의 웹 관리 툴로서 일반 사용자부터 관리자까지 반드시 읽어야 한다.

  - WebAdmin의 화면 구성 및 주요 기능
  - JEUS의 설정 및 제어, 모니터링

- **"Domain 안내서"**

  JEUS 도메인의 구조, 구성에 대해서 전반적으로 설명한다. 따라서 JEUS에 대한 이해를 필요로 하는 대부분의 사람들이 반드시 읽어야 한다.

  - 도메인의 개념과 구성 요소
  - 도메인 생성 방법과 디렉터리 구조
  - 도메인 설정 변경및 반영 방법
  - 도메인을 구성하고 있는 서버의 제어 방법과 그에 대한 상태 변경
  - 도메인을 구성할 수 있는 클러스터 개념
  - 도메인을 관리하는 서버가 비정상 종료될 경우의 문제점과 극복 방안
  - 도메인과 관련된 보안 사항

- **"Node Manager 안내서"**

  JEUS 노드 매니저의 구조와 구성에 대해서 전반적으로 다룬다. 따라서 JEUS에 대한 이해를 필요로 하는 대부분의 사람들이 반드시 읽어야 한다.

  - 노드 매니저의 개념, 목적, 종류 등의 기본적인 사항
  - Java 타입 노드 매니저를 사용하기 위한 설정 방법
  - Java 타입 노드 매니저를 사용하기 위한 동작 방식
  - SSH 타입 노드 매니저의 설정/삭제
  - SSH 타입 노드 매니저의 사용 방법

- **"Server 안내서"**

  JEUS 관리 차원의 주요 안내서이며, JEUS 시스템 관리자들이 가장 많이 읽어야 하는 부분이다.

  - JEUS의 구성 요소 및 구성 요소가 제공하는 서비스 개요
  - JEUS 구성 요소들의 설정 방법
  - JEUS 서버의 제어 및 모니터링 방법

- JEUS JNDI의 기본 사항과 애플리케이션 개발 방법
- JEUS와 연동하여 사용이 가능한 외부 리소스의 종류와 설정 방법
- JEUS에서 제공하는 Connection Pool 및 부가 기능과 사용 방법
- JEUS의 트랜잭션 매니저와 그 주변 요소
- JEUS의 Logging 시스템
- 주요 JDBC 드라이버에 대한 Connection Pool 설정 예제

"Server 안내서"는 JEUS Security, Naming 그리고 트랜잭션 매니저와 같은 서로 다른 많은 주제들을 포함하기 때문에 그 범위가 매우 방대하다. 이러한 구성의 이유는 비록 주제들이 전혀 다른 것이라 할지라도 이러한 구성 요소들이 모두 동일한 환경설정 파일에서 구성되고 동일한 JVM에서 수행되기 때문이다. 또한 각각의 안내서들의 수를 최소한으로 유지하기 위해서이다.

- **"EJB 안내서"**

  JEUS EJB 엔진과 EJB 모듈을 deploy하는 것에 대해 주로 설명한다.

  - JEUS EJB에 대한 개요
  - JEUS EJB 엔진의 설정, 제어 및 모니터링과 튜닝 방법
  - EJB 모듈의 관리, 조립, deploy와 제어 및 모니터링 방법
  - 일반적인 EJB(각각의 Bean들에 대한 구성)의 공통 특성
  - JEUS EJB에 대한 보안
  - JEUS EJB에 대한 보안의 상호작용
  - JEUS EJB 클러스터링
  - Session EJB
  - Entity EJB
  - Message Driven Beans
  - EJB 클라이언트
  - JEUS EJB를 위한 Ant 사용
- **"Web Engine 안내서"**

  JEUS 웹 엔진의 관리를 위한 안내서이며 Jakarta EE WAR Archive와 Servlet/JSP의 관리와 deploy하는 방법에 대해 설명한다.

  - JEUS 웹 엔진에 대한 기본 개념 및 환경설정
  - JEUS 웹 커넥션의 관리
  - JEUS 웹 컨텍스트(웹 애플리케이션/WAR 파일들)의 구조, deploy, 제어 및 모니터링과 튜닝 방법
  - JEUS 웹 엔진의 기능 및 설정 방법
  - 웹 서버와의 연결과 클러스터링(WebtoB, Apache and built-in HTTP server connections and

clusters)
    - 가상 호스팅
    - JEUS WebCache의 개념과 사용 방법
    - Reverse Proxy의 개념과 사용 방법
    - 클래스 동적 반영의 기본 설정 및 동작
- **"세션 관리 안내서"**

  JEUS 웹 엔진, EJB 엔진 등에서 사용되는 세션 매니저(Session Manager), 세션 서버(Session Server)의 구성과 그 설정 등에 대한 설명을 다루고 있다. 주로 클러스터링 환경 내에서, 또는 단일 서버 내에서 세션의 유지, 공유 등을 관리할 시스템 관리자와 개발자들을 대상으로 한다.

    - 세션 트래킹(Session Tracking)의 구조, 동작, 설정 및 튜닝 방법
    - 클러스터링 환경에서 세션 트래킹을 위해 운용되는 분산 세션 서버의 구조, 동작 및 설정 방법
- **"MQ 안내서"**

  JEUS 메시지 기반 시스템(JMS)을 설명한다.

    - JEUS JMS의 개요
    - JMS 엔진에 대한 환경설정, 모니터링과 제어, 장애 발생 시 복구 방법
    - JEUS에서의 JMS 프로그래밍
    - JEUS MQ의 클러스터링 종류와 사용 방법
    - JEUS MQ의 특수 기능
- **"Web Service 안내서"**

  JEUS 내의 웹 서비스에 대해 설명한다.

    - JEUS 웹 서비스에 대한 개요
    - 웹 서비스 back-end 생성
    - 웹 서비스의 호출
    - 웹 서비스 back-end를 사용하는 클라이언트 구현하기
    - 데이터 타입과 JEUS 웹 서비스
    - 웹 서비스에 관련된 Ant
    - 표준 바인딩 선언 및 사용자화
    - 핸들러 프레임워크
    - 프로바이더와 디스패치 인터페이스
    - 비동기 웹 서비스
    - MIME Attachment 메시지 전송 및 Fast Infoset을 이용한 웹 서비스
    - 웹 서비스 정책, Addressing, 신뢰성 메시지 기술 및 트랜잭션에 대한 설명
    - UDDI 이용

    - 웹 서비스 보안
    - 웹 서비스의 XML
- **"JMX 안내서"**

  JMX를 사용해서 JEUS를 관리하기 위한 안내서이다.

    - JEUS JMX Manager 설정
    - JMX 애플리케이션 개발
    - JMX API 레퍼런스
- **"SNMP 안내서"**

  산업 표준인 SNMP 프로토콜을 이용한 JEUS 모니터링에 대해 설명한다.

    - JEUS SNMP Agent의 개요
    - SNMP Agent의 구성
    - JEUS SNMP 프로그래밍
    - JEUS SNMP MIB
- **"Jakarta Connectors 안내서"**

  JEUS와 Legacy 시스템과 연결하기 위한 커넥터에 대한 설명한다.

    - 커넥터에 대한 내용
    - 커넥터 패키징
    - 커넥터 사용과 튜닝
- **"JPA 안내서"**

  JEUS에 통합된 TopLink Essential을 JEUS에서 사용하는 데 필요한 설정에 대해서 설명한다.

    - Jakarta Persistence API 소개
    - 프로바이더 설정
    - JEUS 설정
- **"Scheduler 안내서"**

  JEUS의 Scheduler 기능에 대한 안내서이다.

    - Scheduler 서비스 설정
    - Scheduler 서비스 프로그래밍
- **"Applications & Deployment 안내서"**

  Jakarta EE 애플리케이션을 JEUS에 deploy하기 위한 여러 가지 방법과 툴에 대해 설명한다.

    - 도메인 환경에서의 애플리케이션 관리 방법
    - Graceful Undeployment와 Redeployment

- 모듈과 애플리케이션, 공유 라이브러리에 대한 설명
- Jakarta EE 애플리케이션 파일의 작성과 deploy 방법

- **"Application Client 안내서"**

  Jakarta EE 클라이언트와 JEUS 사이의 상호 운용에 대해 설명한다.

  - Jakarta EE 애플리케이션 클라이언트
  - 애플릿 클라이언트
  - JNLP 클라이언트

- **"Security 안내서"**

  JEUS에서 보안 시스템의 설정, 운영방법과 보완관련 프로그래밍에 대해 설명한다.

  - 보안 시스템의 개요와 설정
  - 애플리케이션과 모듈의 보안 설정
  - 보안 시스템의 API를 이용한 프로그래밍
  - Custom 보안 서비스 개발
  - JACC Provider의 사용
  - JAAS의 사용 방법

- **"Jakarta Concurrency 안내서"**

  JEUS에서 Jakarta Concurrency 프로그래밍에 대해 설명한다.

  - Jakarta Concurrency의 개요와 설정
  - Jakarta Concurrency의 API를 이용한 프로그래밍

- **"Jakarta Batch 안내서"**

  JEUS에서 Jakarta Batch 프로그래밍에 대해 설명한다.

  - Jakarta Batch 의 개요와 설정
  - Jakarta Batch를 이용한 프로그래밍

- **"OSGi 안내서"**

  JEUS에서 제공하는 OSGi 기능에 대해 설명한다.

  - OSGi 관련 기능에 대한 개요와 설정 방법
  - 웹 애플리케이션에서 OSGi 관련 기능을 사용하는 방법

- **"Reference 안내서"**

  JEUS를 사용하는 데 도움이 되는 Reference를 모아둔 안내서이다.

  - 시스템 프로퍼티 내용
  - 콘솔 명령어 사용법

- Ant Task 사용법
- JEUS에서 사용되는 API 모음