

JEUS インストール & スタートガイド

JEUS v8.0



Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2016 TmaxSoft Co., Ltd. All Rights Reserved.

45, Jeongjail-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13613, South Korea

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.

このソフトウェア(JEUS®)マニュアルの内容とプログラムは、日本国の著作権法および国際条約によって保護されています。マニュアルの内容とプログラムは、TmaxSoft Co., Ltd.との使用許諾契約書の下でのみ使用することができ、マニュアルは使用許諾契約で許可されている範囲を除いては、配布または複製することができません。TmaxSoftの書面による事前の承諾を得ることなく、このマニュアルの全部または一部を電子的または機械的な方法を問わず、転送、複製、配布したり、または二次的著作物を作成する等の行為を一切禁じます。

このソフトウェアのマニュアルとプログラムの使用許諾契約は、いかなる場合においても、マニュアル及びプログラムと関連する知的財産権(登録の有無を問わず)を譲渡するものと解釈されず、TmaxSoftのブランド、ロゴ、商標等の使用権限を与えるものではありません。マニュアルは、情報を提供する目的でのみ提供しており、これに伴う契約上の直接的ないしは間接的な責任を負わず、マニュアルの内容は法律上もしくは商業的な特定の条件が満たされることを保証しません。マニュアルの内容は、製品のアップグレード及び修正により、その内容が予告なく変更されることがあり、内容上の誤りがないことを保証しません。

Trademarks

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

JEUS®は、TmaxSoft Co., Ltd.の登録商標です。

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

Java、Solarisは、Oracle Corporation及びその子会社、関連会社の登録商標です。

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

Microsoft、Windows、Windows NTは、Microsoft Corporationの登録商標または商標です。

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

HP-UXは、Hewlett Packard Enterprise Companyの登録商標です。

AIX is a registered trademark of International Business Machines Corporation.

AIXは、International Business Machines Corporationの登録商標です。

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

UNIXは、X/Open Company, Ltd.の登録商標です。

Linux is a registered trademark of Linus Torvalds.

Linuxは、Linus Torvaldsの登録商標です。

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

その他、記載されている会社名、製品名などは、各社の商号、商標または登録商標です。

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

本マニュアルに記載されている会社名、システム名、製品名などには必ずしも商標表示(TM、®)を付記しておりません。

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}/lib/licenses

この製品の一部ファイルまたはモジュールは、APACHE2.0、CDDL1.0、EDL1.0、OPEN SYMPHONY SOFTWARE1.1、TRILEAD-SSH2、Bouncy Castle、BSD、MIT、SIL OPEN FONT1.1のライセンスに準拠します。

文書情報

文書名: JEUS インストール&スタートガイド

発行日: 2016年10月14日

ソフトウェアバージョン: JEUS v8.0

ガイドバージョン: v2.1.1

目次

このガイドについて	xvii
I. JEUS 紹介ガイド	1
第1章 紹介	3
1.1. 概要	3
1.2. システムの概念と役割	7
1.3. 構成要素とアーキテクチャー	8
1.3.1. Client Layer	8
1.3.2. WAS Middleware Layer	9
1.3.3. Source Layer	11
1.4. 相互運用モジュール	11
1.5. エディション	12
第2章 JEUSの環境	15
2.1. 管理ツール	15
2.2. ディレクトリー構造	81
2.3. 環境変数	20
2.4. 環境設定ファイル	21
第3章 JEUSガイドの構成	23
3.1. 概要	23
3.2. ガイドの構成	24
II. JEUS インストールガイド	33
第4章 インストールの概要および準備事項	35
4.1. 概要	35
4.2. システム要件	35
4.3. インストール手順	36
4.4. JDKのインストール	37
4.5. JEUSのライセンス・ポリシー	38
第5章 UNIX環境でのインストールおよびアンインストール	39
5.1. 概要	39
5.2. インストール	39
5.2.1. GUIモードでのインストール	39
5.2.2. コンソール・モードでのインストール	50
5.3. インストールの確認	57
5.3.1. ディレクトリー構造	81
5.3.2. 環境設定	61
5.3.3. 起動の確認	62
5.4. アンインストール	69
5.4.1. GUIモードでのアンインストール	69
5.4.2. コンソール・モードでのアンインストール	71

5.5.	アンインストールの確認	72
第6章	Windows環境でのインストールおよびアンインストール	73
6.1.	概要	73
6.2.	インストール	73
6.3.	インストールの確認	81
6.3.1.	ディレクトリー構造	81
6.3.2.	環境設定	86
6.3.3.	起動の確認	87
6.4.	サービスの登録と削除	92
6.4.1.	サービスの登録	92
6.4.2.	サービス登録の確認	92
6.4.3.	サービスの削除	93
6.5.	アンインストール	93
6.6.	アンインストールの確認	95
III.	JEUS スタートガイド	97
第7章	概要	99
第8章	システムの設定	101
8.1.	概要	101
8.2.	基本環境の構成	101
8.3.	MSの追加と設定	104
8.4.	データソースの追加	113
8.5.	WebAdminを使用したサーバーの起動および終了	116
第9章	Web層の使用	117
9.1.	例	117
9.2.	コンパイル	120
9.3.	デプロイ	120
9.4.	実行および結果	123
第10章	EJBの使用	127
10.1.	セッションBeanの例	127
10.1.1.	例	127
10.1.2.	コンパイル	129
10.1.3.	デプロイ	129
10.1.4.	実行および結果	132
10.2.	Java Persistence APIの例	133
10.2.1.	例	133
10.2.2.	コンパイル	139
10.2.3.	デプロイ	139
10.2.4.	実行および結果	143
第11章	Webサービスの使用	145
11.1.	Webサービスの生成	145

11.1.1. From Java方式	145
11.1.2. From WSDL方式	147
11.2. Webサービス・クライアントの作成	150
11.2.1. Java SEクライアントの作成	150
付録 A. 各プラットフォームのJDKインストール	153
A.1. Sun Solaris	153
A.2. IBM AIX	153
A.3. HP-UX	154
A.4. Linux	154
付録 B. IPv6の設定	155
B.1. 紹介	155
B.2. IPv6の環境設定	156
付録 C. Windowsサービスの設定	159

図目次

[図 1.1]	JEUSの機能と役割	7
[図 1.2]	JEUSのWebアプリケーション・アーキテクチャーのダイアグラム	8
[図 2.1]	JEUSインストール後のディレクトリー構造	82
[図 3.1]	JEUSガイドの構成	24
[図 4.1]	JEUSのインストール手順	37
[図 5.1]	UNIX環境のGUIモードでのインストール - インストール・ツールの初期画面	40
[図 5.2]	UNIX環境のGUIモードでのインストール - Introduction	41
[図 5.3]	UNIX環境のGUIモードでのインストール - License Agreement	41
[図 5.4]	UNIX環境のGUIモードでのインストール - Choose Platform	42
[図 5.5]	UNIX環境のGUIモードでのインストール - Installation Folder	42
[図 5.6]	UNIX環境のGUIモードでのインストール - Installation Type	43
[図 5.7]	UNIX環境のGUIモードでのインストール - Installation Mode	43
[図 5.8]	UNIX環境のGUIモードでのインストール - Choose JAVA Folder	44
[図 5.9]	UNIX環境のGUIモードでのインストール - Choose Password	45
[図 5.10]	UNIX環境のGUIモードでのインストール - Domain Creation	47
[図 5.11]	UNIX環境のGUIモードでのインストール - ノード・マネージャの設定	48
[図 5.12]	UNIX環境のGUIモードでのインストール - Pre Installation Summary	48
[図 5.13]	UNIX環境のGUIモードでのインストール - Installing JEUS	49
[図 5.14]	UNIX環境のGUIモードでのインストール - Install Complete	49
[図 5.15]	JEUSインストール後のディレクトリー構造	82
[図 5.16]	WebAdminのログイン画面	90
[図 5.17]	WebAdminのメイン画面	91
[図 5.18]	サーバー・リストの照会画面	68
[図 5.19]	UNIX環境のGUIモードでのアンインストール - アンインストールの初期画面	69
[図 5.20]	UNIX環境のGUIモードでのアンインストール - Uninstall JEUS	70
[図 5.21]	UNIX環境のGUIモードでのアンインストール - Uninstall Complete	70
[図 6.1]	Windows環境でのインストール - インストール・ツールの初期画面	74
[図 6.2]	Windows環境でのインストール - Introduction	74
[図 6.3]	Windows環境でのインストール - License Agreement	75
[図 6.4]	Windows環境でのインストール - Choose Kernel Bit	75
[図 6.5]	Windowsでのインストール - Installation Folder	76
[図 6.6]	Windows環境でのインストール - Installation Type	76
[図 6.7]	Windows環境でのインストール - Installation Mode	77
[図 6.8]	Windows環境でのインストール - JDK Path	78
[図 6.9]	Windowsサービスの登録選択	78
[図 6.10]	Windows環境でのインストール - Choose Password	79
[図 6.11]	Windows環境でのインストール - Domain name	79
[図 6.12]	Windows環境でのインストール - Pre Installation Summary	80
[図 6.13]	Windows環境でのインストール - Installing JEUS	80
[図 6.14]	Windows環境でのインストール - Install Complete	81

[図 6.15]	JEUSインストール後のディレクトリー構造	82
[図 6.16]	WebAdminのログイン画面	90
[図 6.17]	WebAdminのメイン画面	91
[図 6.18]	Windows環境でのアンインストール - アンインストールの初期画面	94
[図 6.19]	Windows環境でのアンインストール - Uninstall JEUS	94
[図 6.20]	Windows環境でのアンインストール - Uninstall Complete	95
[図 8.1]	ログイン画面	103
[図 8.2]	メイン画面	103
[図 8.3]	MSの追加 - サーバー・リストの照会画面	104
[図 8.4]	MSの追加 - 基本情報の設定	105
[図 8.5]	MSの追加 - サーバー追加結果の確認	105
[図 8.6]	MSの追加 - リスナーの選択	106
[図 8.7]	MSの追加 - リスナー設定の変更	107
[図 8.8]	HTTPリスナーの設定 - リスナーの追加	108
[図 8.9]	HTTPリスナーの設定 - 基本情報の設定	108
[図 8.10]	HTTPリスナーの設定 - Webコネクションの追加	109
[図 8.11]	HTTPリスナーの設定 - Webコネクションの基本設定	109
[図 8.12]	WebtoBコネクタの設定 - Webコネクションの追加	111
[図 8.13]	WebtoBコネクタの設定 - Webコネクションの基本設定	112
[図 8.14]	データソースの追加 - JDBCデータ・リスト	114
[図 8.15]	データソースの追加 - 基本情報の設定	115
[図 8.16]	データソースの追加 - データソース・リストの照会	115
[図 8.17]	データソースの追加 - データソースの選択	116
[図 8.18]	JEUS WebAdmin - MSの制御画面	116
[図 9.1]	DASにインストールされているアプリケーション・リスト	121
[図 9.2]	Webモジュールのインストール	121
[図 9.3]	Webアプリケーションのインストールの成功	122
[図 9.4]	Webモジュールのデプロイ	122
[図 9.5]	WARモジュールのJSP呼び出し	124
[図 9.6]	WARモジュール・サーブレットの呼び出し	125
[図 10.1]	DASにインストールされたアプリケーション・リスト	130
[図 10.2]	HelloEJBのデプロイ	130
[図 10.3]	HelloEJBのデプロイ選択事項	131
[図 10.4]	HelloEJBのデプロイ結果	131
[図 10.5]	HelloEJBのクライアント画面	132
[図 10.6]	HelloEJBのサーブレット・クライアントの実行結果	133
[図 10.7]	DASにインストールされたアプリケーション・リスト	140
[図 10.8]	インストール対象の選択	140
[図 10.9]	Webアプリケーションのインストールの成功	141
[図 10.10]	HelloJPAのデプロイ・オプション	141
[図 10.11]	HelloJPAのデプロイ結果	142
[図 10.12]	HelloJPAのクライアント画面	143
[図 10.13]	HelloJPAのサーブレット・クライアントの実行結果	143

[図 11.1]	正常にデプロイされたAddNumbers	147
[図 11.2]	正常にデプロイされたFrom WSDL方式のWebサービス	149

表目次

[表 1.1]	JEUS Client Layer	9
[表 2.1]	JEUSの管理ツール	15
[表 2.2]	JEUSの環境変数	20
[表 4.1]	システム要件	35
[表 4.2]	プラットフォーム対応環境	36
[表 5.1]	環境変数の設定	86
[表 6.1]	環境変数の設定	86

例目次

[例 9.1]	<<HelloWorldServlet.java>>	117
[例 9.2]	<<snoop.jsp>>	118
[例 9.3]	<<snoop-jstl.jsp>>	119
[例 10.1]	<<Hello.java>>	127
[例 10.2]	<<HelloBean.java>>	127
[例 10.3]	<<HelloClient.java>>	128
[例 10.4]	<<Product.java>>	133
[例 10.5]	<<ProductManager.java>>	135
[例 10.6]	<<ProductManagerBean.java>>	135
[例 10.7]	<<ProductManagerClient.java>>	137
[例 11.1]	<<AddNumbersImpl.java>>	146
[例 11.2]	<<AddNumbersImpl.java>>	148
[例 11.3]	<<AddNumbersClient.java>>	150
[例 B.1]	<<hosts>>	155
[例 C.1]	<<jeus.ini>>	159

このガイドについて

対象読者

本書は、JEUS[®](以下、JEUS)の基本概念とインストール方法および使用例について説明します。

前提知識

JEUSの基本的な使用方法と製品を理解するには、以下のガイドについてあらかじめ熟知することをお勧めします。

- JEUS 紹介ガイド
- JEUS インストール & スタートガイド

本書のすべてのサンプルと環境構成は、UNIXスタイルに準拠します。Microsoft Windows[™](以下、Windows)など他の環境で作業を行う場合は、次のような事項を考慮してください。たとえば、Windowsプラットフォームでは、ディレクトリー区切り子をUNIXスタイルのスラッシュ(/)からWindowsスタイルのバックスラッシュ(\)に変えて使用してください。また、環境変数もWindowsスタイル(%%)に変更して使用してください。本書で触れているJEUS_HOMEは、JEUSがインストールされているディレクトリーです。

制限事項

本書の内容は、Java標準に準拠して作成されていますが、本書で触れているJava EEやJava仕様については詳しく取り上げていません。関連内容についてはJava関連ドキュメントを参照してください。

本書の構成

本書は、3つのPartと11の章および3つの付録で構成されています。

- [Part I, 「JEUS 紹介ガイド」](#)

JEUSサーバーについての紹介と環境およびガイドの構成について説明します。

- [Part II, 「JEUS インストールガイド」](#)

JEUSサーバーについての紹介と環境、UNIXとWindowsでJEUSをインストールおよびアンインストールする方法について説明します。

- [Part III, 「JEUS スタートガイド」](#)

JEUSの環境設定と起動方法について説明します。また、例を挙げてWebTier、EJB、Wevサービス環境での使用方法について説明します。

表記上の規則

表記	意味
<<AaBbCc123>>	プログラム・ソースコードのファイル名
<Ctrl>+C	CtrlキーとCキーを同時に押す
[Button]	GUIのボタン、メニュー名
太字	強調
「」、『』（鍵カッコ）	関連文書、あるいはガイド内の他の章および節の表示
「入力項目」	画面UI上の入力項目
ハイパーリンク	メール・アカウント、Webサイト
>	メニューの実行順
+----	下位ディレクトリー/ファイル有り
----	下位ディレクトリー/ファイル無し
<div>参考</div>	参照/注意事項
<div>注</div>	注意事項
[図 1.1]	図の名前
[表 1.1]	表の名前
AaBbCc123	Javaコード、XMLドキュメント
[<i>command argument</i>]	オプション・パラメータ
< xyz >	「<」と「>」の間の内容は実際に使用される特定の名前または値で置き換えられる
	構文の中の相互に排他的な選択項目の選択肢を示す 例) A B: AとBのいずれかを選択
...	パラメータ、値、または他の情報が繰り返される
\${ }	環境変数

システム要件

	要求事項
プラットフォーム	Solaris 9, 10, 11
	HP-UX 11.x, 11i, 11iV2
	IBM AIX 5L, 6L, AIX 7L
	MS Windows 2008, 2012, Vista, 7, 8
ハードウェア	最小2GB以上、推奨20GBのハードディスク容量
	推奨1GB以上のメモリー容量
JDK	JDK 7, JDK 8

関連文書

ガイド	説明
JEUS 紹介ガイド	JEUSサーバーについて全般的に紹介し、JEUSのアーキテクチャーを含む各構成要素について記述しています
JEUS インストール&スタートガイド	JEUSについて紹介し、JEUSのインストールおよび開始方法について記述しています
JEUS サーバガイド	JEUSシステムおよびサーバーの概要とシステムの管理方法について記述しています
JEUS Webエンジンガイド	JEUS Webエンジンの管理方法、Java EE WARアーカイブとサーブレット/JSPの管理およびデプロイ方法について記述しています
JEUS EJBガイド	JEUS EJBエンジンおよびEJBモジュールのデプロイについて記述しています
JEUS WebAdminガイド	JEUSのWeb管理ツールであるWebAdminを利用したJEUSの設定および制御、モニタリング、クラスタリング、リソースの設定および管理について記述しています

参考文献

- The Java EE 6 Specification
- The JNLP Specification
- The Java WebStart Specification

お問合せ先

Korea

TmaxSoft Co., Ltd.
45, Jeongjail-ro, Bundang-gu,
Seongnam-si, Gyeonggi-do, 13613
South Korea
Tel: +82-31-8018-1000
Fax: +82-31-8018-1115
Email: info@tmax.co.kr
Web (Korean): <http://www.tmaxsoft.com>
TechNet: <http://technet.tmaxsoft.com>

USA

TmaxSoft Inc.
101 North Wacker Drive, Suite 2014,
Chicago, IL 60606
U.S.A
Tel: +1-312-525-8330
Email: info@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/us_en/home

Japan

TmaxSoft Japan Co., Ltd.
5F Sanko Bldg, 3-12-16 Mita, Minato-Ku, Tokyo, 108-0073
Japan
Tel: +81-3-5765-2550
Fax: +81-3-5765-2567
Email: info@tmaxsoft.co.jp
Web (Japanese): <http://www.tmaxsoft.co.jp>

China

Beijing TmaxSoft System Software Co., Ltd.
Room103, No.2 Huizhong Building, Seven Street Shangdi,
Haidian District, Beijing, 100085
P.R.China
Tel: +86-10-6298-8827
Email: info@tmaxsoft.com.cn
Web (Chinese): http://www.tmaxsoft.com/cn_en/home_cn_en

Brazil

Tmax Brasil Sistemas e Serviços Ltda.
Av. Copacabana, 177, sala 32~35 Empresarial 18 do Fortel
Alphaville Barueri, Sao Paulo, 06472-001
Brazil
Tel: +55-11-4191-3100
Fax: +55(11) 4191-3705 (extension#112)
Email: info.bra@tmaxsoft.com
Web (Portuguese): http://www.tmaxsoft.com/br_en/home_br_en

Russia

Tmax Rus L.L.C.
Leninsky prospekt, 113/1 (Park Place Moscow),
Office 318e, Moscow, 117198
Russia
Tel: +7(495)970-01-35
Email: info.rus@tmaxsoft.com
Web (Russian): http://www.tmaxsoft.com/ru_ru/home_ru_ru

Singapore

Tmax Singapore Pte. Ltd.
430 Lorong 6, Toa Payoh #10-02,
OrangeTee Building, 319402
Singapore
Tel: +65-6259-7223
Fax: +65-6258-7112
Email: info.sg@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/sg_en/home_sg_en

United Kingdom

TmaxSoft UK Ltd.
215 Knyvett House, Watermans Business Park,
The Causeway, Staines TW18 3BAB
United Kingdom
Tel: +44-1784-895005
Email: info.uk@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/gb_en/home_gb_en

Canada

TmaxSoft Canada, Inc.
2425 Matheson Blvd East, 8th floor,
Unit 824 Mississauga, ON, L4W 5K4
Canada
Tel: +1-905-361-2888
Email: info.canada@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/ca_en/home_ca_en

Australia

TmaxSoft Proprietary Limited
L32, 101 Miller Street, North Sydney 2060
Australia
Tel: +91-9845-330-704
Email: info.aus@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/au_en/home_au_en

India

TmaxSoft Technologies Private Limited
Sobha Alexander Plaza, 3rd Floor,
16/2 Commissariat Road, Bangalore-560025
India
Tel: +91-9845-330-704
Email: info.india@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/in_en/home_in_en

Turkey

TmaxSoft Co., Ltd. Turkey Liaison Office
Windowist Tower. Eski Buyukdere Cad. No:26,
Maslak 34467 Istanbul
Turkey
Tel: +90-544-553-6045
Email: cslee@tmaxsoft.com
Web (English): http://www.tmaxsoft.com/tr_en/home_tr_en

Part I. JEUS 紹介ガイド

第1章 紹介

本章では、JEUSに関する基本概念とJava EE仕様について説明し、JEUSシステムの概念および構成要素、各エディションの機能について説明します。

1.1. 概要

JEUS(Java Enterprise User Solution)は、Web環境においてアプリケーションを開発、運用、実行することが可能なプラットフォームの役割をします。なお、包括的なJava基盤のWebアプリケーション・サービスと管理機能を提供します。JEUSでは、Java EEアプリケーションを実行するときに必要なプラットフォームと以下の構成要素を提供します。

- EJBコンテナ
- Webコンテナ(JSP/サーブレット・エンジン)
- セキュリティー・モジュール
- ネーミング・サーバー
- トランザクション・マネージャー
- JDBCコネクション・プール
- セッション・マネージャー

Java EE

JEUSは、Oracle社のJava Enterprise Edition仕様に準拠しており、Java EE 7認証を取得しています。以下は、Oracle社のWebサイトに掲載されている内容で、JEUSではそのすべての技術に対応します。

"The Java Platform, Enterprise Edition (Java EE) defines the standard for developing multitier enterprise applications. The Java EE platform simplifies enterprise applications by basing them on standardized, modular components, by providing a complete set of services to those components, and by handling many details of application behavior automatically, without complex programming."

"The primary technologies in Java EE are: Enterprise JavaBeans™ (EJBs™), JavaServer Pages™ (JSPs™), Java Servlets, the Java Naming and Directory Interface™ (JNDI™), the Java Transaction API (JTA), CORBA, and the JDBC™ data access API."

参考

Java EEについての詳細情報は、Oracle社のホームページ(<http://www.oracle.com/technetwork/java/javaee/overview/index.html>)を参照してください。

以下は、Java EEの仕様のうち、JEUS 8に実装可能な機能の一覧です。

Spec	JEUS 8
J2EE	Java EE 7
WebServer	WebtoB 5.0
HTTP	1.0/1.1/2.0
CGI	1.1
PHP	3.x/4.x/5.x
SSL	/ssl3/tls1
WebSocket	1.0
Java API for JSON Processing	
Batch Application for the Java Platform	
Concurrency Utilities for Java EE	1.0
EJB	3.2
RMI-IIOP	対応
JAF	1.1
EJB to CORBA Mapping	1.1
JCA	1.7
JMS	2.0
Common Annotations for the Java Platform	1.2
JPA	2.1
JavaMail API	1.5
JTA	1.2
JTS	1.0
JNDI	1.2.1
CDI	1.1

Spec	JEUS 8
Bean Validation	1.1
Managed Beans	1.0
Interceptors	1.2
JSP	2.3
Servlet	3.1
JSF	2.2
JSTL	1.2
XSLT	1.0
JDBC	4.0
IBM MQ	対応
Sonic MQ	対応
J2EE Management	1.1
J2EE Deployment API	1.2
JAAS	1.0.1
JACC	1.5
JASPIC	1.1
JAXP	(JDK 7に含まれて提供)
StAX	(JDK 7に含まれて提供)
JAXB	2.2
SAAJ	1.3
Implementing Enterprise Web Services	1.3
Web Service Metadata	2.1
JAX-RPC	1.1
JAX-WS	2.2
JAX-RS	2.0
JAXR	1.0
UDDI	2.0/3.0
SOAP	1.1/1.2
WSDL	1.1
WS-I Basic Profile	1.1
WS-Policy	1.5
WS-Policy Attachment	1.5

Spec	JEUS 8
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	対応
Java IDL API	対応
IDE Tool	未対応
GUI Tool	未対応
Web Tool	WebAdmin
Monitoring Tool	Console Tool、WebAdmin
JDK	7、8

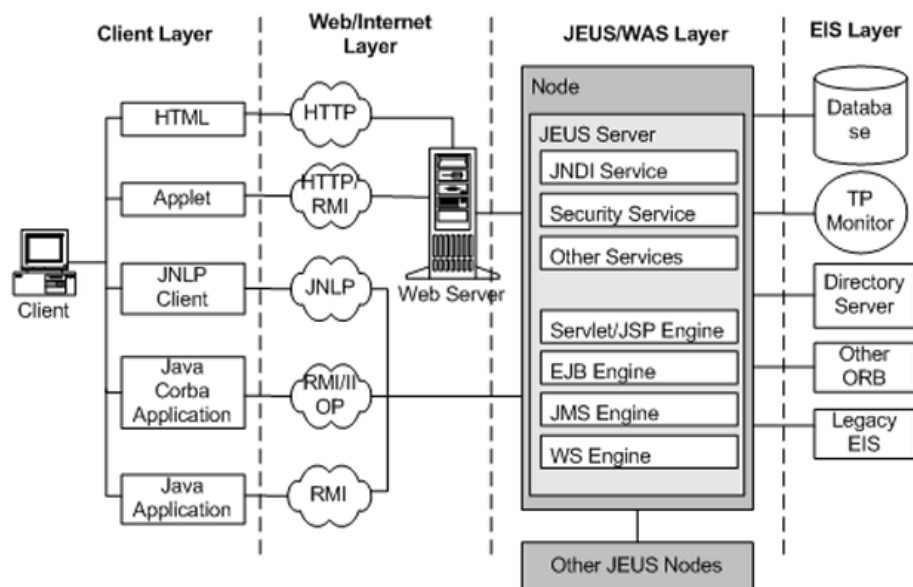
参考

- 仕様については、<http://www.oracle.com/technetwork/java/javaee/tech/index.html>で確認できます。
- 使用するJEUSエディションによっては、上記の表に記されている機能のうち、一部の機能が実装できない場合もあります。詳細内容は、各JEUSエディションの紹介ガイドを参照してください。

1.2. システムの概念と役割

以下は、エンタープライズ・アプリケーション・ソリューションを提供するため、JEUSが他のWebサーバーやDBMSなどどう連携されるのかを示しています。

[図 1.1] JEUSの機能と役割



上記の図で記述した4つのレイヤー(Layer)は以下のとおりです。

- **Client Layer**

Webサーバーやアプリケーション、またはネイティブ・アプリケーションで構成されます。エンドユーザーはWASのサービスを利用するために多様なクライアントを使用しており、このクライアントは様々なプロトコルのうち1つを使ってWASのサービスにアクセスします。

- **Web/Internet Layer**

クライアントとWASの間で動作するWebサーバーやプロトコルで定義されます。このレイヤーでは静的なコンテンツと負荷分散を処理します。

- **JEUS/WAS Layer**

Java基盤のミドルウェアで構成されており、Web LayerまたはClient Layerからの要求を受けて処理します。

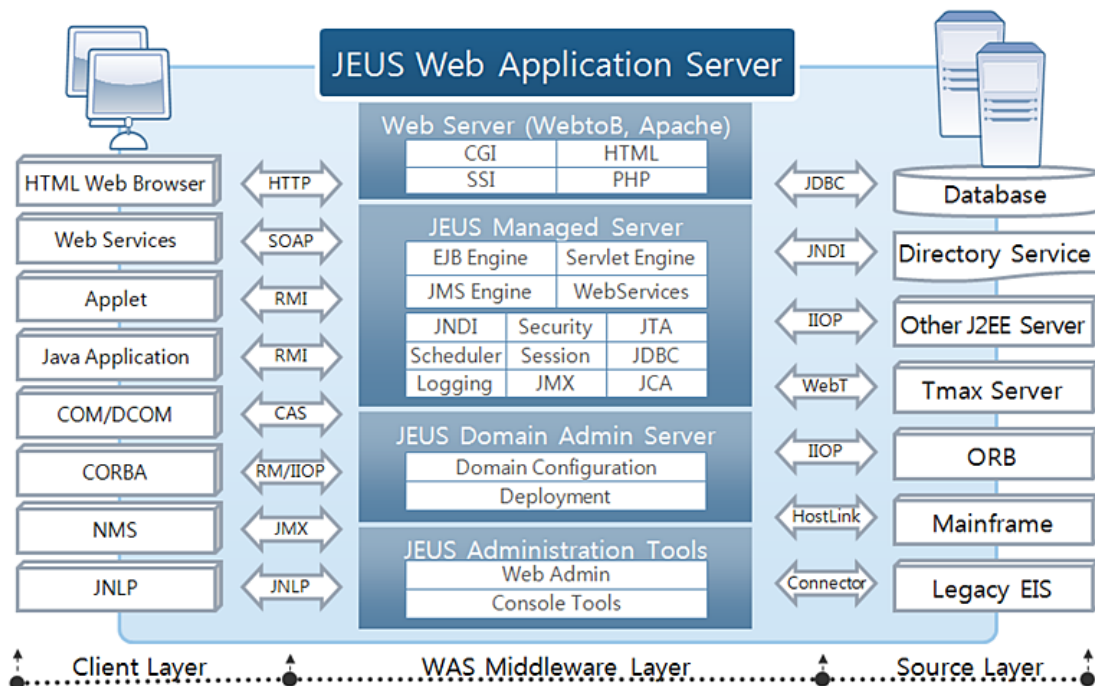
- **EIS Layer**

ビジネス・データや既存のレガシー・サービスです。WASは、JDBCやディレクトリー・サービス、Java EE コネクターなどの多様なメカニズムを通じてレガシー・サービスと相互作用します。

1.3. 構成要素とアーキテクチャー

JEUSは、複数の異なるモジュールで構成されています。このようなモジュールは以下の[図 1.2]のように、クライアント・アプリケーションとデータ記憶装置、JEUS間の通信技術に合わせて使用されます。

[図 1.2] JEUSのWebアプリケーション・アーキテクチャーのダイアグラム



Client Layerには、多様な形式のクライアント・アプリケーションと通信プロトコルが、Source Layerには、多様な形式のバックエンド(back-end)データ記憶装置が示されています。TmaxSoftのJEUS製品は、WebtoBと共に図の中央部を構成しているWAS Middleware Layerです。Webサーバーはクライアント・アプリケーションと連携されており、WebサーバーはWebアプリケーション・サーバーと緊密に統合されています。なお、Webゲートウェイ(WebT)は、WASとTPモニター(Tmax Server)を、メインフレーム・ゲートウェイ(Host-Link)はメインフレームとTmaxサーバー間の連携をサポートします。

以下の節では、上記の図の3レイヤー(Client Layer、WAS Middleware Layer、Source Layer)の構成要素について説明します。

1.3.1. Client Layer

Client Layer(クライアント階層)では、JEUSが使用できるリモート、あるいはローカル・アプリケーションを表しています。

以下は、Client Layerの構成タイプに関する説明です。

[表 1.1] JEUS Client Layer

Client Layer	
Web browser	HTMLコンテンツを取得するため、JEUSサーブレット・エンジンとWebtoB LightのWebサーバに要求する標準Webブラウザとして、最も一般的なクライアント・アプリケーションです。通信プロトコルはHTTPです
Web Services	Webサービスの実装をサポートします
Applet	JEUSの構成要素が参照できるアプレット・コンテナを提供します
Java Application	通常独立したJavaアプリケーションは、JEUSが提供するクライアント・コンテナ内でRMIを使用して実行されます。このようなクライアントをJava EE仕様ではアプリケーション・クライアントといいます
COM/DCOM	Microsoft Windows環境でEJBをCOM形式で呼び出すことができます
CORBA	CORBA技術を使用したアプリケーションもRMI/IIOPを通じてJEUSを使用することができます
NMS	ネットワーク管理システムはJMXを通じてJEUSを管理、使用することができます
JNLP	JEUS 8は、JNLP(Java Network Lanuching Protocol)クライアントをサポートします

1.3.2. WAS Middleware Layer

[\[図 1.2\]](#)でのJEUS/WAS Layerは、Java EE 7仕様が認証されたJEUS 8を示しており、構成要素は以下のとおりです。

● JEUS Domain Administration Server(DAS)

ドメイン内にはDomain Administration Server(以下、DAS)という特別なサーバーが常に存在しています。DASは、ドメイン内のすべてのアプリケーションおよびリソースを中央で管理し、サーバーを制御、モニタリングする管理ツール(WebAdmin、jeusadmin)と通信します。

サービス	説明
Domain Configuration	ドメイン全体の設定を管理します
Application Management	ドメイン全体のアプリケーションを管理します
Administration	WebAdminとjeusadminを利用し、ドメイン内の全サーバー、サービス、アプリケーション、リソースを一括にモニタリングおよび制御することができます

● JEUS Managed Server(MS)

Managed Server(以下、MS)は、JEUSシステムで構成できる多様な形式のエンジンとサービス基盤を提供しています。

エンジン/サービス	説明
EJB Engine	EJBビジネス・アプリケーションを実行します
Servlet Engine	Webコンテナとして、静的なコンテンツ(HTML)のみならず、JSP/サーブレット・アプリケーションを実行します
JMS Engine	JMS基盤の構造を提供します
Web Services Engine	JEUS Webサーバーのインスタンスです。サーブレット・エンジンのプロントエンド(front-end)として実行されます
JNDI Service	ネーミング・システムです
Security Service	認証/権限サービスです
JTA	Webアプリケーション・サーバーで実行される様々なアプリケーションに対する完全なトランザクションを提供します
Scheduler	決まっている時間にイベントを発生させるタイマー機能を提供します
Session Manager	クラスタリングが必要な場合、安全な方式でクライアントのセッション情報を保存します
JDBC	データベース・コネクション・プールが設定可能です
Logging	JEUSの実行中にシステムで実行されていた一連の作業内容を順番どおり保存、記録します
JMX	JEUSシステムをNMS/JMXクライアントが管理できるようにします
JCA	JCA(Java EE Connector Architecture)をサポートするすべてのレガシーEAI(Enterprise Application Integration)ソリューションに対してJEUSへの接続をサポートします

● Web Server(WebtoB、Apache)

WebサーバーはHTMLのような静的コンテンツと、CGIのような動的コンテンツを転送します。また、サーブレット・エンジンのプロントエンドとして相互作用します。WebtoBには、TmaxSoftのWebtoB Webサーバーとして全機能をサポートするバージョンと、一部縮小された機能のみ有するJEUS Webサーバーの2つのバージョンがあります。

JEUS WebサーバーはJEUSに含まれており、WebtoBには含まれていません。なお、オープンソースWebサーバーであるApacheをJEUSで使うことができます。

● JEUS Administration Tools

JEUSには、以下の2つの主要管理ツールが存在します。

ツール	説明
WebAdmin	JEUSのモジュール・パッケージング、設定以外のすべての構成要素を管理しており、Webブラウザで利用可能なWebツールです

ツール	説明
コンソール・ツール (jeusadmin)	グラフィック・ツールを利用して使用可能なすべての機能はコマンドライン・ベースのコンソール・ツールを使っても実行できます。このツールはグラフィック環境を使用しないときに使います

1.3.3. Source Layer

[図 1.2]の右部に位置するSource Layerは、バックエンドのリソースとJEUSシステムによって使用可能なデータの保存場所を表しており、そのタイプは以下のとおりです。

ソース・レイヤー	説明
Database	JEUSでJDBCを介して接続できます
Directory Service	LDAPのようなものがあり、JNDIを利用します
Other Java EE Server	JEUSは他ベンダーのJava EEサーバーと相互作用が可能です
Tmax Server	TmaxSoftで開発したTPモニターです。WebT APIライブラリーは、JEUSとTmaxを連動するために使われます
ORB	IIOP(Internet Inter-ORB Protocol)を介して参照されます
Mainframe	IBMメインフレームは、ホスト・リンク(または、Javaコネクタ)製品を通じて使われます
Legacy EIS	JCAをサポートするレガシーEISとして、JEUSとの相互作用が可能です

1.4. 相互運用モジュール

相互運用性とは、2つもしくはそれ以上のシステム(コンピューター、通信装置、ネットワーク、ソフトウェア、または他の情報技術要素)で定義された方法を利用したデータの送受信や相互作用を意味します(ISO ITC-215)。JEUSでは、異なるプロトコル、Webサービス、JBuilder Bridge、JNLP、RMI-IIOPのような技術をサポートします。

以下は、JEUSの相互運用のために提供されるモジュールです。

モジュール	説明
RMI-IIOP	IIOP(Internet Inter-ORB Protocol)で実行されるRMI技術です。JAVAプラットフォームでのCORBAの分散コンピューティング処理を実行可能にします
JEUS	他ベンダーのWebアプリケーションの使用を可能にし、Webサービスにも対応しています
WebT	TPモニターとJEUSを連携するゲートウェイです
Host-Link	レガシーEISに存在するサービスをクライアントが使用できるようにするアダプター・モジュールです

モジュール	説明
JCA	JEUSとJEUSクライアントがバーチャルで全レガシーEISと相互作用できるようにします

1.5. エディション

以下は、JEUS 8のエディションについての説明です。

エディション	主要特性
JEUS Standard Edition	<ul style="list-style-type: none"> – JEUS Server – JEUS Web Server – Servlet 3.1 – JSP 2.3 – JSF 2.2 – JSTL 1.2 – Security – JDBC connection pooling – XMLユーティリティ – JNDI – JMX – WebAdmin – JEUS node clustering – EJB 3.2 – JMS 2.0 – 2 Phase-CommitサポートJTA – JCA 1.7 – SNMP – Web service
JEUS Enterprise Edition	<ul style="list-style-type: none"> – JEUS Standard Edition – Session clustering

エディション	主要特性
	– JMS clustering

参考

1. 提供されるJEUSガイドには、上記表に記述されているすべての項目が含まれています。この表より、エディション別に使用できる機能を確認してください。なお、JEUSをインストールすると、トライアル・ライセンスをご利用頂けます。
 2. Standard EditionとEnterprise Editionの各々のクラウド・ライセンスを提供しています。詳しいクラウド・ライセンス・ポリシーについては、弊社まで別途お問い合わせください。
-

第2章 JEUSの環境

本章では、JEUSで使用する管理ツール、ディレクトリー構造、環境変数について説明します。なお、JEUSに関するXML設定ファイルの全般的な内容について紹介します。

2.1. 管理ツール

以下は、JEUSに接続して使用するツールについての説明です。

[表 2.1] JEUSの管理ツール

ツール	説明
WebAdmin	HTMLベースの管理ツールです。JEUSの制御、モニタリング、管理などの機能を提供します。詳細内容は、『JEUS WebAdminガイド』を参照してください
startDomainAdminServer	JEUS DASを実行する最も基本的なツールです。詳細内容は、『JEUS XMLリファレンス』を参照してください
startManagedServer	JEUS MSを実行する最も基本的なツールです。詳細内容は、『JEUS リファレンスガイド』を参照してください
jeusadmin	コンソール・ツールは、コマンド・プロンプトでJEUSを管理するために使われます。詳細内容は、『JEUS リファレンスガイド』の「4.2. jeusadmin」を参照してください

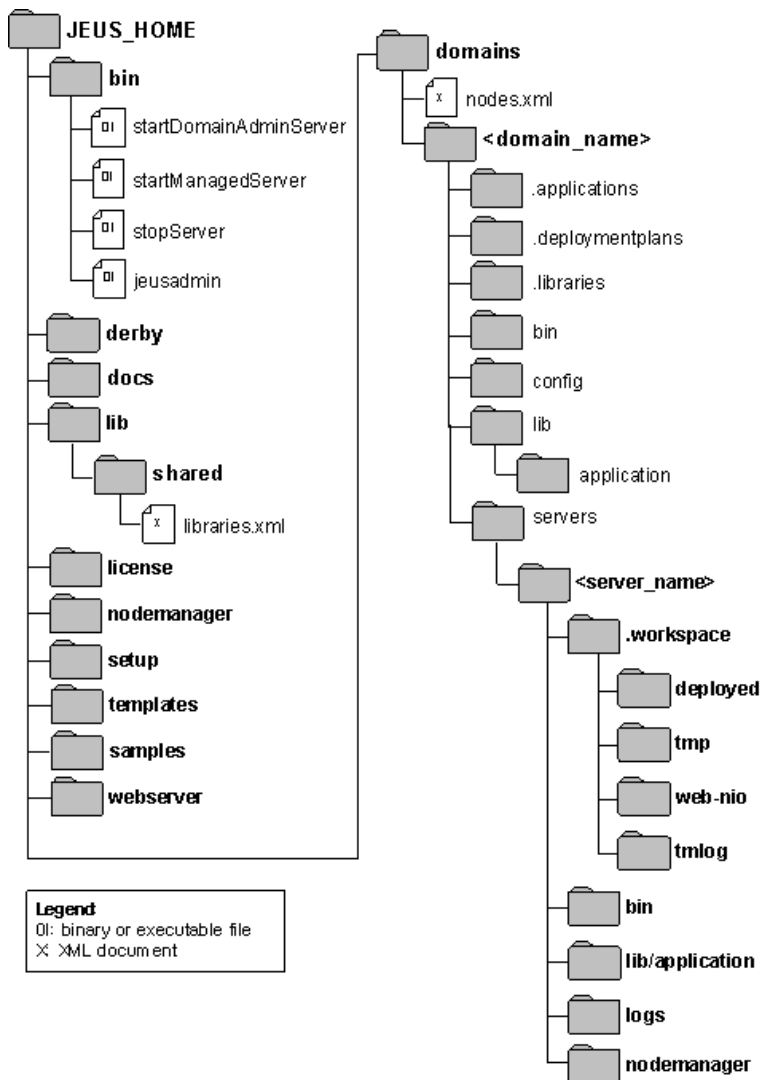
参考

上述以外にもDASとMSをダウンさせるコマンドがあります。詳細内容は、『JEUS リファレンスガイド』を参照してください。

2.2. ディレクトリー構造

以下は、JEUSをインストールしたときのディレクトリー構造です。

【図 2.1】 JEUSインストール後のディレクトリー構造



以下は、ディレクトリーとファイルについての説明です。

{JEUS_HOME}

JEUSのルート・ディレクトリーとして、実際のディレクトリー名と場所はインストール時に決まります。

bin

サーバーの起動および終了スクリプトである`startDomainAdminServer`、`startManagedServer`、`stopServer`とJEUSコンソール・ツール(`jeusadmin`)のような実行ファイルが存在します。

derby

サンプル・アプリケーションの作成やテスト環境で容易に使用できるようにApache Derbyが含まれています。

docs

JEUSが提供するAPIのJavadocが存在します。

lib

JEUSの起動に必要なライブラリーが存在します。ユーザーは、sharedディレクトリー以外のディレクトリーにはアクセスする必要がありません。

ディレクトリー	説明
shared	sharedディレクトリーには、アプリケーションで使用するライブラリーが存在します。 sharedディレクトリーのライブラリーを使用するには、libraries.xmlにライブラリーの情報を追加する必要があります。なお、当該ライブラリーを使用するアプリケーションのJEUS Deployment Descriptor(DD)で当該ライブラリーについてのリファレンス情報を指定します。sharedライブラリーの詳細内容は、『 <i>JEUS アプリケーション&デプロイメントガイド</i> 』の「3.3.2. 共有ライブラリー」を参照してください

license

JEUSのライセンス・ファイルが存在します。ライセンス・ファイルは、JEUSが実行されるための必須ファイルです。

nodemanager

JEUSのノード・マネージャーのための設定ファイルであるjeusnm.xmlファイルが存在します。

setup

JEUSのインストール後、使用できる環境を構築するための必要ファイルが存在します。

templates

各種設定や環境などのテンプレート・ファイルが存在します。

samples

JEUSのサンプル・ファイルが存在します。

webserver

JEUSのインストール時にJEUS Webサーバーがインストールされるディレクトリーです。詳しい内容は、『*JEUS Webエンジンガイド*』を参照してください。

domains

各ドメインの下位にDOMAIN_HOMEとJEUS_HOMEで使用するノード情報が含まれているnodes.xmlが存在します。

以下のディレクトリーおよびファイルは、DOMAIN_HOMEの下位に存在します。

- .applications

各ドメインで管理するアプリケーション・ファイルが存在します。

追加および削除には、**install-application**、**uninstall-application**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.7. install-application」、『*JEUS リファレンスガイド*』の「4.2.6.16. uninstall-application」を参照してください。

- .deploymentplans

ドメインで管理するデプロイメント・プラン・ファイルが存在します。

追加および削除には、**install-deployment-plan**、**uninstall-deployment-plan**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.8. install-deployment-plan」、『*JEUS リファレンスガイド*』の「4.2.6.17. uninstall-deployment-plan」を参照してください。

- .libraries

ドメインで管理するライブラリー・ファイルが存在します。

追加および削除には、**install-application**、**uninstall-application**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.7. install-application」、『*JEUS リファレンスガイド*』の「4.2.6.16. uninstall-application」を参照してください。

- bin

各ドメインに属しているDASとMSの開始および終了スクリプトが存在します。JEUS_HOME/binのstartDomainAdminServer、startManagedServer、stopServerと同様な機能を実行しますが、ドメイン名を設定する必要はありません。

- config

ドメインの設定ファイルであるdomain.xmlが変更された場合、以前のヒストリーのために存在するバックアップ・ファイルが格納されます。ドメイン設定についての詳細説明は、『*JEUS ドメインガイド*』の「第3章 ドメインの設定変更」を参照してください。

区分	説明
security	<ul style="list-style-type: none"> – SYSTEM_DOMAIN : ドメイン単位で適用されるセキュリティ・ドメイン・ファイルのaccounts.xml、policies.xmlが存在しており、各XMLファイルはWebAdminとjeusadminを利用して動的な設定変更が可能です。セキュリティ・ドメイン設定についての詳細説明は、『JEUS セキュリティガイド』の「2.2. セキュリティ・ドメインの定義」を参照してください – security.key : 対称鍵暗号化アルゴリズムのキーを保存するファイルです。JEUS_HOME/bin/encryptionを実行すると作成されます。security.keyファイルについての詳細説明は、『JEUS セキュリティガイド』の「2.5.4. パスワードのセキュリティ設定」を参照してください – policy : Javaパーミッション設定ファイルです。JEUSのセキュリティ・システムとは別途にJava SEセキュリティ・マネージャーで使われます
servlet	<ul style="list-style-type: none"> – web.xml : web.xmlを持っていない場合、Webエンジンが使用するWebモジュールのweb.xmlです。デフォルト値は、空きのXMLファイルです – webcommon.xml: ドメイン内サーバーのWebエンジンの全Webモジュールに適用される共通設定ファイルです。設定についての詳細内容は、『JEUS Webエンジンガイド』の「1.5. ディレクトリー構造」を参照してください

- lib/application

全ドメインに適用したいアプリケーション・ライブラリーを格納するディレクトリーです。

SERVER_HOMEに存在するアプリケーション・ライブラリーと衝突した場合、SERVER_HOME/lib/applicationが優先され、警告メッセージが残ります。lib/applicationディレクトリーについての詳細説明は、『JEUS アプリケーション&デプロイメントガイド』の「3.3.1. lib/applicationディレクトリー」を参照してください。

- servers

このディレクトリーの下位にSERVER_HOMEディレクトリーがサーバー名で作成されます。SERVER_HOMEディレクトリー構造の詳細説明は、『JEUS サーバガイド』の「1.5. サーバー・ディレクトリー構造」を参照してください。

ディレクトリー	説明
.workspace	JEUSが使用するサーバー別のスペースなので、ユーザーが変更してはなりません
bin	<p>サーバーの開始/終了スクリプトが含まれています。JEUS_HOME/binのスクリプトと同様な機能を行いますが、ドメイン名とサーバー名を設定する必要はありません</p> <ul style="list-style-type: none"> – DASの場合 : startDomainAdminServer/stopServerが存在します – MSの場合: startManagedServer/stopserverが存在します

ディレクトリー	説明
lib/application	<p>サーバーに適用したいアプリケーション・ライブラリーが存在します。ドメイン領域のライブラリー(DOMAIN_HOME/lib/application)より優先順位が高いです。ライブラリーが衝突した場合、同ディレクトリーに存在するファイルが適用され、警告メッセージが残ります。</p> <p>lib/applicationの詳細内容は、『<i>JEUS アプリケーション&デプロイメントガイド</i>』の「3.3.1. lib/applicationディレクトリー」を参照してください</p>
logs	<p>サーバーのランチャー(Launcher)ログ、サーバー・ログ、アクセス・ログファイルが残ります。詳細内容は、『<i>JEUS サーバガイド</i>』の「第8章 ロギング」を参照してください</p>
nodemanager	<p>ノード・マネージャーが再起動する際、管理していたサーバーなのかどうかを判断するための情報を保存します。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。</p>

2.3. 環境変数

環境変数はすべて「JEUS_HOME/bin/jeus.properties」に設定されており、JEUS_HOME/binディレクトリーの全スクリプトで使われます。

以下は、JEUSで使用する環境変数です。

[表 2.2] JEUSの環境変数

環境変数	内容
JEUS_HOME	<p>JEUSがインストールされているホーム・ディレクトリーとして、必須項目です</p> <p>(例 : JEUS_HOME=/home/jeus/jeus8)</p>
JAVA_HOME	<p>JDKのホーム・ディレクトリーです</p> <p>(例 : JAVA_HOME=/usr/jdk1.7)</p>

当該変数は必要に応じて修正して使用します。ただし、XML設定ファイルではこれらの環境変数は使用できません。すべての環境変数はインストール時にデフォルト値で設定されます。ほとんどの場合、設定値をそのまま使用します。

参考

環境変数を変更する方法はOSによって異なるため、各OSガイドを参照してください。

2.4. 環境設定ファイル

JEUSは環境設定のためにそれぞれ固有のXMLフォーマットを使用しており、直接またはツールを使用して修正することができます。以下は、JEUSのXML設定ファイルの内容と場所です。

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

位置	JEUS_HOME/domains/<domain_name>/config/
説明	JEUSマネージャとノードを管理する基本設定ファイルです
参照ガイド	『JEUS ドメインガイド』、『JEUS サーバガイド』

- jeusnm.xml (jeus-nodemanager.xsd)

位置	JEUS_HOME/nodemanager
説明	ノードでノード・マネージャの動作方式を設定するファイルです
参照ガイド	『JEUS ノードマネージャガイド』

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

位置	Webアプリケーション・アーカイブのWEB-INF
説明	JEUS Webアプリケーション(Servlet app)のDD(Deployment Descriptors)ファイルです
参照ガイド	『JEUS Webエンジンガイド』

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

位置	EJBアプリケーション・アーカイブのMETA-INF
説明	JEUS EJBモジュールのDDファイルです
参照ガイド	『JEUS EJBガイド』

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

位置	クライアント・アプリケーション・アーカイブのWEB-INF
説明	アプリケーション・クライアントのDDファイルです
参照ガイド	『JEUS アプリケーションクライアントガイド』

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

位置	リソース・アダプター・アーカイブのMETA-INF
説明	リソース・アダプターのDDファイルです

参照ガイド	『JEUS JCAガイド』
--------------	---------------

- policies.xml (policies.xsd)

位置	JEUS_HOME/domains/<domainname>/config/security
説明	JEUSセキュリティ・ポリシーを設定したファイルです
参照ガイド	『JEUS セキュリティガイド』

- accounts.xml (accounts.xsd)

位置	JEUS_HOME/domains/<domainname>/config/security
説明	JEUSセキュリティ・アカウントを設定したファイルです
参照ガイド	『JEUS セキュリティガイド』

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

位置	Webサービス・クライアント・アーカイブのMETA-INF
説明	Webサービス・クライアント情報を設定したファイルです
参照ガイド	『JEUS Webサービスガイド』

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

位置	Webサービス・クライアント・アーカイブのMETA-INF
説明	Webサービス・クライアントのAntタスクで使用する設定ファイルです
参照ガイド	『JEUS Webサービスガイド』

参考

1. Java EEの標準記述子ファイルであるweb.xmlやejb-jar.xmlファイルも使用可能です。各ファイルについては該当のJava EE仕様を参照してください。
 2. すべてのXMLスキーマ・ファイルは、JEUS_HOME/lib/schemas/jeus/に存在します。
-

第3章 JEUSガイドの構成

本章では、ガイドの活用方法および使用された略字について紹介します。JEUSガイドの全般的な情報を得るためには本章を注意深くお読みください。

3.1. 概要

JEUSガイドは以下の3つの方法で取得できます。

- ソフトウェアと一緒に提供されるHTMLドキュメント
- JEUSインストーラーCDにより提供されるPDFファイル

PDFファイルを確認するには、Adobe Acrobat Reader、あるいはPDFに対応するソフトウェアが必要です。<http://get.adobe.com/reader/>よりAdobe Acrobat Readerをダウンロードすることができます。

- TmaxSoftのTechNet(<http://technet.tmax.co.kr/kr/index.do>)より、アップデートされたソフトウェアやガイドがダウンロードできます。

以下は、JEUSガイドを理解するための前提事項です。

- JEUSガイドはJavaとJava EE技術について経験のあるJava EEエンジニアを対象としています。
- 同書は、個別的かつ関連性を排除したドキュメントで構成されています。
- 一部を除いて、基本的には標準的なドキュメント形式で構成されています。
- このガイドは多様な方式で構成されています。

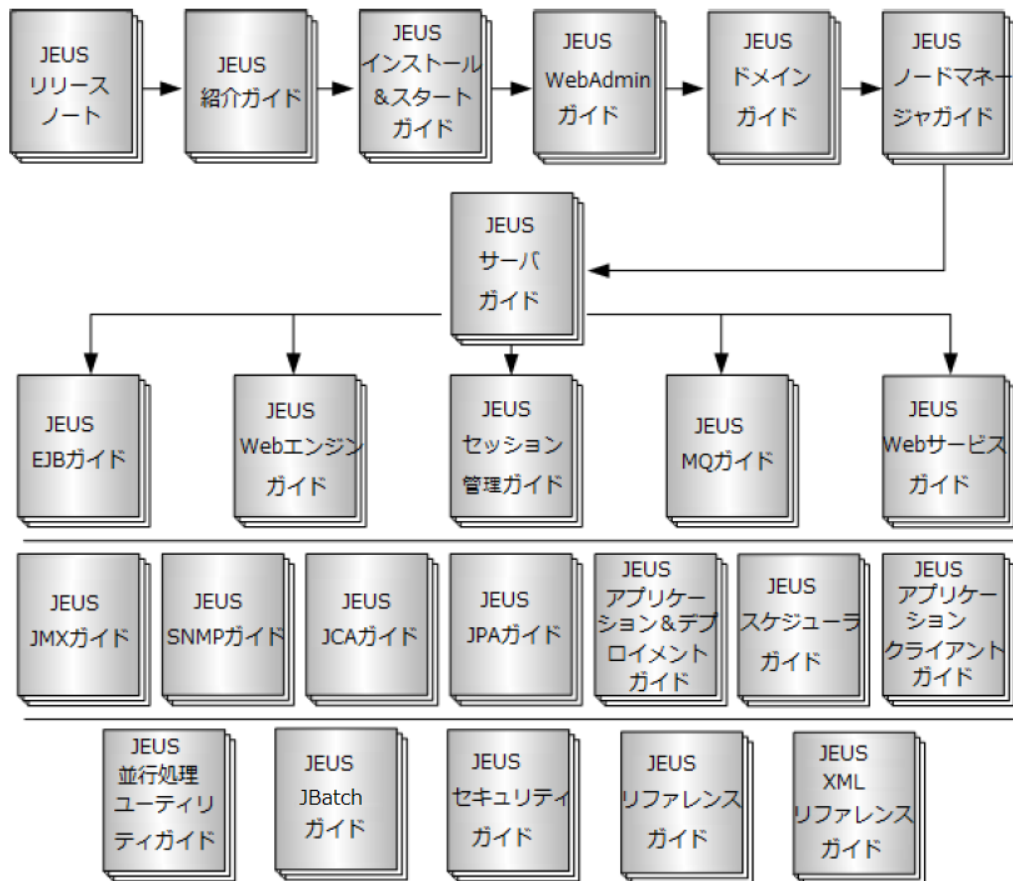
参考

本書を理解するには、JavaとJava EE技術に関する知識が必要です。様々な書籍やJava関連のWebサイト<http://www.oracle.com/technetwork/java/index.html>より、オンライン・ドキュメント、仕様、自習書などが取得できます。JEUSガイドでJava EE仕様に記述されているすべての内容を説明することは困難なため、今回はJEUSに特化された情報のみ記述します。

3.2. ガイドの構成

以下の[図 3.1]は、JEUSガイドの構成です。矢印はJEUSに不慣れなユーザーのため、ガイドの読む順を示しています。JEUSガイドは、計24書で構成されています。

【図 3.1】 JEUSガイドの構成



各ガイドの内容は以下のとおりです。特定のテーマについて検索する際には以下の一覧を参照してください。

● JEUS リリースノート

リリースされる製品の新機能や旧バージョンからアップグレードする方法について説明します。

- JEUSの新機能
- 旧バージョンからのアップグレード方法

● JEUS 紹介ガイド

JEUSの全般的な紹介とアーキテクチャーおよび各構成要素について説明します。

- JEUSサーバーについての紹介

- JEUSの環境
- JEUSガイドの構成
- **JEUS インストール&スタートガイド**

JEUSをインストールする方法やタイプ別に使用する例について説明します。

- WindowsとUNIXでのJavaインストール
- WindowsとUNIXでのJEUSインストール
- サンプル・アプリケーションについての説明(Getting Started)
- JEUSシステム入門
- EJB入門
- サーブレット/JSP入門
- Webサービス入門

- **JEUS WebAdminガイド**

JEUSのWeb管理ツールとして、一般ユーザーから管理者までの必須ガイドです。

- JEUS WebAdminの画面構成および主要機能
- JEUSの設定および制御、モニタリング

- **JEUS ドメインガイド**

JEUSドメインの構造、構成について全般的に説明しています。JEUSについての知識が必要な全ユーザーを対象としています。

- ドメインの概念と構成要素
- ドメインの作成方法とディレクトリー構造
- ドメインの設定変更および適用方法
- ドメインを構成しているサーバーの制御方法と、それに関するステータスの変更
- ドメインが構成できるクラスターの概念

- ドメインを管理するサーバーが異常終了した場合の問題点と解決方法
- MSを管理するためのノード・マネージャーに関する説明
- ドメインに関するセキュリティ事項

● JEUS ノードマネージャガイド

JEUSノード・マネージャーの構造と構成について説明します。JEUSについての知識が必要な全ユーザーを対象としています。

- ノード・マネージャーの概念、目的、種類などの基本的な事項
- Javaタイプのノード・マネージャーを使用するための設定方法
- Javaタイプのノード・マネージャーを使用するための動作方式
- SSHタイプのノード・マネージャーの設定と削除
- SSHタイプのノード・マネージャーの使用方法

● JEUS サーバガイド

JEUSを管理するためのガイドとして、JEUSシステムの管理者に最も必要な内容です。

- JEUSの構成要素および構成要素が提供するサービスの概要
- JEUSの構成要素を設定する方法
- JEUSサーバーの制御およびモニタリング方法
- JEUS JNDIの基本事項とアプリケーションの開発方法
- JEUSと連携して使用できる外部リソースの種類と設定方法
- JEUSが提供するコネクション・プールおよび付加機能と使用方法
- JEUSのトランザクション・マネージャーとその周辺要素
- JEUSのロギング・システム
- 主要JDBCドライバーに関するコネクション・プールの設定例

参考

『**JEUS サーバガイド**』の範囲は非常に広く、JEUSセキュリティ、ネーミング、そしてトランザクション・マネージャーのような異なるテーマも多く含まれています。このように構成された理由は、異なるテーマであってもこれらの構成要素は同様な環境設定ファイルで構成され、同様なJVMで実行されるためです。さらに、各ガイドの数を最小限にするためでもあります。

● JEUS EJBガイド

JEUS EJBエンジンとEJBモジュールをデプロイする方法について説明します。

- JEUS EJBの概要
- JEUS EJBエンジンの設定、制御およびモニタリングとチューニング方法
- EJBモジュールの管理、構成、デプロイ、制御、モニタリング方法
- 一般的なEJB(各Beanの構成)の共通的な特性
- JEUS EJBのセキュリティ
- JEUS EJBセキュリティの相互作用
- JEUS EJBクラスタリング
- セッションEJB
- エンティティEJB
- メッセージドリブンBean
- EJBクライアント
- JEUS EJBのためのAnt使用

● JEUS Webエンジンガイド

JEUS Webエンジンを管理するためのガイドです。Java EE WARアーカイブとサーブレット/JSPの管理やデプロイする方法について説明します。

- JEUS Webエンジンの概念および環境設定
- JEUS Webコネクションの管理

- JEUS Webコンテキスト(Webアプリケーション、WARファイルなど)の構造およびデプロイ、制御、モニタリング、チューニング方法
- JEUS Webエンジンの機能および設定方法
- Webサーバーとの接続とクラスタリング(WebtoB、Apache and built-in HTTP server connections and clusters)
- 仮想ホスティング
- JEUS WebCacheの概念と使用方法
- リバースプロキシ(Reverse Proxy)の概念と使用方法
- クラス動的適用の基本設定および動作

● JEUS セッション管理ガイド

JEUS Webエンジン、EJBエンジンなどで使用されるセッション・マネージャー(Session Manager)、セッション・サーバー(Session Server)の構成と、その設定について説明しています。主に、クラスタリング環境か、単一サーバーでのセッション維持、共有などを管理するシステム管理者と開発者を対象とします。

- セッション・トラッキング(Session Tracking)の構造、動作、設定およびチューニング方法
- クラスタリング環境でセッション・トラッキングのために運用される分散セッション・サーバーの構造、動作および設定方法

● JEUS MQガイド

JEUSメッセージ・ベースのシステム(JMS)について説明します。

- JEUS JMSの概要
- JMSエンジンの環境設定、モニタリングと制御、障害発生時の復旧方法
- JEUSでのJMSプログラミング
- JEUS MQのクラスタリングの種類と使用方法
- JEUS MQの特殊機能

● JEUS Webサービスガイド

JEUSのWebサービスについて説明します。

- JEUS Webサービスの概要
- Webサービス・バックエンドの作成
- Webサービスの呼び出し
- Webサービス・バックエンドを使用するクライアントの実装
- データ型とJEUS Webサービス
- Webサービスに関するAnt
- 標準バインディング宣言およびユーザー化
- ハンドラー・フレームワーク
- プロバイダーとディスパッチ・インターフェース
- 非同期Webサービス
- MIMEアタッチメント(Attachment)メッセージの転送およびFast Infosetを利用したWebサービス
- Webサービス・ポリシー、アドレッシング、信頼性メッセージ技術およびトランザクションに関する説明
- UDDIの利用
- Webサービス・セキュリティー
- WebサービスのXML

● JEUS JMXガイド

JMXを使用してJEUSを管理するためのガイドです。

- JEUS JMXマネージャーの設定
- JMXアプリケーションの開発
- JMX APIリファレンス

● JEUS SNMPガイド

産業標準であるSNMPを利用したJEUSモニタリングについて説明します。

- JEUS SNMPエージェントの概要
- SNMPエージェントの構成
- JEUS SNMPプログラミング
- JEUS SNMP MIB
-
- **JEUS JCAガイド**

JEUSとレガシー・システムを接続するためのコネクタについて説明します。

 - コネクタの概要
 - コネクタのパッケージング
 - コネクタの使用とチューニング
-
- **JEUS JPAガイド**

JEUSに統合されたTopLink EssentialをJEUSで使用するために必要な設定について説明します。

 - Java Persistence APIの紹介
 - プロバイダーの設定
 - JEUSの設定
-
- **JEUS スケジューラガイド**

JEUSのスケジューラ機能について説明します。

 - スケジューラ・サービスの設定
 - スケジューラ・サービスのプログラミング
-
- **JEUS アプリケーション&デプロイメントガイド**

Java EEアプリケーションをJEUSにデプロイするための様々な方法とツールについて説明します。

 - ドメイン環境におけるアプリケーションの管理方法
 - グレースフル・アンデプロイメントと再デプロイメント
 - モジュールとアプリケーション、共有ライブラリーに関する説明

- Java EEアプリケーション・ファイルの作成とデプロイ方法

● JEUS アプリケーションクライアントガイド

Java EEクライアントとJEUS間の相互運用について説明します。

- Java EEアプリケーション・クライアント
- アプレット・クライアント
- JNLPクライアント

● JEUS セキュリティガイド

JEUSでのセキュリティ・システムの設定、運用方法およびセキュリティ関連のプログラミングについて説明します。

- セキュリティ・システムの概要と設定
- アプリケーションとモジュールのセキュリティ設定
- セキュリティ・システムのAPIを利用したプログラミング
- カスタム・セキュリティ・サービスの開発
- JACCプロバイダーの使用方法
- JAASの使用方法

● JEUS 並行処理ユーティリティガイド

JEUSでの並行処理ユーティリティについて説明します。

- 並行処理ユーティリティの概要と設定
- 並行処理ユーティリティAPIを利用したプログラミング

● JEUS JBatchガイド

JEUSでのJBatchプログラミングについて説明します。

- JBatchの概要と設定
- JBatchを利用したプログラミング

- **JEUS リファレンスガイド**

JEUSの使用に役立つリファレンスをまとめたガイドです。

- システム・プロパティの内容
- コンソール・コマンドの使用方法
- Antタスクの使用方法
- JEUSで使用するAPI集

Part II. JEUS インストールガイド

第4章 インストールの概要および準備事項

本章では、インストールの手順に関する基本的な説明やインストール前に準備するシステムの環境および必要なJDKの環境構成について説明します。

4.1. 概要

WindowsおよびUNIX環境でのJEUSインストールを容易にするため、JEUSインストール・ツールはグラフィック・ユーザー・インターフェース(GUI)を提供します。コマンド・モード慣れのUNIX/Linux環境ではコンソール・モードでインストールすることが可能です。基本的にはコンソール・モードでインストールが行われますが、GUI形式を希望する場合は「-i」GUIオプションを使って実行します。

インストール・ファイルは以下のとおり実行されます。

- JEUSライセンスを契約します。インストールを実行するための必須事項なので注意深くお読みください。
- JEUSの構成ファイルおよびディレクトリーをデプロイします。
- JEUSのショートカット・アイコンを作成します。(Windows環境)
- JEUSの環境変数を設定します。
- JEUSのアンインストール・ツールを作成します。

4.2. システム要件

以下は、JEUSをインストールするためのシステム(H/W、S/W)要件です。

- システム要件

JEUSをインストールするために必要なハードウェアとソフトウェアの要件は以下のとおりです。

[表 4.1] システム要件

プラットフォーム	インストールに必要な環境
Windows 2008	JDK 7、JDK 8
Windows 2012	2GB以上のハードディスクの空き容量
Windows Vista	
Windows 7	

プラットフォーム	インストールに必要な環境
Windows 8	
Solaris, HP-UX, AIX, Linux	JDK 7、JDK 8 2GB以上のハードディスクの空き容量

- プラットフォーム対応環境

JEUSの動作に必要なプラットフォーム別の対応環境は以下のとおりです。

[表 4.2] プラットフォーム対応環境

OSバージョン	CPU	RAMメモリー	ハードディスク容量	JDKバージョン
Solaris 9, 10, 11	UltraSPARC 9, 10, 11 Intel X86 Series 10	1GB	20GB	JDK 7、 JDK 8
HP-UX 11.x, 11i, 11iV2	PA-RISC 11.x(11.11) Intel Itanium64 11.x(11.23, 11.31)	1GB	20GB	JDK 7、 JDK 8
IBM AIX 5L, 6L, AIX 7L	RS6000 IBM pSeries(PowerPC)	1GB	20GB	JDK 7、 JDK 8
Linux系 (Kernel 2.6以上)	Intel x86 series k2.6以上(k2.4に対応) Intel Itanium Series k2.6以上 IBM pSeries(PowerPC) k2.6以上	1GB	20GB	JDK 7、 JDK 8
Windows 2008, 2012, Vista, 7, 8	Intel x86 Series 2008, 2012, Vista, 7, 8	1GB	20GB	JDK 7、 JDK 8

4.3. インストール手順

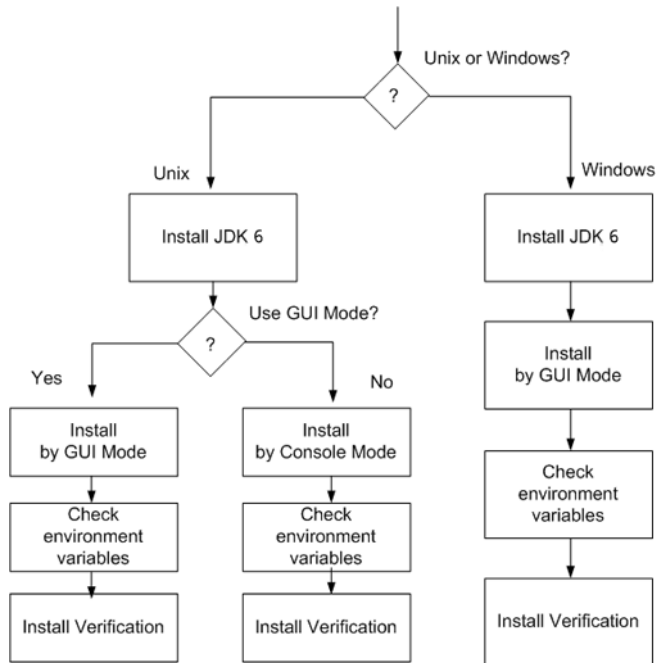
以下は、UNIXとWindows環境におけるJEUSのインストール手順です。全般的なインストール手順は同様ですが、一部異なる手順がありますので、詳細内容については各節でご確認ください。

1. JDK 7またはJDK 8をインストールします。本書ではJDK 7を基準に説明します。
2. GUIモードでJEUSをインストール(ファイルをコピー)します。
3. 環境変数を設定し、検証します。

4. インストールされた内容を確認(動作を確認)します。

以下は、JEUSのインストール手順と選択事項を示しています。

[図 4.1] JEUSのインストール手順



上記の図についての詳細内容は、各節の説明を参照してください。

- UNIX環境でGUIモードを利用してJEUSをインストールするときは、「[5.2.1. GUIモードでのインストール](#)」を参照してください。
- UNIX環境でコンソール・モードを利用してJEUSをインストールおよびアンインストールするときは、「[5.2.2. コンソール・モードでのインストール](#)」, 「[5.4.1. GUIモードでのアンインストール](#)」を参照してください。
- Windows環境でGUIインストール・ツールを利用してJEUSをインストールおよびアンインストールするときは、「[第6章 Windows環境でのインストールおよびアンインストール](#)」を参照してください。
- 「[付録 A. 各プラットフォームのJDKインストール](#)」は、Linux、Solaris、AIX、HP-UXなどの特定したプラットフォームにおける追加的なインストール情報を提供します。

4.4. JDKのインストール

JEUSを使用するには、JDK(Java SE Development Kit)がインストールされている必要があります。

以下より、JDK 7をダウンロードできます。

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html>

詳しい内容は、<http://www.oracle.com/technetwork/java/javase/install-142943.html>を参照してください。JDKインストールについての詳細内容は、「付録 A. 各プラットフォームのJDKインストール」を参照してください。

JEUSをインストールする前にJDKがインストールされているか確認します。インストール後には、Windows環境変数のPATHにインストールされたJDKのbinディレクトリーを追加します。

4.5. JEUSのライセンス・ポリシー

JEUSを使用するにはTmaxSoftが発行するLicense-Keyファイルが必要となります。インストーラーを利用してインストールしたJEUSには、基本的にトライアル・ライセンス(Trial License)が含まれています。ライセンス・ファイルは、JEUS_HOME/licenseに「license」というファイル名で存在します。

ライセンスのエディションには、StandardとEnterpriseがあり、機能および有効期限はそれぞれ異なります。

ライセンス・ファイルを使用するためには、JEUS_HOME/licenseディレクトリーの下位にLicense-Keyファイルを「license」という名前でコピーします。なお、ライセンスのアップグレードや正規ライセンスの購入については、弊社の営業部までお問合せください。

参考

ライセンス・ファイル名が「license」でない場合は「license」に変更してください。

第5章 UNIX環境でのインストールおよびアンインストール

本章では、UNIX環境でJEUSをインストールおよびアンインストールする手順について説明します。

5.1. 概要

UNIX 環境でのJEUSのインストールとアンインストールは、GUIモードまたはコンソール・モードの選択が可能です。

JEUSのインストールは以下の手順で行われます。

1. JDK 7をインストール
2. JEUSのインストール(ファイル・コピー)
3. インストールの確認
 - 起動の確認
 - 環境変数の設定と確認

5.2. インストール

本節では、GUIモードとコンソール・モードでのインストール手順について説明します。

5.2.1. GUIモードでのインストール

UNIX環境でGUIモードを利用してJEUSをインストールする手順について説明します。GUIでのインストールができない場合は、[「5.2.2. コンソール・モードでのインストール」](#)を参照してください。

参考

Linux、Solaris、HP、AIXにおけるJEUSのインストール手順は同様であり、インストール時の画面構成はWindows環境でのインストールと類似しています。一方、UNIX環境でGUIインストール・ツールを使用するには、JavaベースのGUI環境をサポートするコンソール、あるいはターミナルがインストールされている必要があります。

GUIモードでのJEUSのインストール手順は以下のとおりです。

1. CDをCD-ROMドライブに挿入にマウントします。CDのjeus80_unix_generic.binファイルが存在するディレクトリーに移動します。
2. 別の場所に存在するインストーラーを実行するには、以下のようにjeus80_unix_generic.binファイルの実行権限を付与します。

```
chmod u+x jeus80_unix_generic.bin
```

コマンドラインで以下を入力した後、<ENTER>キーを押します。

```
jeus80_unix_generic.bin -i GUI
```

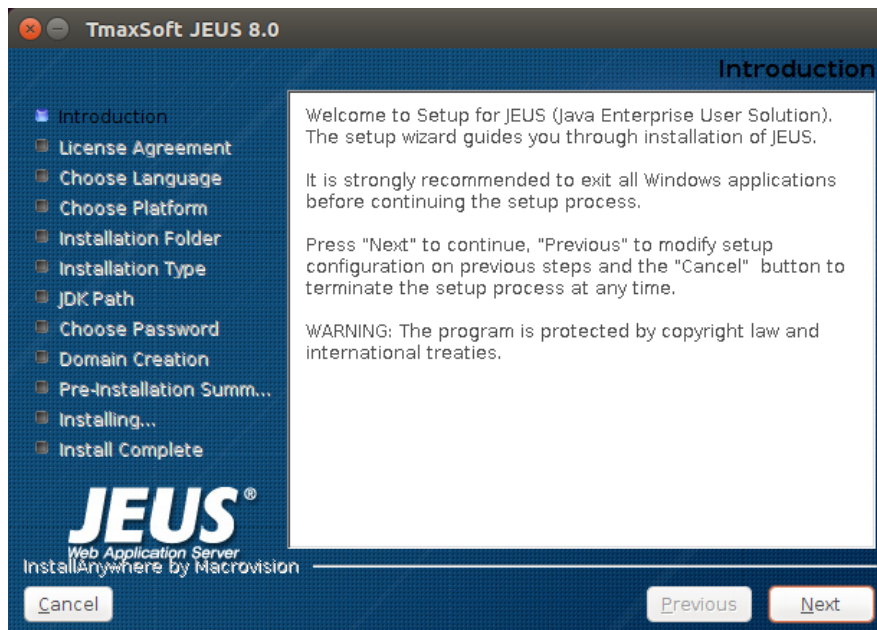
3. JEUSインストール・ツールの初期画面が表示されます。進行状況が100%になるまで待ちます。

[図 5.1] UNIX環境のGUIモードでのインストール - インストール・ツールの初期画面



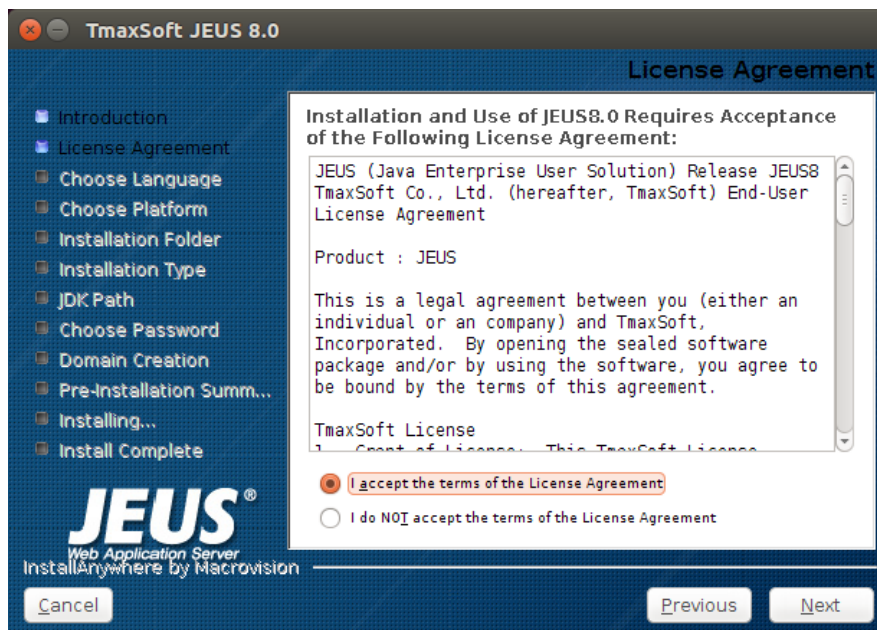
4. 以下の[Introduction]画面が表示されます。

【図 5.2】 UNIX環境のGUIモードでのインストール - Introduction



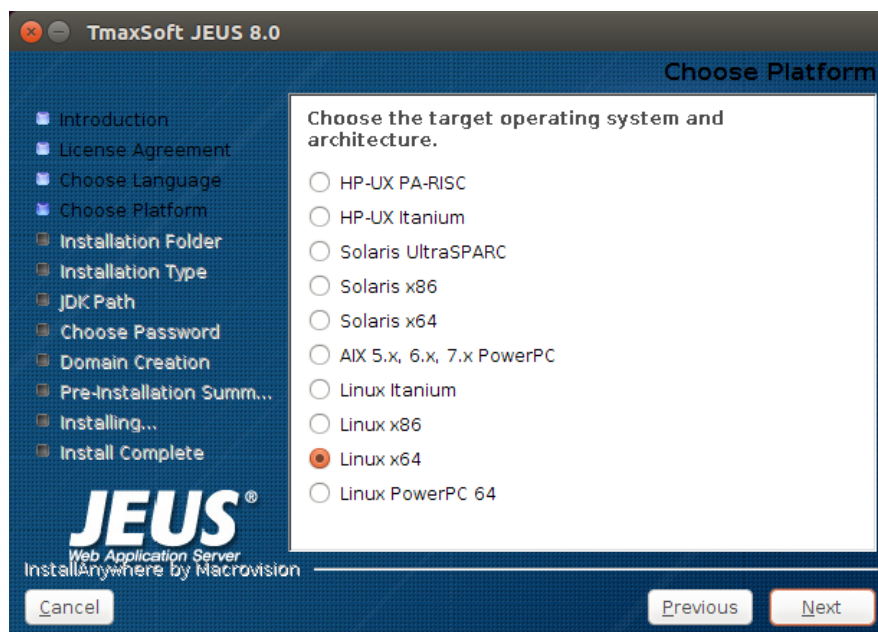
5. JEUSの紹介内容を確認し、[Next]ボタンをクリックします。インストールを中止したい場合は、[Cancel]ボタンをクリックします。
6. [License Agreement]画面で使用許諾契約の内容を確認し、「I accept the terms of ...」を選択してから[Next]ボタンをクリックします。

【図 5.3】 UNIX環境のGUIモードでのインストール - License Agreement



7. プラットフォームの一覧からインストールするプラットフォームを選択します。

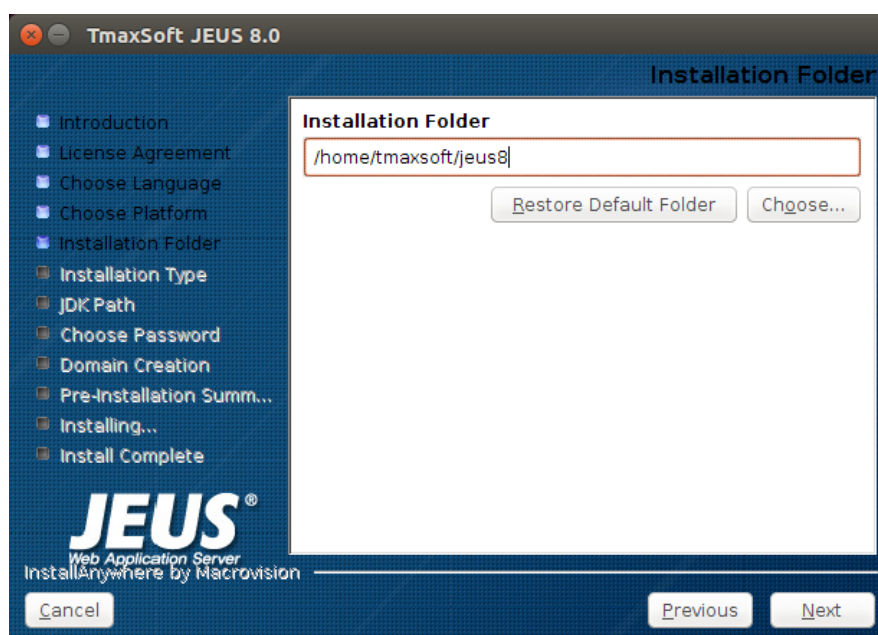
[図 5.4] UNIX環境のGUIモードでのインストール - Choose Platform



8. [Installation Folder]画面が表示されます。JEUSバイナリーとその他のファイルはインストールしたパスのディレクトリーの下位に格納されます。

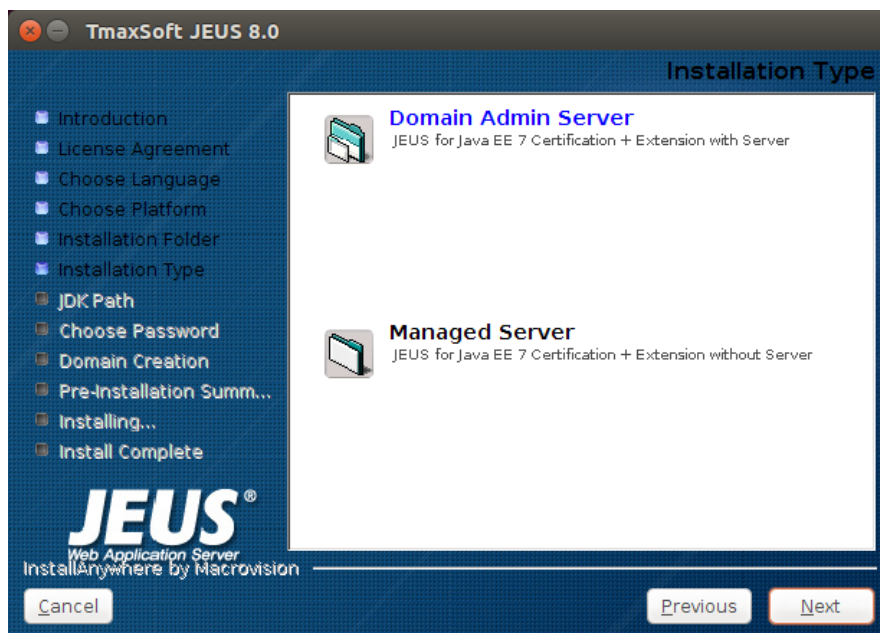
「Choose...」をクリックすると、別のインストール・フォルダーが選択できます。「Restore Default Folder」をクリックすると初期のデフォルト・フォルダーに戻ります。デフォルト・フォルダーの使用をお勧めします(例: /home/tmaxsoft/jeus8)。インストールするディレクトリーを選択し、[Next]ボタンをクリックします。

[図 5.5] UNIX環境のGUIモードでのインストール - Installation Folder



9. [Installation Type]画面が表示されます。インストール・オプションの「Domain Admin Server」と「Managed Server」から1つを選択し、[Next]ボタンをクリックします。(デフォルト: Domain Admin Server)

【図 5.6】 UNIX環境のGUIモードでのインストール - Installation Type



以下は、インストール・オプションについての説明です。

項目	説明
Domain Admin Server	Domain Admin Server(DAS)とManaged Server(MS)をインストールします。 Domain Admin ServerはManaged Serverを管理するためのサーバーです
Managed Server	Managed Serverをインストールします

- 10 Domain Admin Serverを選択した場合、Installation Modeを選択します。

【図 5.7】 UNIX環境のGUIモードでのインストール - Installation Mode



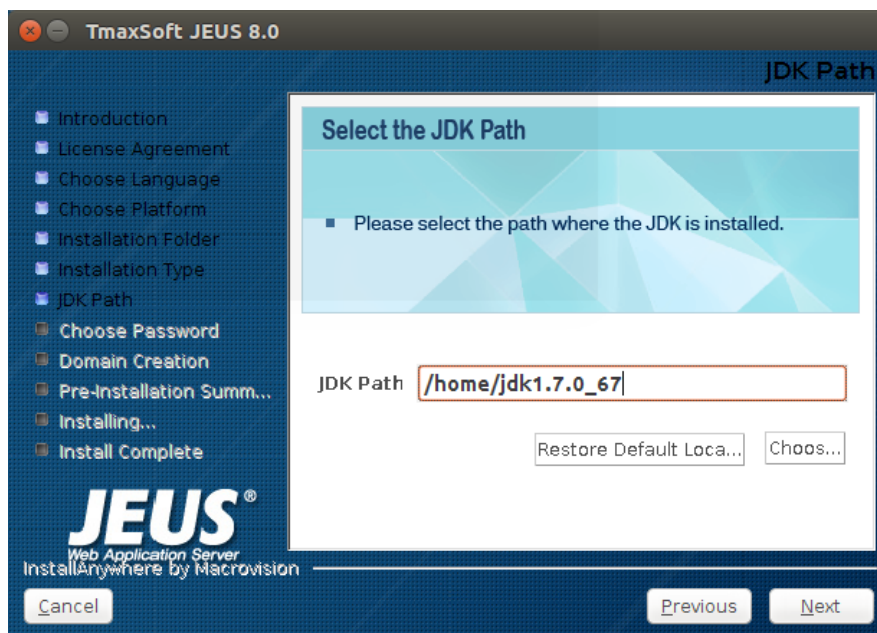
以下は、インストール・オプションについての説明です。

項目	説明
Production Mode	インストール時にプロダクション・モードでインストールします。JEUSホットスワップ、自動リロードを使用しません。デモライセンスが使用されると、警告メッセージが出力されます
Development Mode	インストール時にデプロイメント・モードでインストールします。JEUSホットスワップ、自動リロードを使用します

11. **[Choose JAVA Folder]**画面で初期にインストールされたJDKパス(例: /usr/jdk1.7)を設定します。

インストール・プログラムはインストールされたJDKパスをお勧めしますが、「**Choose...**」をクリックすると別のフォルダーが選択でき、「**Restore Default Folder**」をクリックするとデフォルト・フォルダーに戻ります。

【図 5.8】 UNIX環境のGUIモードでのインストール - Choose JAVA Folder

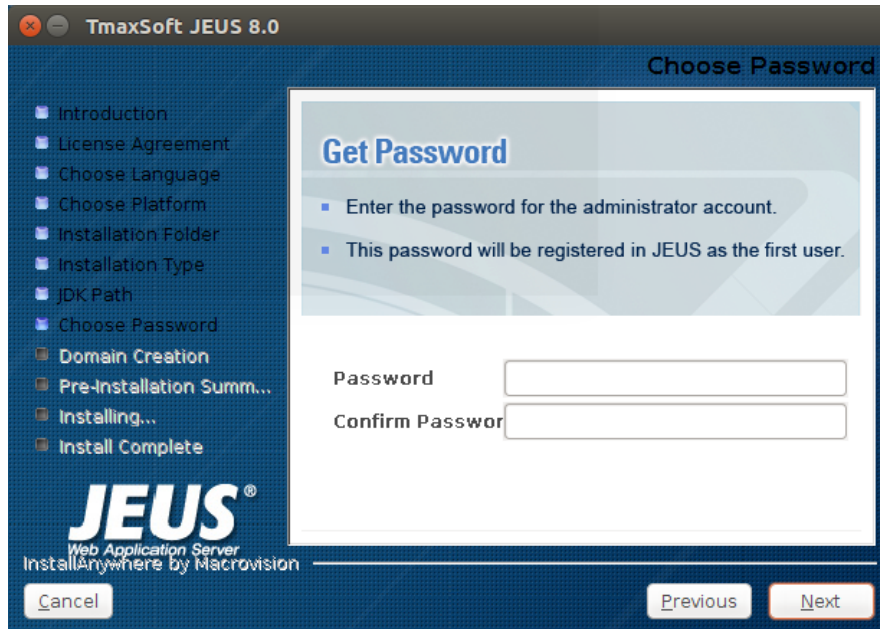


参考

JEUSのインストール後にJDKをインストールすると、JDKのインストール・パスを含むJAVA_HOME環境変数を修正する必要があるため、JDKをインストールした後にJEUSをインストールすることをお勧めします。

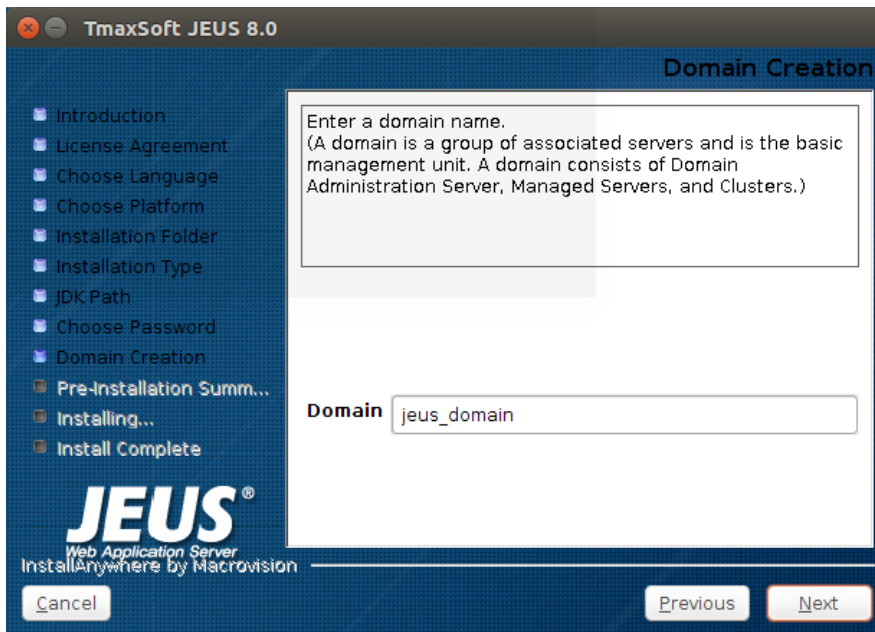
12 **[Choose Password]**画面でJEUS管理者のパスワードを入力し、**[Next]**ボタンをクリックします。パスワードは、「administrator」管理者アカウントで割り当てられます。

[図 5.9] UNIX環境のGUIモードでのインストール - Choose Password



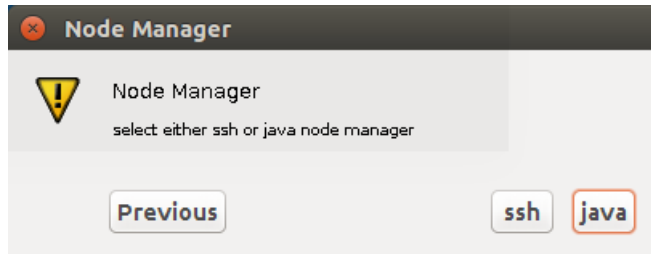
JEUSで使用するドメイン名を入力し、[Next]ボタンをクリックします。

[図 5.10] UNIX環境のGUIモードでのインストール - Domain Creation



- 14 Domain Admin Serverを選択した場合に使用するノード・マネージャを選択します。基本的にはJavaタイプのノード・マネージャが設定されます。

[図 5.11] UNIX環境のGUIモードでのインストール - ノード・マネージャの設定

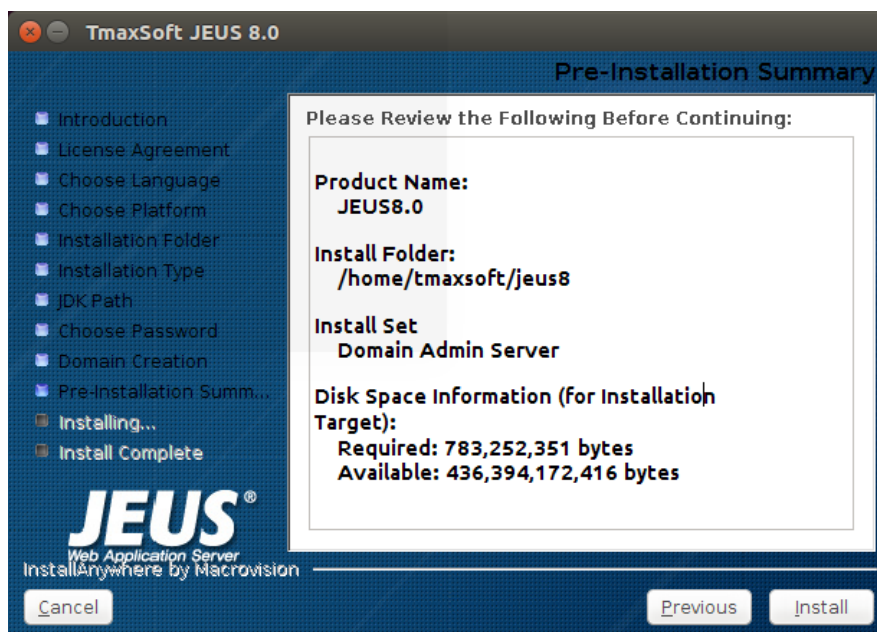


以下は、各インストール項目についての説明です。

項目	説明
java	Javaで実装されたノード・マネージャです。Javaタイプのノード・マネージャについての詳細内容は、『JEUS ノードマネージャガイド』の「第2章 Javaタイプのノード・マネージャ」を参照してください
ssh	OSで提供するSSHを使用するノード・マネージャです。SSHタイプのノード・マネージャについての詳細内容は、『JEUS ノードマネージャガイド』の「第3章 SSHタイプのノード・マネージャ」を参照してください。

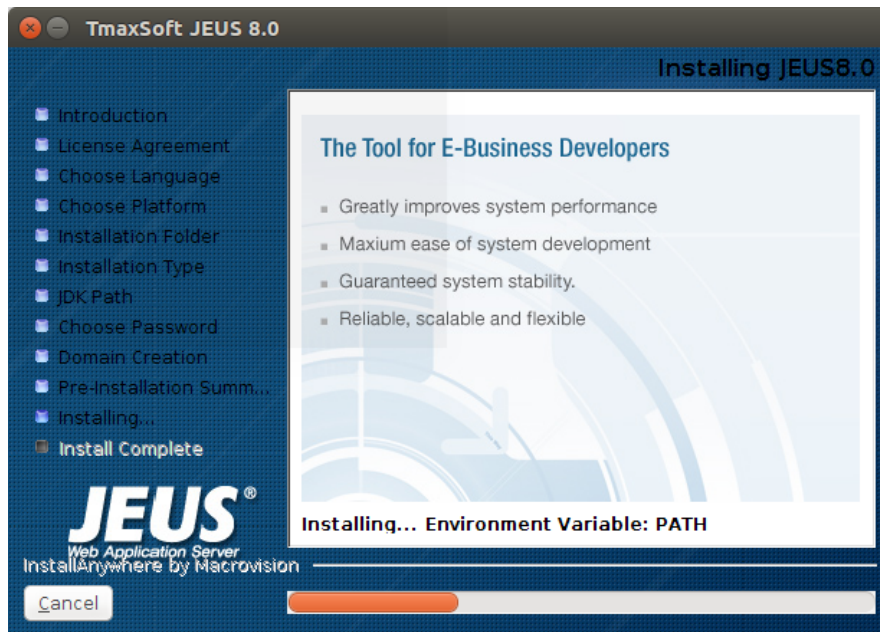
- 15 これまで設定したインストール情報が出力されます。インストール情報を確認した後、[Install]ボタンをクリックします。

[図 5.12] UNIX環境のGUIモードでのインストール - Pre Installation Summary



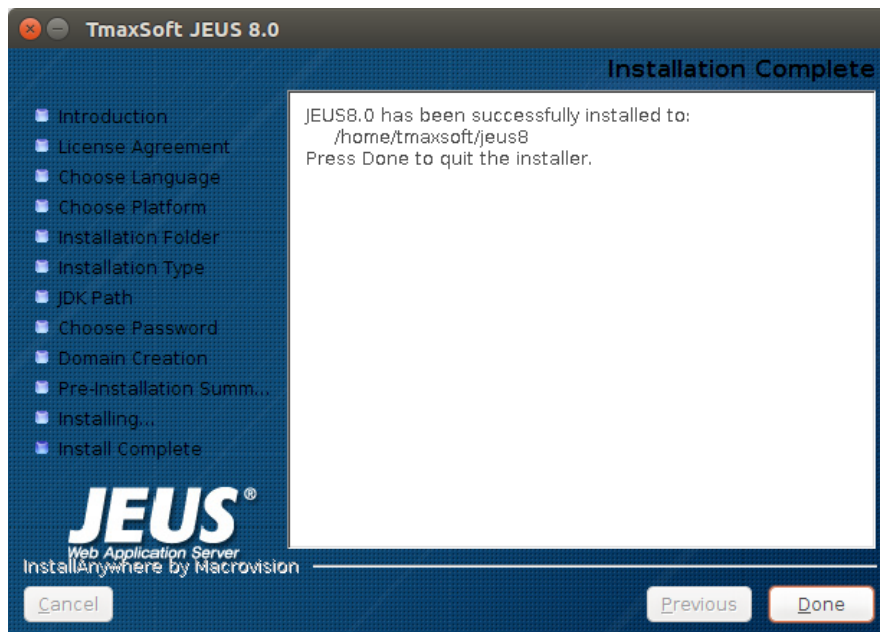
16 インストールの進行状況が確認できます。

[図 5.13] UNIX環境のGUIモードでのインストール - Installing JEUS



17. インストールの完了を確認してから、[Done]ボタンをクリックするとインストールが終了します。

[図 5.14] UNIX環境のGUIモードでのインストール - Install Complete



5.2.2. コンソール・モードでのインストール

本節では、コンソール・モード(コマンドライン)のUNIX/Linux環境において、JEUSをインストールする方法について説明します。コンソール・モードでインストールするためには、コンソール・インストーラーを実行します。

参考

コンソール・インストーラーは、インストール中に入力される値をデフォルト値に設定します。シェルは、前のフェーズに戻ることができないため、インストールが完了した後に変更してください。

以下は、テキスト・ベースのシェルを実行する手順についての説明です。

1. JEUSのインストールCDを挿入してマウントします。CDの`jeus80_unix_generic.bin`ファイルが存在するディレクトリーに移動します。
2. 別のディレクトリーに存在するインストール・コンソール・インストーラーを実行可能にするには、`jeus80_unix_generic.bin`ファイルの実行権限を以下のとおり与えます。

```
chmod u+x jeus80_unix_generic.bin
```

3. コンソールで`jeus80_unix_generic.bin`ファイルを実行するために、「./jeus80_unix_generic.bin」を入力した後、<ENTER>キーを押します。

```
[jeususer@matrix jeus]$ ./jeus80_unix_generic.bin
Preparing to install...
Extracting the installation resources from the installer archive...
Configuring the installer for this system's environment...

Launching installer...

Preparing CONSOLE Mode Installation...
```

4. JEUSの使用許諾契約画面が表示されます。

```
=====
License Agreement
-----

Installation and Use of JEUS8.0 Requires Acceptance of the Following License
Agreement:

JEUS (Java Enterprise User Solution) Release JEUS8
TmaxSoft Co., Ltd. (hereafter, TmaxSoft) End-User License Agreement

Product : JEUS
```


This is a legal agreement between you (either an individual or an company) and TmaxSoft, Incorporated. By opening the sealed software package and/or by using the software, you agree to be bound by the terms of this agreement.

TmaxSoft License

1. Grant of License: This TmaxSoft License Agreement ("License") permits you to use one copy of the TmaxSoft product JEUS, on any single computer, provided the software is in use on only one computer at any one time. If this package is a license pack, you may make and use additional copies of the software up to the number of licensed copies authorized. If you have multiple licenses for the software, then at any time you may have as many copies of the software in use as you have licenses.

The software is "in use" on a computer when it is loaded into the temporary memory (i.e., RAM) or installed into the permanent memory (e.g., hard disk, CD-ROM, or other storage devices) of that computer, except that a copy installed on a network server for the sole purpose of distribution to other computers is not "in use". If the anticipated number of users of the software will exceed the number of applicable licenses, then you must have a reasonable

5. <ENTER>キーを押して、引き続きJEUSの使用許諾契約の内容を確認してください。

PRESS <ENTER> TO CONTINUE:

mechanism or process in place to ensure that the number of persons using the software concurrently does not exceed the number of licenses.

2. Copyright: The software (including any images, "applets," photographs, animations, video, audio, music and text incorporated into the software) is owned by TmaxSoft or its suppliers and international treaty provisions. Therefore, you must treat the software like any other copyrighted materials (e.g., a book or musical recording) except that you may either (a) make one copy of the software solely for backup or archival purposes, or (b) transfer the software to a single hard disk provided you keep the original solely for backup or archival purposes. You may not copy the printed materials accompanying the software, nor print copies of any user documentation provided in "online" or electronic form.

3. Other restrictions: This license is your proof of license to exercise the rights granted herein and must be retained by you. You may not rent, lease, or transfer your rights under this license on a permanent basis provided you transfer this license, the software, and all accompanying printed materials, retain no copies, and the recipient agrees to the terms of this license. You may not reverse engine, decompile, or disassemble the software, except to the extent that the foregoing restriction is expressly prohibited by applicable law.

6. 使用許諾契約の内容に同意するか否かを答えます。

同意する場合は「y」を同意しない場合は「n」を入力してから<ENTER>キーを押します。

```
PRESS <ENTER> TO CONTINUE:
```

```
DO YOU ACCEPT THE TERMS OF THIS LICENSE AGREEMENT? (Y/N): y
```

7. プラットフォームの一覧からインストールするプラットフォームを決めて入力します。

```
=====
Choose Platform
-----

Choose the operating system and architecture :
1)HP-UX PA-RISC
2)HP-UX Itanium
3)Solaris UltraSPARC
4)Solaris x86
5)Solaris x64
6)AIX 5.x, 6.x, 7.x PowerPC
7)Linux Itanium
8)Linux x86
9)Linux x64
10)Linux PowerPC 64
Quit) Quit Installer

Choose Current System (DEFAULT: 8):
```

8. インストール・ディレクトリーを選択します。

デフォルト値を使用する場合は<ENTER>キーを押し、変更する場合はインストール・パスを入力します。

```
=====
Installation Folder
-----

Enter the installation folder.

Default Install Folder: /home/jeus8

ENTER AN ABSOLUTE PATH, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
:
```

9. Domain Admin Server(DAS)とManaged Server(MS)のうち、インストールするサーバーを選択します。
デフォルトではDomain Admin Serverが設定されます。

```

=====
Installation type
-----

Please choose the Install Set to be installed by this installer.

->1- Domain Admin Server
    2- Managed Server

ENTER THE NUMBER FOR THE INSTALL SET, OR PRESS <ENTER> TO ACCEPT THE DEFAULT
:

```

以下は、各インストール項目についての説明です。

項目	説明
Domain Admin Server	Domain Admin Server(DAS)とManaged Server(MS)をインストールします。 Domain Admin ServerはManaged Serverを管理するためのサーバーです
Managed Server	Managed Serverをインストールします

10 Domain Admin Serverを選択した場合、Installation Modeを選択します。基本的にはProduction Modeが設定されます。

```

=====
Installation Mode
-----

* Production Mode
- Disables JEUS Hot Swap.
- Disables Automatic Reloading.
- Displays a warning message and recommends using a full license if a demo
  license is used.

* Development Mode
- Enables JEUS Hot Swap.
- Enables Automatic Reloading.

->1- Production Mode
    2- Development Mode
    3- Cancel

ENTER THE NUMBER OF THE DESIRED CHOICE, OR PRESS <ENTER> TO ACCEPT THE
DEFAULT:

```

以下は、各インストール項目についての説明です。

項目	説明
Production Mode	インストール時にプロダクション・モードでインストールします。JEUSホットスワップ、自動リロードを使用しません。デモライセンスが使用されると、警告メッセージが出力されます
Development Mode	インストール時にデプロイメント・モードでインストールします。JEUSホットスワップ、自動リロードを使用します

11. JDKの位置を入力します。<ENTER>キーを押すと、自動で認識したJDKの位置を使用します。ディレクトリを変更する場合はインストール・パスを入力します。

```
=====
Enter the JDK path.
-----

Enter the JDK path:

Enter the JDK path (DEFAULT: /usr/jdk1.7):
```

- 12 Domain Admin Serverを選択した場合、JEUS管理者のパスワードを入力します。同パスワードは「administrator」IDで割り当てられます。

```
=====
Password Input
-----

Enter the Password for the administrator account.

Input Password::
```

参考

パスワードはJEUSを起動するための必須要素です。入力したパスワードは記憶装置に記録されるので慎重に入力します。

- 13 Domain Admin Serverを選択した場合、Domain Admin Serverに使用するドメイン名を入力します。

```
=====
User Domain name
-----

Enter the Domain name

Enter the domain name (DEFAULT: jeus_domain):
```

- 14 Domain Admin Serverを選択した場合に使用するノード・マネージャを選択します。基本的にはJavaタイプのノード・マネージャが設定されます。

```
=====
Node Manager
-----

select either ssh or java node manager

->1- java
   2- ssh
   3- Cancel

ENTER THE NUMBER OF THE DESIRED CHOICE, OR PRESS <ENTER> TO ACCEPT THE
DEFAULT:
```

以下は、各インストール項目についての説明です。

java	Javaで実行されたノード・マネージャです。Javaタイプのノード・マネージャについての詳細説明は、『 <i>JEUS ノードマネージャガイド</i> 』の「第2章 Javaタイプのノード・マネージャ」を参照してください
ssh	OSで提供するSSHを使用するノード・マネージャです。SSHタイプのノード・マネージャについての詳細内容は、『 <i>JEUS ノードマネージャガイド</i> 』の「第3章 SSHタイプのノード・マネージャ」を参照してください

- 15 以下の画面はインストール関連の要約情報です。<ENTER>キーを押すとインストールが開始されます。

```
=====
Pre-Installation Summary
-----

Review the Following Before Continuing:

Product Name:
    JEUS8.0

Install Folder:
    /home/jeus

Install Set
    Domain Admin Server

Disk Space Information (for Installation Target):
    Required:  441,732,614 bytes
    Available: 156,301,557,760 bytes
```

```
PRESS <ENTER> TO CONTINUE:
```

16. プログレスバーが表示され、インストールが行われます。

```
=====
Installing...
-----

[=====|=====|=====|=====]
[-----|-----|-----|-----]
```

17. インストールが完了します。

```
=====
Installation Complete
-----

JEUS8.0 has been successfully installed to:

    /home/jeus

PRESS <ENTER> TO EXIT THE INSTALLER:
```

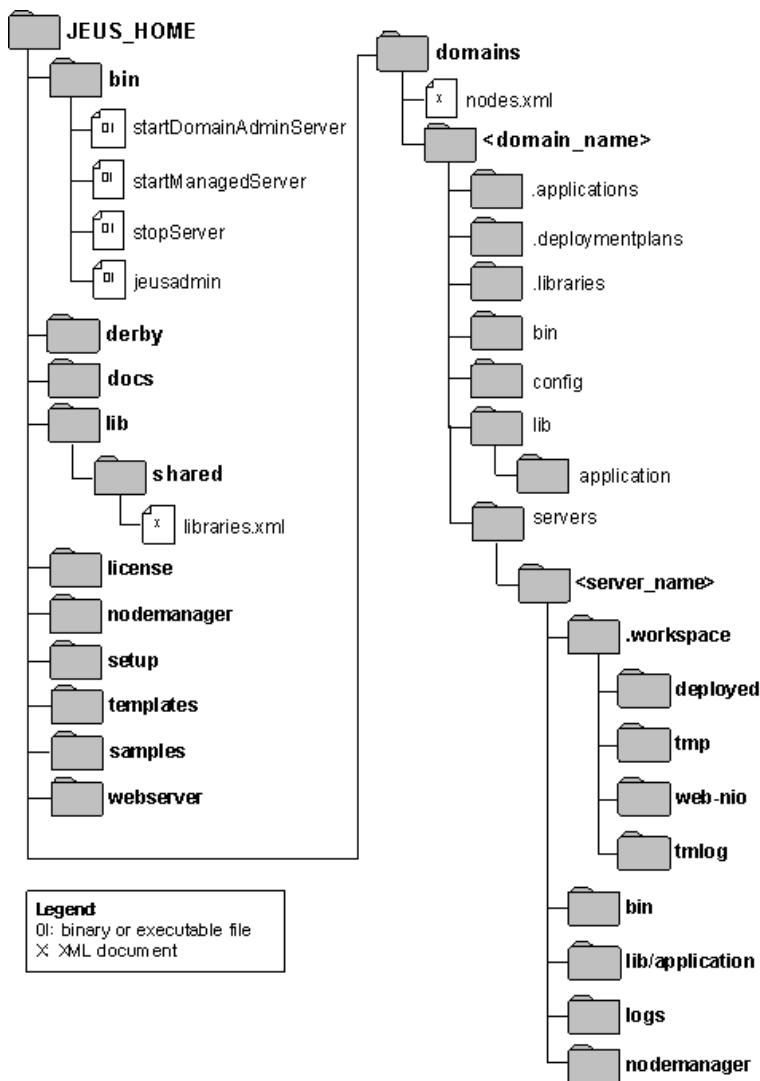
5.3. インストールの確認

GUIモードまたはコンソール・モードを使用してJEUSをインストールした後、使用に必要な環境変数を設定します。環境変数を設定した後、JEUSを起動して正常にインストールされているのかを確認します。環境変数を設定する前にJEUSがインストールされたパスのディレクトリー構造を確認します。

5.3.1. ディレクトリー構造

以下は、JEUSをインストールしたときのディレクトリー構造です。

[図 5.15] JEUSインストール後のディレクトリー構造



以下は、ディレクトリーとファイルについての説明です。

{JEUS_HOME}

JEUSのルート・ディレクトリーとして、実際のディレクトリー名と場所はインストール時に決まります。

bin

サーバーの起動および終了スクリプトであるstartDomainAdminServer、startManagedServer、stopServerとJEUSコンソール・ツール(jeusadmin)のような実行ファイルが存在します。

derby

サンプル・アプリケーションの作成やテスト環境で容易に使用できるようにApache Derbyが含まれています。

docs

JEUSが提供するAPIのJavadocが存在します。

lib

JEUSの起動に必要なライブラリーが存在します。ユーザーは、sharedディレクトリー以外のディレクトリーにはアクセスする必要がありません。

ディレクトリー	説明
shared	sharedディレクトリーには、アプリケーションで使用するライブラリーが存在します。 sharedディレクトリーのライブラリーを使用するには、libraries.xmlにライブラリーの情報を追加する必要があります。なお、当該ライブラリーを使用するアプリケーションのJEUS Deployment Descriptor(DD)で当該ライブラリーについてのリファレンス情報を指定します。sharedライブラリーの詳細内容は、『 <i>JEUS アプリケーション&デプロイメントガイド</i> 』の「3.3.2. 共有ライブラリー」を参照してください

license

JEUSのライセンス・ファイルが存在します。ライセンス・ファイルは、JEUSが実行されるための必須ファイルです。

nodemanager

JEUSのノード・マネージャーのための設定ファイルであるjeusnm.xmlファイルが存在します。

setup

JEUSのインストール後、使用できる環境を構築するための必要ファイルが存在します。

templates

各種設定や環境などのテンプレート・ファイルが存在します。

samples

JEUSのサンプル・ファイルが存在します。

webserver

JEUSのインストール時にJEUS Webサーバーがインストールされるディレクトリーです。詳しい内容は、『JEUS Webエンジンガイド』を参照してください。

domains

各ドメインの下位にDOMAIN_HOMEとJEUS_HOMEで使用するノード情報が含まれているnodes.xmlが存在します。

以下のディレクトリーおよびファイルは、DOMAIN_HOMEの下位に存在します。

- .applications

各ドメインで管理するアプリケーション・ファイルが存在します。

追加および削除には、**install-application**、**uninstall-application**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.7. install-application」、『*JEUS リファレンスガイド*』の「4.2.6.16. uninstall-application」を参照してください。

- .deploymentplans

ドメインで管理するデプロイメント・プラン・ファイルが存在します。

追加および削除には、**install-deployment-plan**、**uninstall-deployment-plan**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.8. install-deployment-plan」、『*JEUS リファレンスガイド*』の「4.2.6.17. uninstall-deployment-plan」を参照してください。

- .libraries

ドメインで管理するライブラリー・ファイルが存在します。

追加および削除には、**install-application**、**uninstall-application**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.7. install-application」、『*JEUS リファレンスガイド*』の「4.2.6.16. uninstall-application」を参照してください。

- bin

各ドメインに属しているDASとMSの開始および終了スクリプトが存在します。JEUS_HOME/binのstartDomainAdminServer、startManagedServer、stopServerと同様な機能を実行しますが、ドメイン名を設定する必要はありません。

- config

ドメインの設定ファイルであるdomain.xmlが変更された場合、以前のヒストリーのために存在するバックアップ・ファイルが格納されます。ドメイン設定についての詳細説明は、『*JEUSドメインガイド*』の「第3章 ドメインの設定変更」を参照してください。

区分	説明
security	<ul style="list-style-type: none"> – SYSTEM_DOMAIN : ドメイン単位で適用されるセキュリティー・ドメイン・ファイルのaccounts.xml、policies.xmlが存在しており、各XMLファイルはWebAdminとjeusadminを利用して動的な設定変更が可能です。セキュリティー・ドメイン設定についての詳細説明は、『<i>JEUS セキュリティガイド</i>』の「2.2. セキュリティー・ドメインの定義」を参照してください – security.key : 対称鍵暗号化アルゴリズムのキーを保存するファイルです。JEUS_HOME/bin/encryptionを実行すると作成されます。security.keyファイルについての詳細説明は、『<i>JEUS セキュリティガイド</i>』の「2.5.4. パスワードのセキュリティー設定」を参照してください – policy : Javaパーミッション設定ファイルです。JEUSのセキュリティー・システムとは別途にJava SEセキュリティー・マネージャーで使われます
servlet	<ul style="list-style-type: none"> – web.xml : web.xmlを持っていない場合、Webエンジンが使用するWebモジュールのweb.xmlです。デフォルト値は、空きのXMLファイルです – webcommon.xml: ドメイン内サーバーのWebエンジンの全Webモジュールに適用される共通設定ファイルです。設定についての詳細内容は、『<i>JEUS Webエンジンガイド</i>』の「1.5. ディレクトリー構造」を参照してください

- lib/application

全ドメインに適用したいアプリケーション・ライブラリーを格納するディレクトリーです。

SERVER_HOMEに存在するアプリケーション・ライブラリーと衝突した場合、SERVER_HOME/lib/applicationが優先され、警告メッセージが残ります。lib/applicationディレクトリーについての詳細説明は、『*JEUS アプリケーション&デプロイメントガイド*』の「3.3.1. lib/applicationディレクトリー」を参照してください。

- servers

このディレクトリーの下位にSERVER_HOMEディレクトリーがサーバー名で作成されます。SERVER_HOMEディレクトリー構造の詳細説明は、『*JEUS サーバガイド*』の「1.5. サーバー・ディレクトリー構造」を参照してください。

ディレクトリー	説明
.workspace	JEUSが使用するサーバー別のスペースなので、ユーザーが変更してはなりません
bin	サーバーの開始/終了スクリプトが含まれています。JEUS_HOME/binのスクリプトと同様な機能を行いますが、ドメイン名とサーバー名を設定する必要はありません

ディレクトリー	説明
	<ul style="list-style-type: none"> – DASの場合：startDomainAdminServer/stopServerが存在します – MSの場合：startManagedServer/stopserverが存在します
lib/application	<p>サーバーに適用したいアプリケーション・ライブラリーが存在します。ドメイン領域のライブラリー(DOMAIN_HOME/lib/application)より優先順位が高いです。ライブラリーが衝突した場合、同ディレクトリーに存在するファイルが適用され、警告メッセージが残ります。</p> <p>lib/applicationの詳細内容は、『<i>JEUS アプリケーション&デプロイメントガイド</i>』の「3.3.1. lib/applicationディレクトリー」を参照してください</p>
logs	サーバーのランチャー(Launcher)ログ、サーバー・ログ、アクセス・ログファイルが残ります。詳細内容は、『 <i>JEUS サーバガイド</i> 』の「第8章 ロギング」を参照してください
nodemanager	ノード・マネージャーが再起動する際、管理していたサーバーなのかどうかを判断するための情報を保存します。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。

5.3.2. 環境設定

JEUSを使用するには環境変数が必要となります。インストールの手順で一部の環境変数を設定しますが、場合によっては修正して使用します。インストール時に、環境変数PATHは「**.profile/.cshrc**」に適用され、以外の環境変数は、「**\$JEUS_HOME/bin/jeus.properties**」ファイルに設定されます。

以下は、主な環境変数についての説明です。

[表 5.1] 環境変数の設定

環境変数	説明
PATH	<p>システム・パスを設定します。</p> <p>以下が含まれている必要があります</p> <ul style="list-style-type: none"> – /home/jeus/bin – /home/jeus/lib/system
JEUS_HOME	JEUSのインストール・ディレクトリーを設定します(例: /home/jeus)
JEUS_LIBPATH	JEUSのライブラリー・ファイルのパスを設定します(例: /home/jeus/lib/system)
VM_TYPE	Java HotSpot JVMの使用有無を設定します(例: hotspot or old)
USERNAME	管理者アカウントのIDを設定します

環境変数	説明
PASSWORD	管理者パスワードを設定します
JAVA_HOME	JDK のインストール・ディレクトリー・パスを設定します (例: /usr/jdk1.7)
JAVA_ARGS	JDKのパラメータを設定します
JAVA_VENDOR	JDKのベンダーを設定します (例: Sun, IBM, HP)

環境変数を設定するときにCシェルの場合は、「setenv」を使ってJEUS_HOMEの変数を設定します。

```
setenv JEUS_HOME "/home/jeus"
```

以下は、システムPATHを設定する例です。

```
setenv PATH "${PATH}:/home/jeus/bin:
/home/jeus/lib/system"
```

参考

Javaの実行ディレクトリー(/usr/jdk1.7/bin)はJEUSで利用されるため、環境変数に追加するときは前方に追加します。

上記の内容を確認した後、環境を適用するためにログオフしてから、再ログインします。

5.3.3. 起動の確認

JEUSが正常にインストールされたのかを確認するため、以下の手順でJEUSを起動します。

1. コンソール・プロンプトに「**startDomainAdminServer**」を入力すると、以下のメッセージが出力されます。「-p」の次にインストール時に入力したパスワードを入力します。

```
[/home/jeus]$ startDomainAdminServer -u administrator -p <password>
*****
- JEUS Home           : /home/jeus
- Java Vendor         : Sun
- Added Java Option  :
*****
+ /opt/jdk1.7.0_21/jre/..bin/java -server -Xmx512m -Xbootclasspath/p:
/home/michael/jeus8_b27/lib/system/extension.jar -classpath
/home/michael/jeus8_b27/lib/system/bootstrap.jar
-Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.library.path=/home/michael/jeus8_b27/lib/system
-Djava.endorsed.dirs=/home/michael/jeus8_b27/lib/endorsed
-Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djava.net.preferIPv4Stack=true
-Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djava.util.logging.config.file=/home/michael/jeus8_b27/bin/logging.properties
```

```

-Djeus.home=/home/michael/jeus8_b27
-Djeus.jvm.version=hotspot -Djeus.tm.checkReg=true
-Djeus.properties.replicate=jeus,sun.rmi,java.util,java.net jeus.server.LauncherBootstrapper
-u administrator -p <password>

===== JEUS LICENSE INFORMATION =====
=== VERSION : JEUS 8.0 (Fix#0) (8.0.0.0-b27)
=== EDITION: Enterprise (Trial License)
=== NOTICE: This license restricts the number of allowed clients.
=== Max. Number of Clients: 5
=====

[2016.08.24 14:05:32][2] [launcher-1] [Launcher-0012] Starting the server [server1] with
the command
/opt/jdk1.7.0_21/jre/./bin/java -Dserver1 -Xmx512m -XX:MaxPermSize=256m
-Djeus.ejb.enable.configDeleteOption=true -server
-Xbootclasspath/p:/home/michael/jeus8_b27/lib/system/extension.jar -classpath
/home/michael/jeus8_b27/lib/system/bootstrap.jar
-Djava.security.policy=/home/michael/jeus8_b27/domains/jeus8/config/security/policy
-Djava.library.path=/home/michael/jeus8_b27/lib/system
-Djava.endorsed.dirs=/home/michael/jeus8_b27/lib/endorsed
-Djeus.properties.replicate=jeus,sun.rmi,java.util,java.net -Djeus.jvm.version=hotspot
-Djava.util.logging.config.file=/home/michael/jeus8_b27/bin/logging.properties
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djeus.home=/home/michael/jeus8_b27 -Djava.net.preferIPv4Stack=true
-Djeus.tm.checkReg=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Djeus.domain.name=jeus8
-Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djeus.server.protectmode=false
-XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput
-XX:LogFile=/home/michael/jeus8_b27/domains/jeus8/servers/server1/logs/jvm.log
jeus.server.admin.DomainAdminServerBootstrapper
-u administrator -domain jeus8 -server server1 .
[2016.08.24 14:05:32][2] [launcher-1] [Launcher-0014] The server[server1] is being started
...
[2016.08.24 14:05:36][2] [server1-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 14:05:36][0] [server1-1] [SERVER-0000] Version information - JEUS 8.0 (Fix#0)
(8.0.0.0-b27).

... 中略

[2016.08.24 14:06:21][2] [server-1] [SERVER-0248] The JEUS server is STANDBY.
[2016.08.24 14:06:21][2] [server-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 14:06:21][2] [server-1] [WEB-3413] The web engine is ready to receive requests.
[2016.08.24 14:06:21][2] [server-1] [UNIFY-0100] Listener information
BASE (plain, 0.0.0.0 : 9736) - VIRTUAL - SecurityServer
- FileTransfer
- BootTimeFileTransfer
- ClassFTP
- JNDI
- JMXConnectionServer/JeusMBeanServer
- JMXConnectionServer/JEUSMP_adminServer
- GMS-NetworkManager
- TransactionManager

```

```

- HTTP Listener
http-server (plain, 0.0.0.0 : 8808) - VIRTUAL
- HTTP Listener
jms-internal (plain, 0.0.0.0 : 9941) - VIRTUAL - JMSServiceChannel-internal

[2016.08.24 14:06:21][0] [server-1] [SERVER-0242] Successfully started the server.
[2016.08.24 14:06:21][2] [server-1] [SERVER-0248] The JEUS server is RUNNING.
[2016.08.24 14:06:21][2] [server-1] [SERVER-0401] The elapsed time to start: 10337ms.
[2016.08.24 14:06:21][2] [launcher-10] [Launcher-0034] The server[adminServer]
initialization completed successfully[pid : 21031].
[2016.08.24 14:06:21][0] [launcher-1] [Launcher-0040] Successfully started the server. The
server state is now RUNNING.

```

上記は、JEUS DASが正常に起動される過程を示しており、起動が完了すると「Successfully started the server. The server state is now RUNNING.」というメッセージが出力されます。

参考

1. 「Invalid License」メッセージはライセンスの異常を意味します。TmaxSoftからライセンスを取得し、\$JEUS_HOME/licenseディレクトリーにコピーします。
 2. 全フェーズの進行および環境変数が正常に設定されているのかを確認します。特に、「/jeus/bin」ディレクトリーがシステム・パス(「startDomainAdminServer」スクリプトが実行されるよう)に存在しているのかを確認します。
-

2. コンソール・プロンプトに**startManagedServer**を入力すると以下のメッセージが出力されます。「-p」の次にはインストール時に入力したパスワードを入力します。

```

[/home/jeus]$ startManagedServer -domain jeus8 -server server2 -u administrator -p
<password>
*****
- JEUS Home      : /home/jeus
- JEUS Base Port :
- Java Vendor    : Sun
- Added Java Option :
*****
+ /opt/jdk1.7.0_21/jre/./bin/java -server -Xmx512m -Xbootclasspath/p:
/home/michael/jeus8_b27/lib/system/extension.jar -classpath
/home/michael/jeus8_b27/lib/system/bootstrap.jar
-Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.library.path=/home/michael/jeus8_b27/lib/system
-Djava.endorsed.dirs=/home/michael/jeus8_b27/lib/endorsed
-Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djava.net.preferIPv4Stack=true
-Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djava.util.logging.config.file=/home/michael/jeus8_b27/bin/logging.properties
-Djeus.home=/home/michael/jeus8_b27 -Djeus.jvm.version=hotspot -Djeus.tm.checkReg=true
-Djeus.properties.replicate=jeus,sun.rmi,java.util,java.net
jeus.server.ManagedServerLauncherBootstrapper
-domain jeus8 -server server2 -u administrator -p <password>

```

```

===== JEUS LICENSE INFORMATION =====
=== VERSION : JEUS 8.0 (Fix#0) (8.0.0.0-b27)
=== EDITION: Enterprise (Trial License)
=== NOTICE: This license restricts the number of allowed clients.
=== Max. Number of Clients: 5
=====
[2016.08.24 14:31:16][0] [launcher-1] [Launcher-0054] Starting the server using the local
configuration.
[2016.08.24 14:31:16][2] [launcher-1] [Launcher-0012] Starting the server [server2] with
the command
/opt/jdk1.7.0_21/jre/./bin/java -Dserver2 -Xmx512m -XX:MaxPermSize=256m
-Djeus.ejb.enable.configDeleteOption=true -server
-Xbootclasspath/p:/home/michael/jeus8_b27/lib/system/extension.jar -classpath
/home/michael/jeus8_b27/lib/system/bootstrap.jar
-Djava.security.policy=/home/michael/jeus8_b27/domains/jeus8/config/security/policy
-Djava.library.path=/home/michael/jeus8_b27/lib/system
-Djava.endorsed.dirs=/home/michael/jeus8_b27/lib/endorsed
-Djeus.properties.replicate=jeus,sun.rmi,java.util,java.net -Djeus.jvm.version=hotspot
-Djava.util.logging.config.file=/home/michael/jeus8_b27/bin/logging.properties
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djeus.home=/home/michael/jeus8_b27 -Djava.net.preferIPv4Stack=true
-Djeus.tm.checkReg=true -Dsun.rmi.dgc.client.gcInterval=3600000 -Djeus.domain.name=jeus8
-Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djeus.server.protectmode=false
-XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput
-XX:LogFile=/home/michael/jeus8_b27/domains/jeus8/servers/server2/logs/jvm.log
jeus.server.ServerBootstrapper
-domain jeus8 -server server2 -u administrator -independent .
[2016.08.24 14:31:16][2] [launcher-1] [Launcher-0014] The server[server2] is being started
...
[2016.08.24 14:31:21][2] [server2-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 14:31:22][0] [server2-1] [SERVER-0000] Version information - JEUS 8.0 (Fix#0)
(8.0.0.0-b27).

... 中略

[2016.08.24 14:31:36][2] [server2-1] [SERVER-0248] The JEUS server is STANDBY.
[2016.08.24 14:31:36][2] [server2-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 14:31:36][2] [server2-1] [WEB-3413] The web engine is ready to receive requests.
[2016.08.24 14:31:36][2] [server2-1] [UNIFY-0100] Listener information
    BASE (plain, 0.0.0.0 : 19736) - VIRTUAL - FileTransfer
                                     - JMXConnectionServer/JEUSMP_server2
                                     - TransactionManager
                                     - SecurityServer
                                     - JMXConnectionServer/JeusMBeanServer
                                     - ClassFTP
                                     - JNDI
                                     - JMSServiceChannel-default
                                     - HTTP
http-server2 (plain, 0.0.0.0 : 9090) - HTTP

```

```
[2016.08.24 14:31:36][0] [server2-1] [SERVER-0249] Successfully started the server
INDEPENDENTLY.
[2016.08.24 14:31:36][2] [server2-1] [SERVER-0248] The JEUS server is RUNNING.
[2016.08.24 14:31:36][2] [server2-1] [SERVER-0401] The elapsed time to start: 19624ms.
[2016.08.24 14:31:36][2] [launcher-10] [Launcher-0034] The server[server2] initialization
completed successfully[pid : 31806].
[2016.08.24 14:31:36][0] [launcher-1] [Launcher-0040] Successfully started the server. The
server state is now RUNNING.
```

上記は、JEUS MSが正常に起動される過程を示しており、起動が完了すると「Successfully started the server. The server state is now RUNNING.」というメッセージが出力されます。

参考

1. 「Invalid License」メッセージはライセンスの異常を意味します。TmaxSoftからライセンスを取得し、`$JEUS_HOME/license`ディレクトリーにコピーします。
2. 全フェーズの進行および環境変数が正常に設定されているのかを確認します。特に、「/jeus/bin」ディレクトリーがシステム・パス(「startManagedServer」スクリプトが実行されるよう)に存在しているのかを確認します。
3. JEUS MSが正常に起動されると、エグザンプル(examples)がデプロイされます。これを確認するには、Webブラウザで<http://localhost:8088/examples/>を呼び出します。

-
3. 別のコンソール画面で以下のコマンドを実行します。「-p」の次にはインストール時に入力したパスワードを入力します。通常、管理者アカウントはadministratorであり、パスワードはJEUSのインストール時に入力した値です。

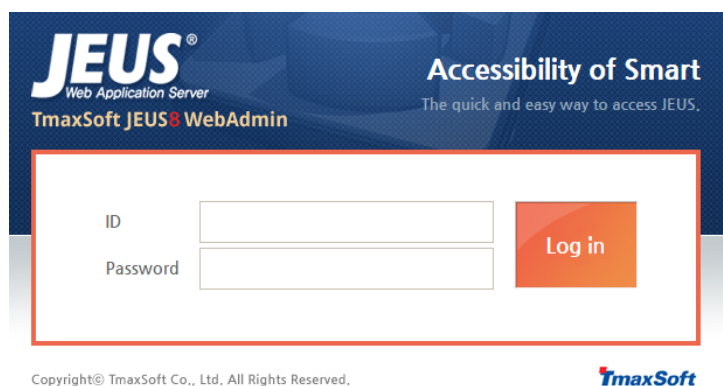
```
[/home/jeus]$ jeusadmin -u administrator -p <password>
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
in the domain jeus_domain.
JEUS8 Administration Tool
To view help, use the 'help' command.
[DAS]jeus_domain.adminServer>
```

4. しばらくしてからプロンプトが表示されますが、JEUSが正常に起動され、再びコマンドが実行できる状態になったことを意味します。
5. Webブラウザを開いて、アドレスバーに以下のアドレスを入力します。

```
http://localhost:9736/webadmin
```

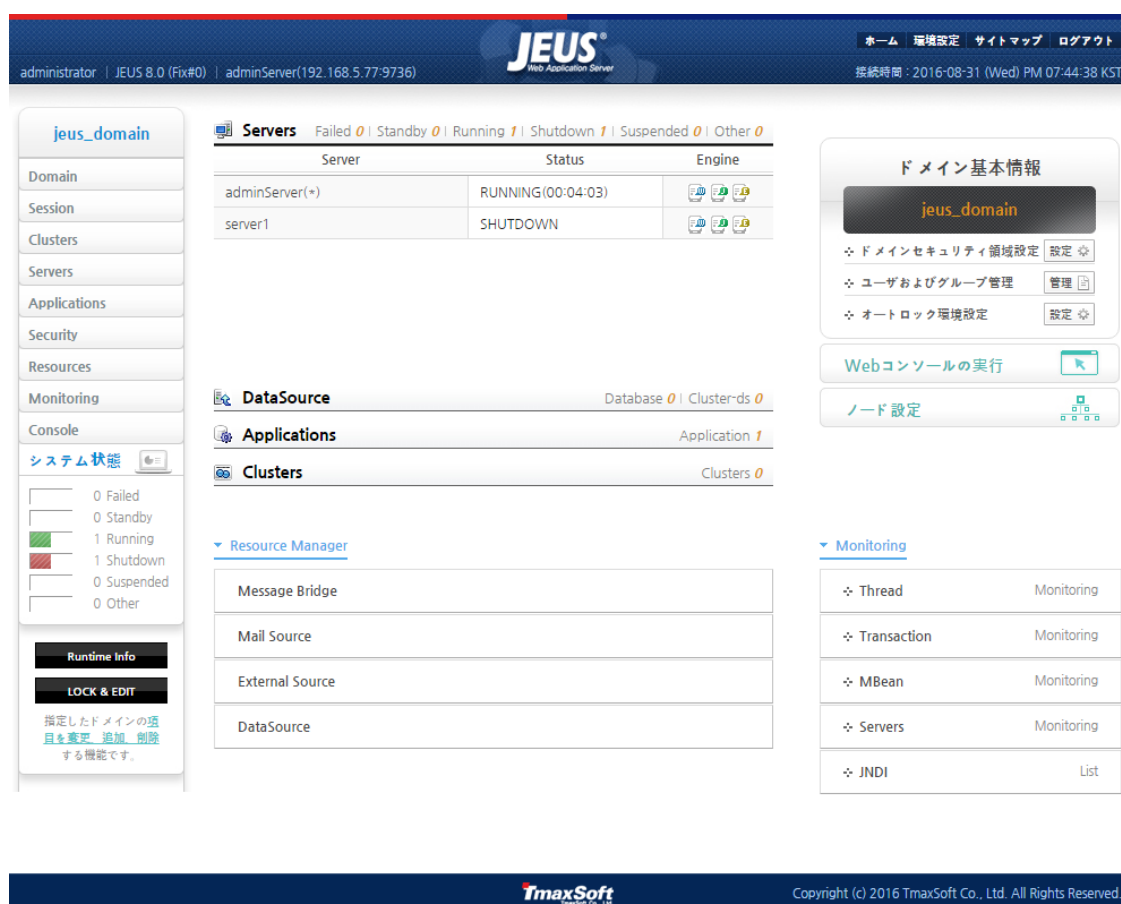

6. JEUS管理者のアカウントとパスワードを入力し、**[Log in]**ボタンをクリックします。通常、管理者アカウントはadministratorであり、パスワードはJEUSのインストール時に入力した値です。

[図 5.16] WebAdminのログイン画面



7. WebAdminを使用してJEUSを管理することができます。

[図 5.17] WebAdminのメイン画面



8. JEUS Managed Serverを起動または停止するときは、左側のメニューから**[Servers]**を選択します。サーバー・リストから目的のサーバーの**[start]**または**[stop]**ボタンをクリックすると、当該サーバーを起動または停止することができます。そのとき、サーバーにはノード・マネージャーが起動されているか、SSHが設定されている必要があります。UNIX/Linux系のサーバーにのみ使用できます。詳細内容については、『JEUS ノードマネージャガイド』を参照してください。

【図 5.18】サーバー・リストの照会画面

The screenshot shows the JEUS web interface. On the left is a sidebar with a menu containing 'Domain', 'Session', 'Clusters', 'Servers' (highlighted), 'Applications', 'Security', 'Resources', 'Monitoring', and 'Console'. Below the menu is a 'システム 状態' (System Status) section with a legend for '0 Failed', '0 Standby', '1 Running', '1 Shutdown', '0 Suspended', and '0 Other'. There are also buttons for 'Runtime Info' and 'LOCK & EDIT'. The main content area is titled 'Servers' and has a search bar and a 'HISTORY' dropdown. Below this is a message: 'ドメイン内でJEUSサーバを構成するとき、各サーバの設定を行います。' (When configuring JEUS servers within a domain, configure each server). There are tabs for 'No Group', 'node', 'Group', and 'Cluster'. A table lists servers with columns: Name, Status, Pid, Need To Restart, Command, and actions (Delete, Add, Duplicate). The table shows 'server1 (*)' as RUNNING and 'server2' as SHUTDOWN. Below the table is a 'Server Templates' section with a message: '該当する内容が存在しません。' (No matching content exists).

9. jeusadminツールにログインした後、**local-start-server**と**local-shutdown**コマンドを使ってJEUSサーバーを制御することができます。

JEUSサーバーを終了するには、**local-shutdown**コマンドを実行します。

```
[DAS]jeus_domain.adminServer>local-shutdown
The server [adminServer] has been shut down successfully.
```

- 10 jeusadminを終了するときは**exit**コマンドを実行します。

```
exit
```

5.4. アンインストール

本節では、GUIモードとコンソール・モードでのアンインストール手順について説明します。

5.4.1. GUIモードでのアンインストール

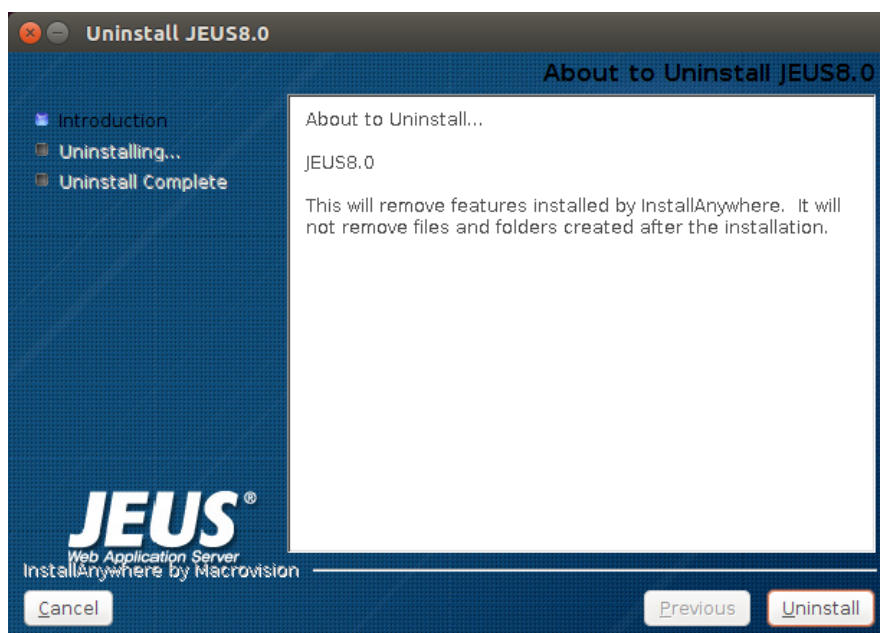
以下は、GUIモードでJEUSをアンインストールする手順についての説明です。GUIモードでのアンインストールができない場合は、「[5.4.2. コンソール・モードでのアンインストール](#)」を参照してください。UNIX環境でGUIインストール・ツールを使用するには、JavaベースのGUI環境をサポートするコンソール、またはターミナルがインストールされている必要があります。

1. 「jeus」スクリプトやWebAdminなど、すべてのJEUSプログラムを終了します。
2. JEUSがインストールされているパスで、\$JEUS_HOME/UninstallerData/Uninstall -i GUIコマンドを実行します。

```
[jeususer@matrix UninstallerData]$./Uninstall -i GUI
```

3. 以下のようなアンインストール初期画面が表示されます。

[図 5.19] UNIX環境のGUIモードでのアンインストール - アンインストールの初期画面



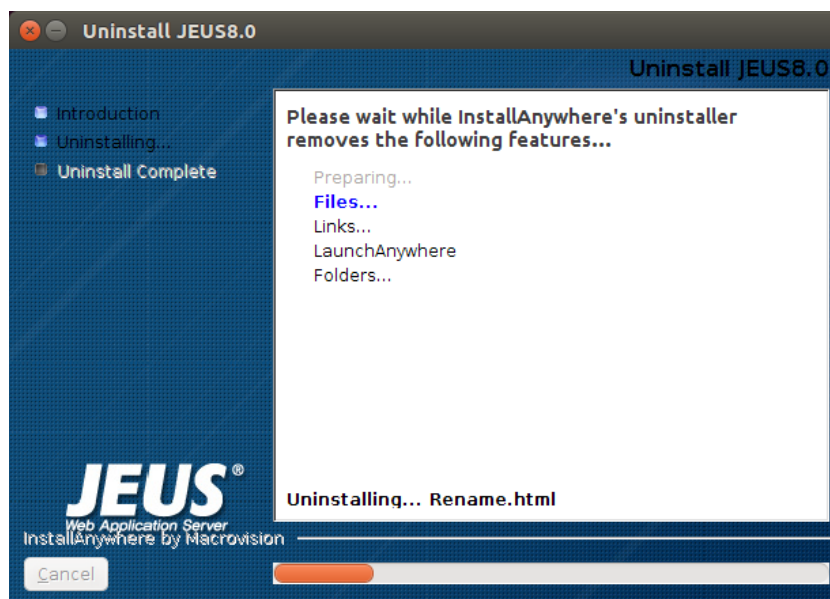
4. アンインストール画面の内容を確認してから、[Uninstall]ボタンをクリックします。JEUSのアンインストールを中止する場合は、[Cancel]ボタンをクリックします。

参考

アンインストールを行ってもJEUSのインストールにより作成されたファイルは削除されません。これらのファイルは直接削除します。

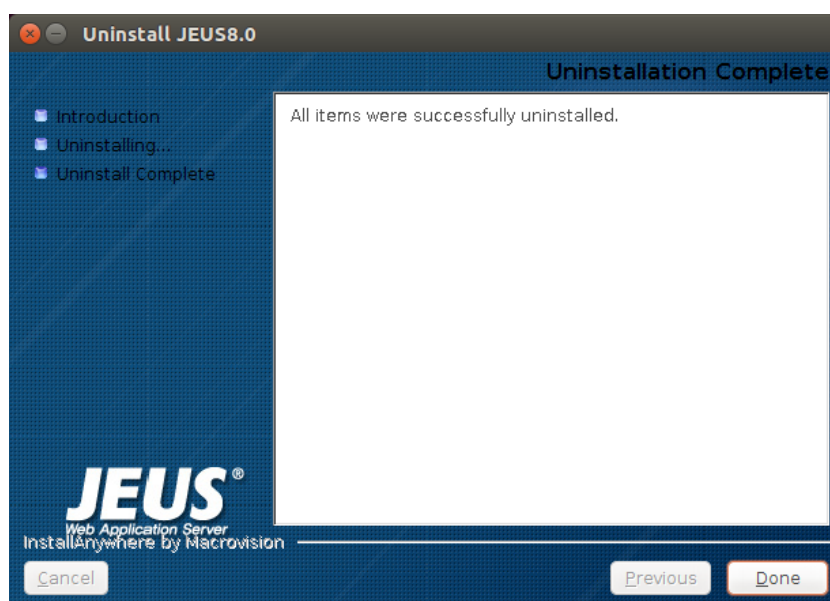
5. JEUSのアンインストールの進行状況を確認します。

[図 5.20] UNIX環境のGUIモードでのアンインストール - Uninstall JEUS



6. JEUSのアンインストールが完了したら、**[Done]**ボタンをクリックしてJEUSのアンインストールを終了します。

[図 5.21] UNIX環境のGUIモードでのアンインストール - Uninstall Complete



5.4.2. コンソール・モードでのアンインストール

以下は、コンソール・モードにおいてJEUSをアンインストールする手順についての説明です。

1. JEUSがインストールされているパスで、\$JEUS_HOME/UninstallerData/Uninstallを実行してJEUSコアとJEUSのインストール・ディレクトリーを削除します。

```
[jeususer@matrix UninstallerData]$./Uninstall
```

2. アンインストールが行われます。削除が完了するとメッセージが出力されます。

```
Preparing CONSOLE Mode Installation...

=====
JEUS8.0                                (created with InstallAnywhere by Macrovision)
-----

=====

Uninstall JEUS8.0
-----

About to uninstall...

JEUS8.0

This will remove features installed by InstallAnywhere.  It will not remove
files and folders created after the installation.

PRESS <ENTER> TO CONTINUE:

=====

Uninstalling...
-----

...*
*
*****
*****
*****
*****

...*
*
*****
*****
*****
*****

...

```

```
=====
Uninstallation Complete
-----

All items were successfully uninstalled.
```

5.5. アンインストールの確認

JEUSがインストールされているパスで、ディレクトリーやファイルが削除されたことを確認します。JEUSのインストールにより作成されたファイルは削除されないため、直接削除します。

第6章 Windows環境でのインストールおよびアンインストール

本章では、Windows環境でJEUSをインストールおよびアンインストールする手順について説明します。

6.1. 概要

Windows環境でのJEUSのインストールとアンインストールは、GUIインストール・ツールを使用します。

以下は、Windows環境でのJEUSのインストール手順です。

1. JDK 7をインストール
2. GUIモードでのJEUSインストール(ファイル・コピー)
3. インストールの確認
 - 起動の確認
 - 環境変数の設定と確認

6.2. インストール

本節では、インストール・ツールを使用してWindows環境でJEUSをインストールする方法について説明します。

参考

UNIX環境のコンソール・モードとは違って、Windows環境ではコマンドライン・ツールを提供しません。

JDK 7をインストールした後、WindowsにJEUSをインストールする手順は以下のとおりです。

1. JEUSインストール・プログラムである`jeus80_win.exe`が格納されているディレクトリに移動します。当該ファイルをダブルクリックして実行します。

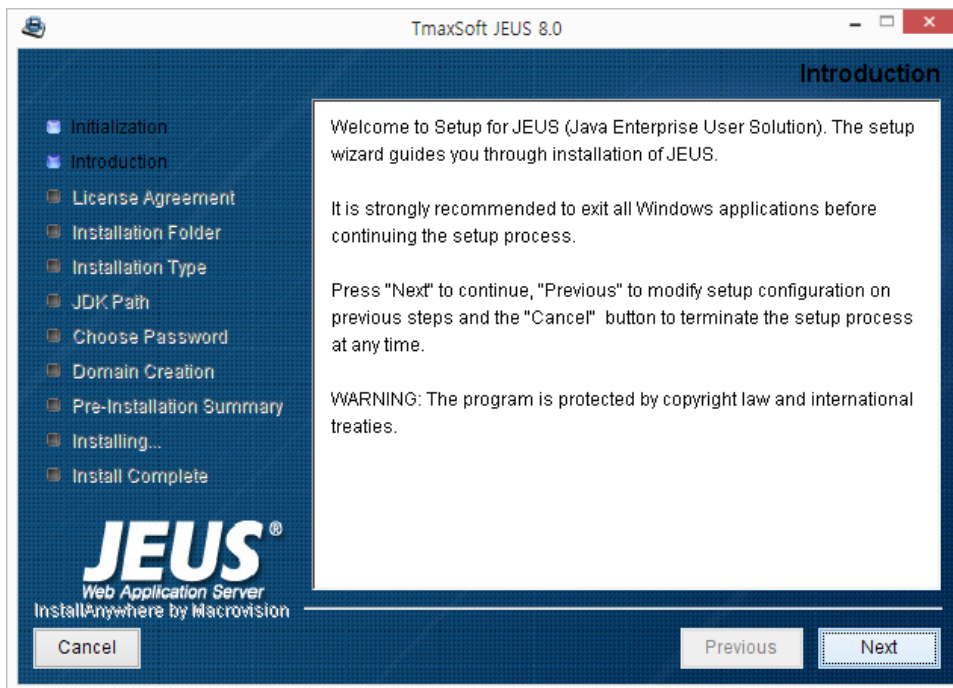
2. インストール・ツールの初期画面が表示されます。進行状況が100%になるまで待ちます。

[図 6.1] Windows環境でのインストール - インストール・ツールの初期画面



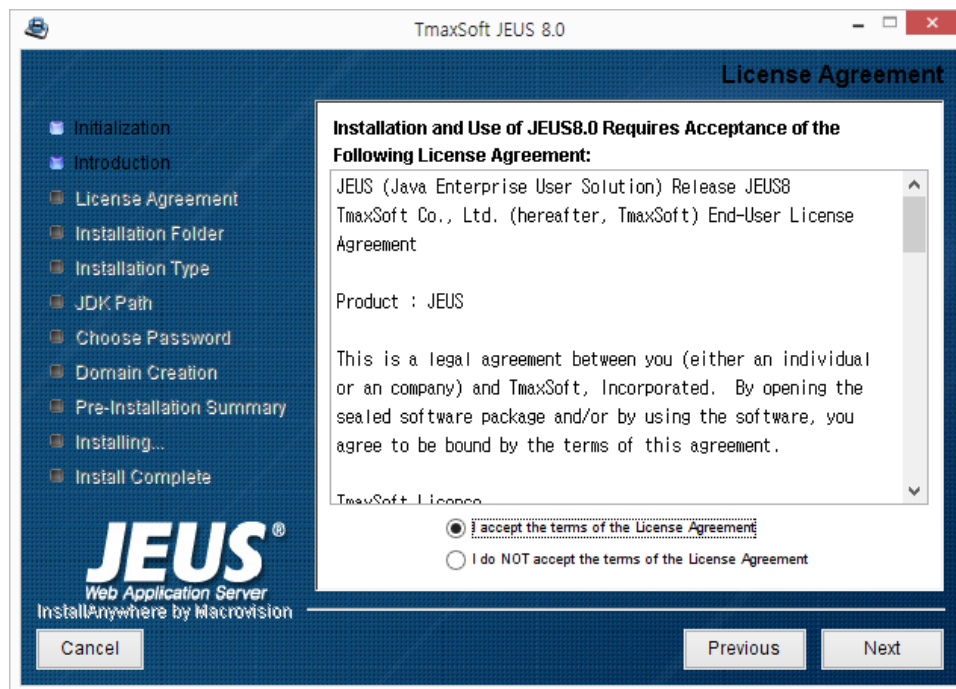
3. [Introduction]画面が表示されます。内容を確認してから[Next]ボタンをクリックします。インストールを中止する場合は、[Cancel]ボタンをクリックします。

[図 6.2] Windows環境でのインストール - Introduction



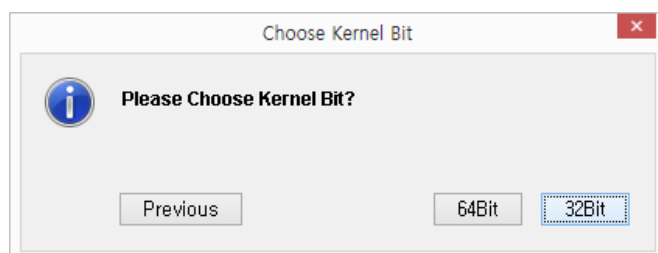
4. **[License Agreement]**画面が表示されます。JEUSの使用許諾契約の内容を確認してから「**I accept the terms of ...**」を選択して**[Next]**ボタンをクリックします。

【図 6.3】Windows環境でのインストール - License Agreement



5. **[Choose Kernel Bit]**画面が表示されます。この画面は、Windows 64ビットのマシンにインストールするときのみ表示されます。インストールされているJavaのビット数に合わせて選択してください。

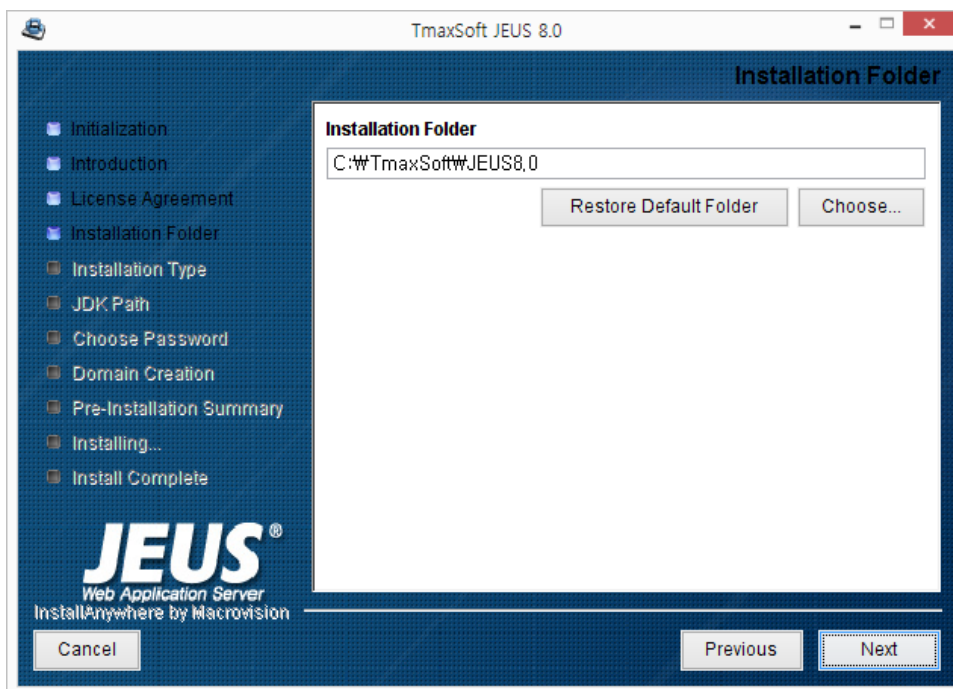
【図 6.4】Windows環境でのインストール - Choose Kernel Bit



6. **[Installation Folder]**画面が表示されます。

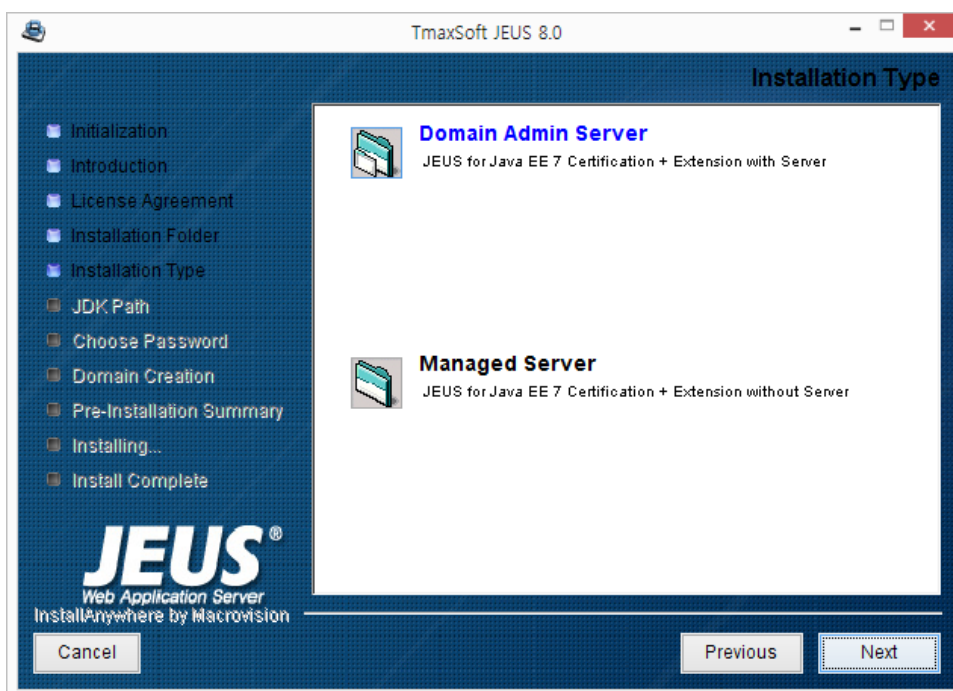
JEUSバイナリーとその他のファイルはインストールしたパスのディレクトリーの下位に格納されます。「**Choose...**」をクリックすると、別のインストール・フォルダーが選択できます。「**Restore Default Folder**」をクリックすると初期のデフォルト・フォルダーに戻ります。デフォルト・フォルダーの使用をお勧めします(例: C:\TmaxSoft\JEUS8.0)。インストールするディレクトリーを選択し、**[Next]**ボタンをクリックします。

【図 6.5】 Windowsでのインストール - Installation Folder



7. [Installation Type]画面が表示されます。インストール・オプションの「Domain Admin Server」と「Managed Server」から1つを選択し、[Next]ボタンをクリックします。

【図 6.6】 Windows環境でのインストール - Installation Type

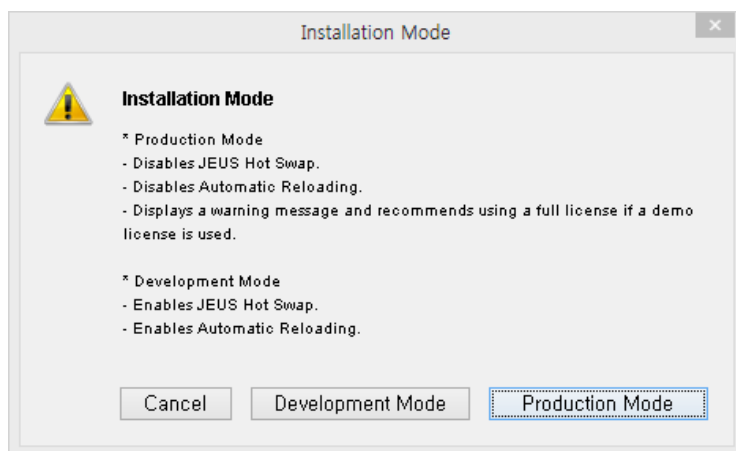


以下は、インストール・オプションについての説明です。

項目	説明
Domain Admin Server	Domain Admin Server(DAS)とManaged Server(MS)をインストールします。 Domain Admin ServerはManaged Serverを管理するためのサーバーです(デフォルト値)
Managed Server	Managed Serverをインストールします

8. **Domain Admin Server**を選択した場合、**Installation Mode**を選択します。

[図 6.7] Windows環境でのインストール - Installation Mode

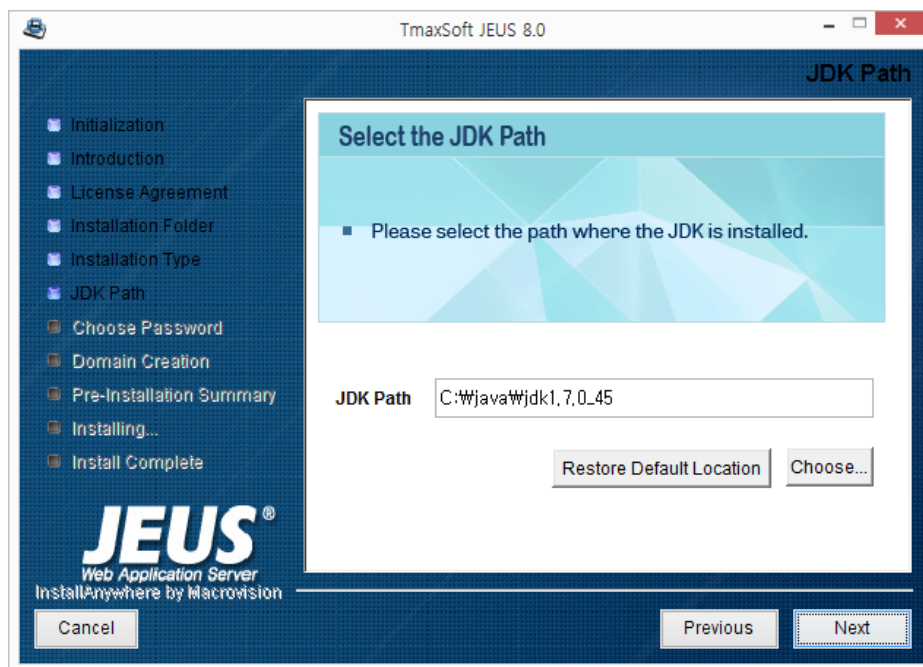


以下は、インストール・オプションについての説明です。

項目	説明
Production Mode	インストール時にプロダクション・モードでインストールします。JEUSホットスワップ、自動リロードを使用しません。デモライセンスが使用されると、警告メッセージが出力されます
Development Mode	インストール時にデプロイメント・モードでインストールします。JEUSホットスワップ、自動リロードを使用します

9. **JDK Path**欄にインストールされたJDKパス(例: C:\java\1.7.0.0)を入力します。インストール・プログラムはインストールされたJDKパスをお勧めしますが、「**Choose...**」をクリックすると別のフォルダーが選択でき、**[Restore Default Location]**をクリックするとデフォルト・フォルダーに戻ります。

[図 6.8] Windows環境でのインストール - JDK Path



参考

JEUSのインストール後にJDKをインストールすると、JDKのインストール・パスを含むJAVA_HOME環境変数を修正する必要があるため、JDKをインストールした後にJEUSをインストールすることをお勧めします。

たとえば、Windowsコマンド・プロンプトで`set JAVA_HOME=C:\java\jdk1.7.0.0`コマンドを実行して設定します。

-
- 10 **Windows NTサービスの登録選択**画面が表示されます。サービスの登録は、Windows環境のインストール・ファイルにのみ表示されます。

[図 6.9] Windowsサービスの登録選択



11. **[Choose Password]**画面でJEUS管理者のパスワードを入力します。パスワードは「administrator」管理者アカウントで割り当てられます。

[図 6.10] Windows環境でのインストール - Choose Password

TmaxSoft JEUS 8.0

Choose Password

- Initialization
- Introduction
- License Agreement
- Installation Folder
- Installation Type
- JDK Path
- Choose Password
- Domain Creation
- Pre-Installation Summary
- Installing...
- Install Complete

JEUS®
Web Application Server
InstallAnywhere by Macrovision

Cancel Previous Next

Get Password

- Enter the password for the administrator account.
- This password will be registered in JEUS as the first user.

Password

Confirm Password

- 12 JEUSで使用するドメイン名を入力し、**[Next]**ボタンをクリックします。

[図 6.11] Windows環境でのインストール - Domain name

TmaxSoft JEUS 8.0

Domain Name

- Initialization
- Introduction
- License Agreement
- Installation Folder
- Installation Type
- JDK Path
- Choose Password
- Domain Creation
- Pre-Installation Summary
- Installing...
- Install Complete

JEUS®
Web Application Server
InstallAnywhere by Macrovision

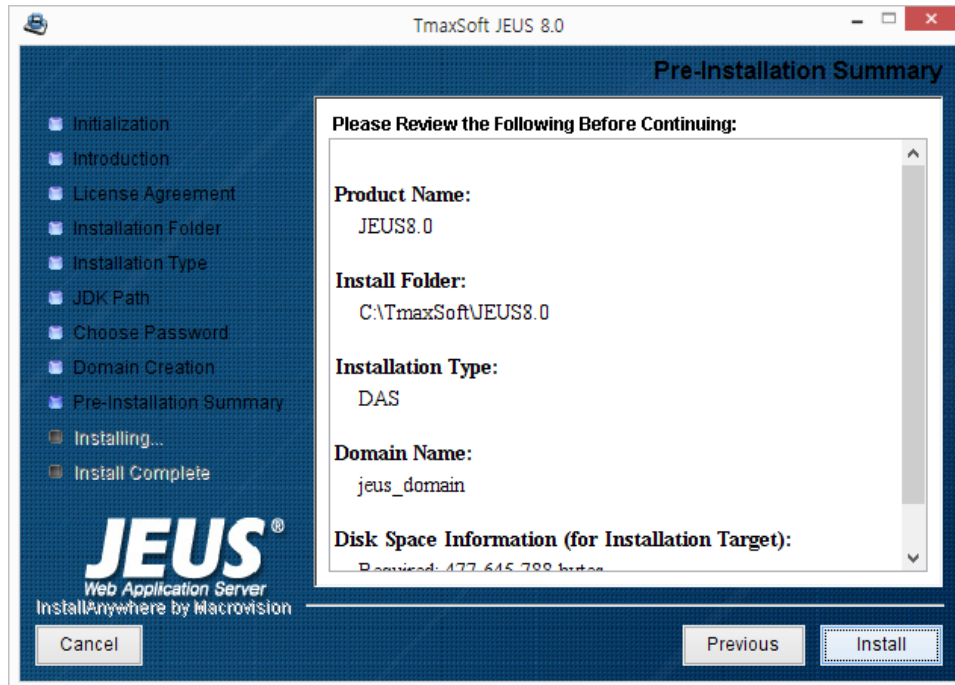
Cancel Previous Next

Enter a domain name.
(A domain is a group of associated servers and is the basic management unit. A domain consists of Domain Administration Server, Managed Servers, and Clusters.)

Domain jeus_domain

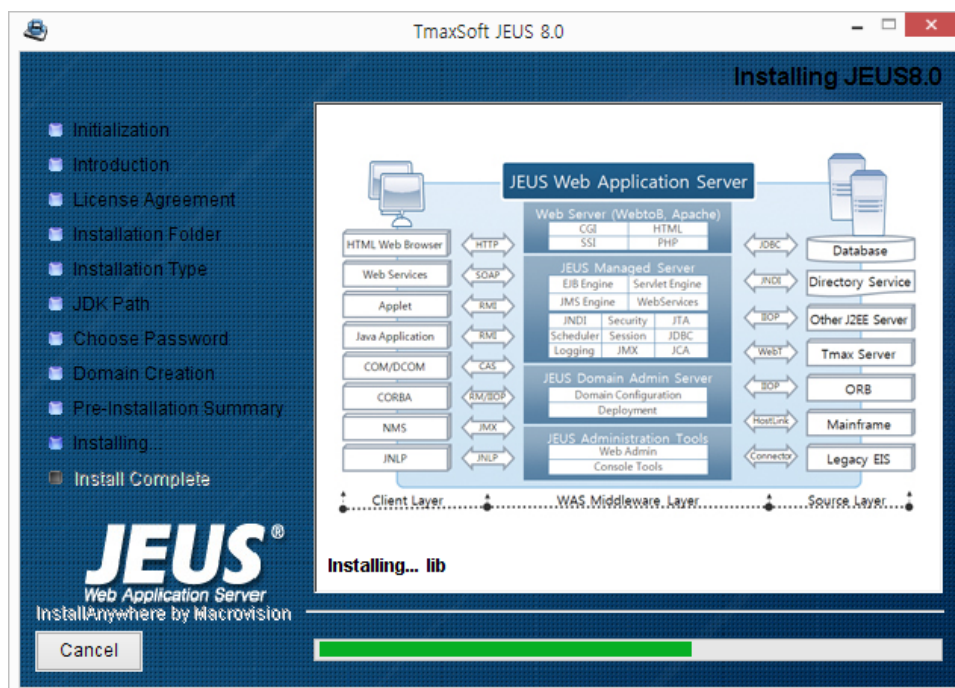
13 これまで設定したインストール情報が出力されます。インストール情報を確認した後、[Install]ボタンをクリックしてインストールを開始します。

[図 6.12] Windows環境でのインストール - Pre Installation Summary



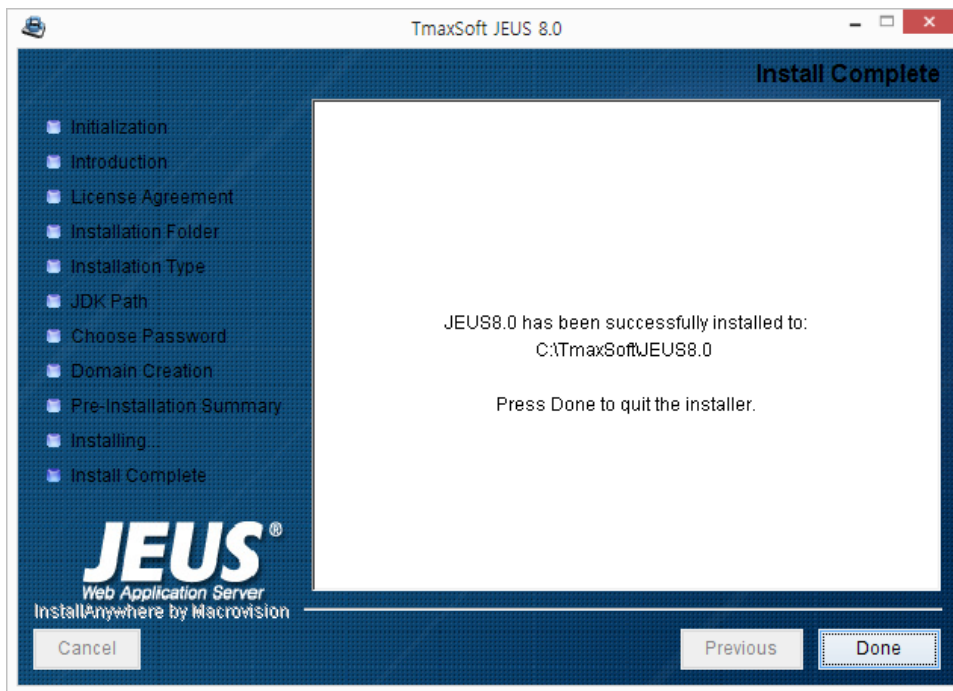
14 インストールの進行状況が確認できます。

[図 6.13] Windows環境でのインストール - Installing JEUS



15 インストールの完了を確認し、[Done]ボタンをクリックするとインストールが終了します。

[図 6.14] Windows環境でのインストール - Install Complete



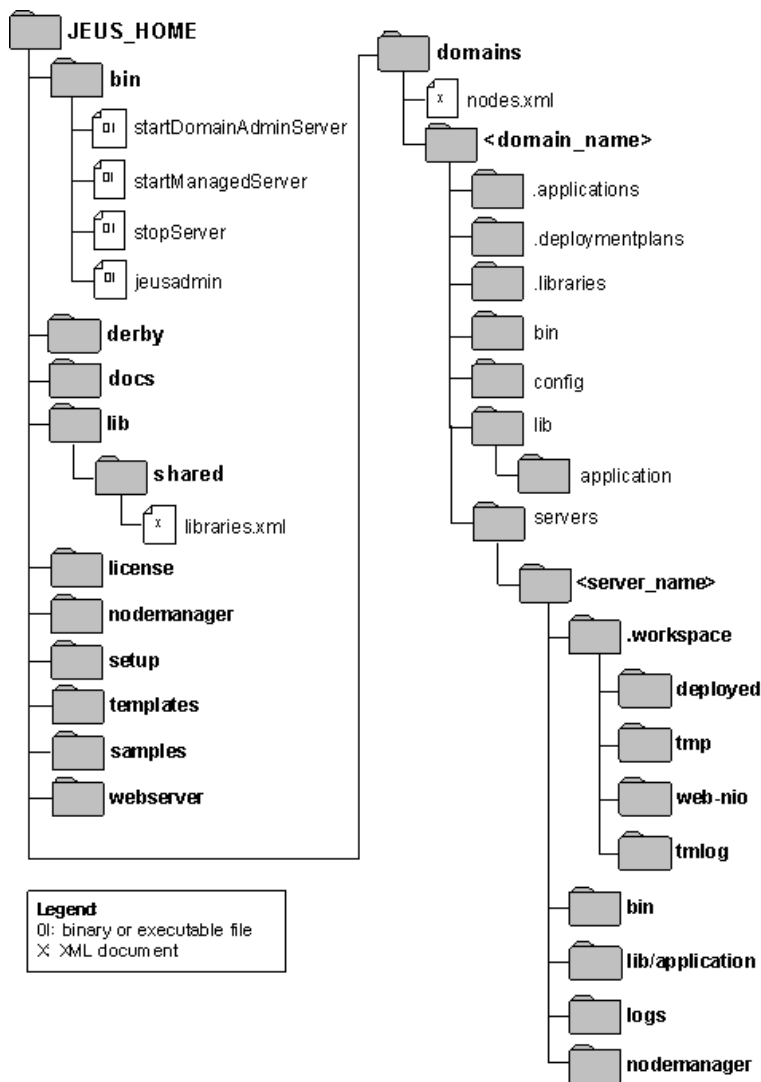
6.3. インストールの確認

JEUSをインストールした後、使用に必要な環境変数を設定します。環境変数を設定した後はJEUSを起動して正常にインストールされているのかを確認します。環境変数を設定する前にJEUSがインストールされたパスのディレクトリーを確認します。

6.3.1. ディレクトリー構造

以下は、JEUSをインストールしたときのディレクトリー構造です。

[図 6.15] JEUSインストール後のディレクトリー構造



以下は、ディレクトリーとファイルについての説明です。

{JEUS_HOME}

JEUSのルート・ディレクトリーとして、実際のディレクトリー名と場所はインストール時に決まります。

bin

サーバーの起動および終了スクリプトであるstartDomainAdminServer、startManagedServer、stopServerとJEUSコンソール・ツール(jeusadmin)のような実行ファイルが存在します。

derby

サンプル・アプリケーションの作成やテスト環境で容易に使用できるようにApache Derbyが含まれています。

docs

JEUSが提供するAPIのJavadocが存在します。

lib

JEUSの起動に必要なライブラリーが存在します。ユーザーは、sharedディレクトリー以外のディレクトリーにはアクセスする必要がありません。

ディレクトリー	説明
shared	sharedディレクトリーには、アプリケーションで使用するライブラリーが存在します。 sharedディレクトリーのライブラリーを使用するには、libraries.xmlにライブラリーの情報を追加する必要があります。なお、当該ライブラリーを使用するアプリケーションのJEUS Deployment Descriptor(DD)で当該ライブラリーについてのリファレンス情報を指定します。sharedライブラリーの詳細内容は、『JEUS アプリケーション&デプロイメントガイド』の「3.3.2. 共有ライブラリー」を参照してください

license

JEUSのライセンス・ファイルが存在します。ライセンス・ファイルは、JEUSが実行されるための必須ファイルです。

nodemanager

JEUSのノード・マネージャーのための設定ファイルであるjeusnm.xmlファイルが存在します。

setup

JEUSのインストール後、使用できる環境を構築するための必要ファイルが存在します。

templates

各種設定や環境などのテンプレート・ファイルが存在します。

samples

JEUSのサンプル・ファイルが存在します。

webserver

JEUSのインストール時にJEUS Webサーバーがインストールされるディレクトリーです。詳しい内容は、『JEUS Webエンジンガイド』を参照してください。

domains

各ドメインの下位にDOMAIN_HOMEとJEUS_HOMEで使用するノード情報が含まれているnodes.xmlが存在します。

以下のディレクトリーおよびファイルは、DOMAIN_HOMEの下位に存在します。

- .applications

各ドメインで管理するアプリケーション・ファイルが存在します。

追加および削除には、**install-application**、**uninstall-application**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.7. install-application」、『*JEUS リファレンスガイド*』の「4.2.6.16. uninstall-application」を参照してください。

- .deploymentplans

ドメインで管理するデプロイメント・プラン・ファイルが存在します。

追加および削除には、**install-deployment-plan**、**uninstall-deployment-plan**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.8. install-deployment-plan」、『*JEUS リファレンスガイド*』の「4.2.6.17. uninstall-deployment-plan」を参照してください。

- .libraries

ドメインで管理するライブラリー・ファイルが存在します。

追加および削除には、**install-application**、**uninstall-application**コマンドのみ使用できます。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。各コマンドの説明は、『*JEUS リファレンスガイド*』の「4.2.6.7. install-application」、『*JEUS リファレンスガイド*』の「4.2.6.16. uninstall-application」を参照してください。

- bin

各ドメインに属しているDASとMSの開始および終了スクリプトが存在します。JEUS_HOME/binのstartDomainAdminServer、startManagedServer、stopServerと同様な機能を実行しますが、ドメイン名を設定する必要はありません。

- config

ドメインの設定ファイルであるdomain.xmlが変更された場合、以前のヒストリーのために存在するバックアップ・ファイルが格納されます。ドメイン設定についての詳細説明は、『*JEUS ドメインガイド*』の「第3章 ドメインの設定変更」を参照してください。

区分	説明
security	<ul style="list-style-type: none">– SYSTEM_DOMAIN : ドメイン単位で適用されるセキュリティー・ドメイン・ファイルのaccounts.xml、policies.xmlが存在しており、各XMLファイルはWebAdminとjeusadminを利用して動的な設定変更が可能です。セキュリティー・ドメイン設定についての詳細説明は、『<i>JEUS セキュリティガイド</i>』の「2.2. セキュリティー・ドメインの定義」を参照してください– security.key : 対称鍵暗号化アルゴリズムのキーを保存するファイルです。JEUS_HOME/bin/encryptionを実行すると作成されます。security.keyファイルについての詳細説明は、『<i>JEUS セキュリティガイド</i>』の「2.5.4. パスワードのセキュリティー設定」を参照してください

区分	説明
	<ul style="list-style-type: none"> – policy : Javaパーミッション設定ファイルです。JEUSのセキュリティ・システムとは別途にJava SEセキュリティ・マネージャーで使われます
servlet	<ul style="list-style-type: none"> – web.xml : web.xmlを持っていない場合、Webエンジンが使用するWebモジュールのweb.xmlです。デフォルト値は、空きのXMLファイルです – webcommon.xml: ドメイン内サーバーのWebエンジンの全Webモジュールに適用される共通設定ファイルです。設定についての詳細内容は、『<i>JEUS Webエンジンガイド</i>』の「1.5. ディレクトリー構造」を参照してください

- lib/application

全ドメインに適用したいアプリケーション・ライブラリーを格納するディレクトリーです。

SERVER_HOMEに存在するアプリケーション・ライブラリーと衝突した場合、SERVER_HOME/lib/applicationが優先され、警告メッセージが残ります。lib/applicationディレクトリーについての詳細説明は、『*JEUS アプリケーション&デプロイメントガイド*』の「3.3.1. lib/applicationディレクトリー」を参照してください。

- servers

このディレクトリーの下位にSERVER_HOMEディレクトリーがサーバー名で作成されます。SERVER_HOMEディレクトリー構造の詳細説明は、『*JEUS サーバガイド*』の「1.5. サーバー・ディレクトリー構造」を参照してください。

ディレクトリー	説明
.workspace	JEUSが使用するサーバー別のスペースなので、ユーザーが変更してはなりません
bin	<p>サーバーの開始/終了スクリプトが含まれています。JEUS_HOME/binのスクリプトと同様な機能を行います。ドメイン名とサーバー名を設定する必要はありません</p> <ul style="list-style-type: none"> – DASの場合 : startDomainAdminServer/stopServerが存在します – MSの場合: startManagedServer/stopserverが存在します
lib/application	<p>サーバーに適用したいアプリケーション・ライブラリーが存在します。ドメイン領域のライブラリー(DOMAIN_HOME/lib/application)より優先順位が高いです。ライブラリーが衝突した場合、同ディレクトリーに存在するファイルが適用され、警告メッセージが残ります。</p> <p>lib/applicationの詳細内容は、『<i>JEUS アプリケーション&デプロイメントガイド</i>』の「3.3.1. lib/applicationディレクトリー」を参照してください</p>

ディレクトリー	説明
logs	サーバーのランチャー(Launcher)ログ、サーバー・ログ、アクセス・ログファイルが残ります。詳細内容は、『JEUS サーバガイド』の「第8章 ロギング」を参照してください
nodemanager	ノード・マネージャーが再起動する際、管理していたサーバーなのかどうかを判断するための情報を保存します。JEUSが使用するディレクトリーとして、ユーザーのアクセスが制限されます。

6.3.2. 環境設定

環境変数のPATHが正しく設定されたのかを確認するため、Windowsコマンド・プロンプトに**set**コマンドを実行します。また、Windowsの**[スタート] > [設定] > [コントロール・パネル] > [システム] > [詳細設定] > [環境変数]**で確認できます。以外の環境変数は、%JEUS_HOME%\bin\jeus.properties.cmdファイルに設定されます。

以下は、主な環境変数についての説明です。

[表 6.1] 環境変数の設定

環境変数	説明
PATH	システム・パスを設定します。 以下が含まれている必要があります <ul style="list-style-type: none"> – C:\TmaxSoft\JEUS8.0\bin – C:\TmaxSoft\JEUS8.0\lib\system
JEUS_HOME	JEUSのインストール・ディレクトリーを設定します (例: C:\TmaxSoft\JEUS8.0)
JEUS_LIBPATH	JEUSのライブラリー・ファイルのパスを設定します (例: C:\TmaxSoft\JEUS8.0\lib\system)
VM_TYPE	Java HotSpot JVMの使用有無を設定します。 以下の値で設定します <ul style="list-style-type: none"> – hotspot : -serverオプションで動作します – old : vmオプションが含まれません。vmオプションが動作しないサーバーは「old」に設定します
USERNAME	管理者アカウントのIDを設定します

環境変数	説明
PASSWORD	管理者パスワードを設定します
JAVA_HOME	JDKのインストール・ディレクトリー・パスを設定します (例: C:\java\1.7.0.0)
JAVA_ARGS	JDKのパラメータを設定します
JAVA_VENDOR	JDKのベンダーを設定します (例: Sun、IBM、HP)

6.3.3. 起動の確認

JEUSが正常にインストールされたのかを確認するため、以下の手順でJEUSを起動します。

1. コマンド・プロンプトを実行し、「**startDomainAdminServer -u administrator -p <password>**」を入力します。

以下のメッセージがコマンド・プロンプト画面に表示されます。

```
C:\TmaxSoft\JEUS8.0\bin>startDomainAdminServer -u administrator -p <password>
*****
- JEUS Home           : C:\TmaxSoft\JEUS8.0
- Added Java Option  :
- Java Vendor        : Sun
*****

C:\Users\paper>"D:\java\jdk1.7.0_80\jre\..\bin\java" -server -Xmx512m
-Xbootclasspath/p:"C:\TmaxSoft\JEUS8.0\lib\system\extension.jar"
-classpath "C:\TmaxSoft\JEUS8.0\lib\system\bootstrap.jar" -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djeus.jvm.version=hotspot -Djeus.home="C:\TmaxSoft\JEUS8.0" -Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djava.library.path="C:\TmaxSoft\JEUS8.0\lib\system"
-Djava.endorsed.dirs="C:\TmaxSoft\JEUS8.0\lib\endorsed" -Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djava.util.logging.config.file="C:\TmaxSoft\JEUS8.0\bin\logging.properties"
-Djeus.properties.replicate=jeus,java.util.logging,sun.rmi.dgc
-Djava.net.preferIPv4Stack=true jeus.server.LauncherBootstrapper -u administrator -p <password>

===== JEUS LICENSE INFORMATION =====
=== VERSION : JEUS 8.0 (Fix#0) (8.0.0.0-b27)
=== EDITION: Enterprise (Trial License)
=== NOTICE: This license restricts the number of allowed clients.
=== Max. Number of Clients: 5
=====
[2016.08.24 15:39:19][2] [launcher-1] [Launcher-0012] Starting the server [server1] with the command
D:\java\jdk1.7.0_80\jre\..\bin\java -Dserver1 -Xmx512m -XX:MaxPermSize=256m -Djeus.ejb.enable.configDeleteOption=true
-server
-Xbootclasspath/p:C:\TmaxSoft\JEUS8.0\lib\system\extension.jar -classpath C:\TmaxSoft\JEUS8.0\lib\system\bootstrap.jar
-Djava.security.policy=C:\TmaxSoft\JEUS8.0\domains\jeus8\config\security\policy
-Djava.library.path=C:\TmaxSoft\JEUS8.0\lib\system
-Djava.endorsed.dirs=C:\TmaxSoft\JEUS8.0\lib\endorsed -Djeus.properties.replicate=jeus,java.util.logging,sun.rmi.dgc
-Djeus.jvm.version=hotspot
-Djava.util.logging.config.file=C:\TmaxSoft\JEUS8.0\bin\logging.properties -Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.util.logging.manager=jeus.util.logging.JeusLogManager -Djeus.home=C:\TmaxSoft\JEUS8.0
-Dsun.rmi.dgc.client.gcInterval=3600000
-Djeus.domain.name=jeus8 -Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
-Djeus.server.protectmode=false -XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput
```

```

-XX:LogFile=C:\TmaxSoft\JEUS8.0\domains\jeus8\servers\server1\logs\jvm.log jeus.server.admin.DomainAdminServerBootstrapper
-u administrator -domain jeus8 -server server1 .
[2016.08.24 15:39:19][2] [launcher-1] [Launcher-0014] The server[server1] is being started ...
[2016.08.24 15:39:21][2] [server1-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 15:39:21][0] [server1-1] [SERVER-0000] Version information - JEUS 8.0 (Fix#0) (8.0.0.0-b27)

... 中略

[2016.08.24 15:39:31][2] [server1-1] [WEB-3413] The web engine is ready to receive requests.
[2016.08.24 15:39:31][2] [server1-1] [UNIFY-0100] Listener information
                        BASE (plain, 0.0.0.0 : 9736) - VIRTUAL - SecurityServer
                                                    - ClassFTP
                                                    - JMSServiceChannel-channel_oneport
                                                    - JMXConnectionServer/JeusMBeanServer
                                                    - JMSServiceChannel-channel_noconf
                                                    - JMXConnectionServer/JEUSMP_server1
                                                    - FileTransfer
                                                    - JNDI
                                                    - TransactionManager
                                                    - BootTimeFileTransfer
                                                    - JMSServiceChannel-channel_security
                        - RMI
                        - HTTP
                        http-server (plain, 0.0.0.0 : 9099) - HTTP
                        jms-jmstest (plain, 0.0.0.0 : 9931) - VIRTUAL - JMSServiceChannel-jmstest
jms-channel_serverport (plain, 0.0.0.0 : 39742) - VIRTUAL - JMSServiceChannel-channel_serverport
                                                    - JMSServiceChannel-channel_serverurl
                        jms-channel_port0 (plain, 0.0.0.0 : 0) - VIRTUAL - JMSServiceChannel-channel_port0
                        jms-channel_nio (plain, 0.0.0.0 : 9932) - VIRTUAL - JMSServiceChannel-channel_nio
jms-channel_clientlimit (plain, 0.0.0.0 : 9933) - VIRTUAL - JMSServiceChannel-channel_clientlimit

[2016.08.24 15:39:31][0] [server1-1] [SERVER-0242] Successfully started the server.
[2016.08.24 15:39:31][2] [server1-1] [SERVER-0248] The JEUS server is RUNNING.
[2016.08.24 15:39:31][2] [server1-1] [SERVER-0401] The elapsed time to start: 12254ms.
[2016.08.24 15:39:31][2] [launcher-10] [Launcher-0034] The server[server1] initialization completed successfully[pid :
11216].
[2016.08.24 15:39:31][0] [launcher-1] [Launcher-0040] Successfully started the server. The server state is now RUNNING.

```

出力メッセージの「Successfully started the server. The server state is now RUNNING.」は、JEUSのDASが起動されたことを意味します。

参考

1. 「Invalid License」メッセージはライセンスの異常を意味します。TmaxSoftからライセンスを取得し、%JEUS_HOME%\licenseディレクトリーにコピーします。
2. 全フェーズの進行および環境変数が正常に設定されているのかを確認します。特に、「C:\TmaxSoft\JEUS8.0\bin」ディレクトリーがシステム・パス(startDomainAdminServer.cmdスクリプトが実行)に含まれたのかを確認します。

2. コマンド・プロンプトを実行し、「startManagedServer -domain <domain_name> -server <server_name> -u administrator -p <password>」を入力します。

以下のメッセージがコマンド・プロンプト画面に表示されます。

```

C:\TmaxSoft\JEUS8.0\bin>startManagedServer -server server2 -u administrator -p <password>
*****
- JEUS Home      : C:\TmaxSoft\JEUS8.0
- JEUS Base Port :
- Added Java Option :
- Java Vendor    : Sun

```

```

*****

C:\TmaxSoft\JEUS8.0\bin>"D:\java\jdk1.7.0_80\jre\..\bin\java" -server -Xmx512m
-Xbootclasspath/p:"C:\TmaxSoft\JEUS8.0\lib\system\extension.jar"
-classpath "C:\TmaxSoft\JEUS8.0\lib\system\bootstrap.jar" -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djeus.jvm.version=hotspot -Djeus.home="C:\TmaxSoft\JEUS8.0" -Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djava.library.path="C:\TmaxSoft\JEUS8.0\lib\system"
-Djava.endorsed.dirs="C:\TmaxSoft\JEUS8.0\lib\endorsed" -Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djava.util.logging.config.file="C:\TmaxSoft\JEUS8.0\bin\logging.properties"
-Djeus.properties.replicate=jeus,java.util.logging,sun.rmi.dgc
-Djava.net.preferIPv4Stack=true jeus.server.ManagedServerLauncherBootstrapper -server server2 -u administrator -p
<password>

===== JEUS LICENSE INFORMATION =====
=== VERSION : JEUS 8.0 (Fix#0) (8.0.0.0-b27)
=== EDITION: Enterprise (Trial License)
=== NOTICE: This license restricts the number of allowed clients.
=== Max. Number of Clients: 5
=====
[2016.08.24 15:40:48][0] [launcher-1] [Launcher-0054] Starting the server using the local configuration.
[2016.08.24 15:40:48][2] [launcher-1] [Launcher-0012] Starting the server [server2] with the command
D:\java\jdk1.7.0_80\jre\..\bin\java -Dserver2 -Xmx512m -XX:MaxPermSize=256m -Djeus.ejb.enable.configDeleteOption=true
-server
-Xbootclasspath/p:C:\TmaxSoft\JEUS8.0\lib\system\extension.jar -classpath C:\TmaxSoft\JEUS8.0\lib\system\bootstrap.jar
-Djava.security.policy=C:\TmaxSoft\JEUS8.0\domains\jeus8\config\security\policy
-Djava.library.path=C:\TmaxSoft\JEUS8.0\lib\system
-Djava.endorsed.dirs=C:\TmaxSoft\JEUS8.0\lib\endorsed -Djeus.properties.replicate=jeus,java.util.logging,sun.rmi.dgc
-Djeus.jvm.version=hotspot -Djava.util.logging.config.file=C:\TmaxSoft\JEUS8.0\bin\logging.properties
-Dsun.rmi.dgc.server.gcInterval=3600000 -Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djeus.home=C:\TmaxSoft\JEUS8.0
-Dsun.rmi.dgc.client.gcInterval=3600000 -Djeus.domain.name=jeus8 -Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djeus.server.protectmode=false -XX:+UnlockDiagnosticVMOptions
-XX:+LogVMOutput
-XX:LogFile=C:\TmaxSoft\JEUS8.0\domains\jeus8\servers\server2\logs\jvm.log jeus.server.ServerBootstrapper -u administrator
-server server2 -verbose -domain jeus8 -independent .
[2016.08.24 15:40:48][2] [launcher-1] [Launcher-0014] The server[server2] is being started ...
[2016.08.24 15:40:51][2] [server2-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 15:40:51][0] [server2-1] [SERVER-0000] Version information - JEUS 8.0 (Fix#0) (8.0.0.0-b27).

... 中略

[2016.08.24 15:40:53][2] [server2-1] [WEB-3413] The web engine is ready to receive requests.
[2016.08.24 15:40:53][2] [server2-1] [UNIFY-0100] Listener information
                                BASE (plain, 0.0.0.0 : 19736) - VIRTUAL - JMXConnectionServer/JEUSMP_server2
                                                - JNDI
                                                - JMSServiceChannel-default
                                                - JMXConnectionServer/JeusMBeanServer
                                                - SecurityServer
                                                - ClassFTP
                                                - FileTransfer
                                                - TransactionManager
                                - HTTP
                                http-server2 (plain, 0.0.0.0 : 9090) - HTTP

[2016.08.24 15:40:53][0] [server2-1] [SERVER-0242] Successfully started the server.
[2016.08.24 15:40:53][2] [server2-1] [SERVER-0248] The JEUS server is RUNNING.
[2016.08.24 15:40:53][2] [server2-1] [SERVER-0401] The elapsed time to start: 5486ms.
[2016.08.24 15:40:54][2] [launcher-10] [Launcher-0034] The server[server2] initialization completed successfully[pid :
7660].
[2016.08.24 15:40:54][0] [launcher-1] [Launcher-0040] Successfully started the server. The server state is now RUNNING.

```

出力メッセージの「Successfully started the server. The server state is now RUNNING.」は、JEUSのMSが起動されたことを意味します。

参考

1. 「Invalid License」メッセージはライセンスの異常を意味します。TmaxSoftからライセンスを取得し、%JEUS_HOME%\licenseディレクトリーにコピーします。
 2. 全フェーズの進行および環境変数が正常に設定されているのかを確認します。特に、「C:\TmaxSoft\JEUS8.0\bin」ディレクトリーがシステム・パス(startManagedServer.cmdスクリプトが実行)に含まれたのかを確認します。
 3. JEUS MSが正常に起動されると、エグザンプル(examples)がデプロイされます。これを確認するには、Webブラウザーでhttp://localhost:8088/examples/を呼び出します。
-

3. 以下は、jeusadminに接続する方法です。

```
C:\Documents and Settings\admin>jeusadmin -u administrator -p <password>
Attempting to connect to 127.0.0.1:9736.
The connection has been established to Domain Administration Server adminServer
in the domain jeus_domain.
JEUS8 Administration Tool
To view help, use the 'help' command.
[DAS]jeus_domain.adminServer>
```

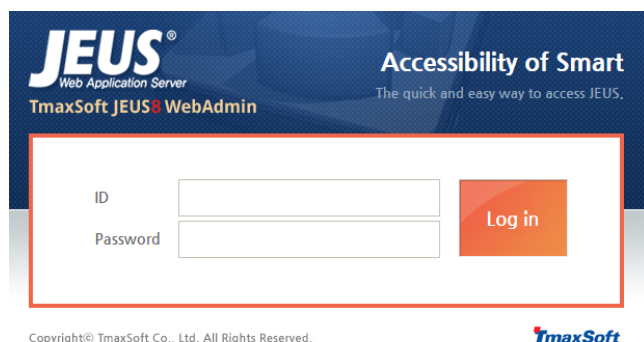
JEUS管理者のユーザー名とパスワードを入力します。通常、管理者アカウントはadministratorであり、パスワードはJEUSのインストール時に入力した値です。

4. しばらくしてからプロンプトが表示されますが、JEUSが正常に起動され、再びコマンドが実行できる状態になったことを意味します。
5. Webブラウザーを開いて、アドレスバーに以下のアドレスを入力します。

```
http://localhost:9736/webadmin
```

6. JEUS管理者のアカウントとパスワードを入力し、**[Log in]**ボタンをクリックします。通常、管理者アカウントはadministratorであり、パスワードはJEUSのインストール時に入力した値です。

[図 6.16] WebAdminのログイン画面



7. WebAdminを使用してJEUSを管理することができます。

【図 6.17】 WebAdminのメイン画面

The screenshot shows the JEUS WebAdmin interface. At the top, there's a header with the JEUS logo and navigation links: ホーム, 環境設定, サイトマップ, ログアウト. Below the header, the user is logged in as administrator on JEUS 8.0 (Fix#0) at adminServer(192.168.5.77:9736). The connection time is 2016-08-31 (Wed) PM 07:44:38 KST.

The left sidebar contains a navigation menu with the following items: jeus_domain, Domain, Session, Clusters, Servers, Applications, Security, Resources, Monitoring, Console, システム状態 (System Status), Runtime Info, and LOCK & EDIT. The System Status section shows a legend for server states: 0 Failed, 0 Standby, 1 Running, 1 Shutdown, 0 Suspended, and 0 Other.

The main content area is divided into several sections:

- Servers:** A table showing the status of servers. The table has columns for Server, Status, and Engine. The data is as follows:

Server	Status	Engine
adminServer(*)	RUNNING(00:04:03)	[Icons]
server1	SHUTDOWN	[Icons]
- DataSource:** Shows Database 0 and Cluster-ds 0.
- Applications:** Shows Application 1.
- Clusters:** Shows Clusters 0.
- Resource Manager:** A section with a list of resources: Message Bridge, Mail Source, External Source, and DataSource.
- Monitoring:** A section with a list of monitoring items: Thread, Transaction, MBean, Servers, and JNDI.

On the right side, there's a 'Domain Basic Information' panel for 'jeus_domain' with links for Domain Security Settings, User and Group Management, and Auto-lock Environment Settings. Below this is a 'Web Console Execution' button and a 'Node Settings' button.

The bottom of the page features the TmaxSoft logo and the copyright notice: Copyright (c) 2016 TmaxSoft Co., Ltd. All Rights Reserved.

8. jeusadminツールにログインした後、**local-start-server**と**local-shutdown**コマンドを使ってJEUSサーバーを制御することができます。JEUSサーバーを終了するには、**local-shutdown**コマンドを実行します。

```
[DAS]jeus_domain.adminServer>local-shutdown
The server [adminServer] has been shut down successfully.
```

9. jeusadminを終了するときは**exit**コマンドを実行します。

```
exit
```

6.4. サービスの登録と削除

JEUSは、Windows環境でのサービスとして動作できるように設計されています。この機能を使用するために、JEUSフォルダーに存在するサービスを実行します。本節では、Windows NTサービスとして動作するようにJEUSを登録および削除する手順について説明します。

6.4.1. サービスの登録

以下は、Windowsサービスを登録する手順です。

1. **svcinstall.exe**が存在する%JEUS_HOME%の下位のbinディレクトリーに移動します。

(デフォルト: C:\TmaxSoft\JEUS8.0\bin)

2. **svcinstall.exe**を実行して、WindowsにJEUSをサービスとして登録します。

```
svcinstall <jeus.ini> <id> <password>
```

%JEUS_HOME%\binの下位の<jeus.ini>ファイルは、JEUSをWindows NTサービスとして実行するときに使用する設定ファイルです。サービス名として、jeus.iniファイル名(jeusという名前で登録)が設定されます。このファイルには、JEUS_HOME、JAVA_HOMEなどの設定情報が保存されます。サービス設定ファイルについての詳細内容は、「[付録 C. Windowsサービスの設定](#)」を参照してください。

以下は、svcinstallコマンドの使用例です。

```
C:\TmaxSoft\JEUS8.0\bin>svcinstall jeus.ini administrator <password>
JEUS Windows service installer Ver. 4.1.0
Copyright (c) 2014, TmaxSoft Co., Ltd.

JEUS Service[jeuservice-jeus] installed
```

6.4.2. サービス登録の確認

サービスの登録は、以下の2つの方法で確認することができます。

- GUIモードでの登録確認

Windowsで、[スタート] > [設定] > [コントロールパネル] > [管理ツール] > [サービス]をクリックすると、サービス画面にてJEUSサービスの存在有無を確認することができます。

サービス名(デフォルト : JEUS8 Windows service)をダブルクリックすると、[開始]、[停止]、[一時停止]、[再開]ボタンが表示されます。[開始]ボタンをクリックするとサービスが開始されます。これは、インストール・ファイルに登録されたエンジンと、その他のコンポーネントがロードされることを意味します。[停止]ボタンをクリックすると、サービスが中断されます。

● コマンド・プロンプトでの登録確認

サービスを確認するために、コマンド・プロンプトで以下のように入力します。

```
svcremove -list
```

以下は、サービスを確認する例です。

```
C:\TmaxSoft\JEUS8.0\bin>svcremove -list
JEUS Windows service remover Ver. 4.1.0
Copyright (c) 2014, TmaxSoft Co., Ltd.

    ** JEUS Services **
-----
- jeus

Total count: 1
```

結果メッセージから、Windowsサービスが合計1つ登録されていることが確認できます。

6.4.3. サービスの削除

サービスを停止するには、Windowsで[スタート] > [設定] > [コントロールパネル] > [管理ツール] > [サービス]を選択し、[停止]ボタンをクリックします。

サービスを完全に削除するには、%JEUS_HOME%\binディレクトリーのsvcremove.exeを実行します。引数としてサービス名を使用します。

```
C:\TmaxSoft\JEUS8.0\bin> svcremove.exe jeus
JEUS Windows service remover Ver. 4.1.0
Copyright (c) 2014, TmaxSoft Co., Ltd.

Removing windows service 'jeus'...
Successfully done.
```

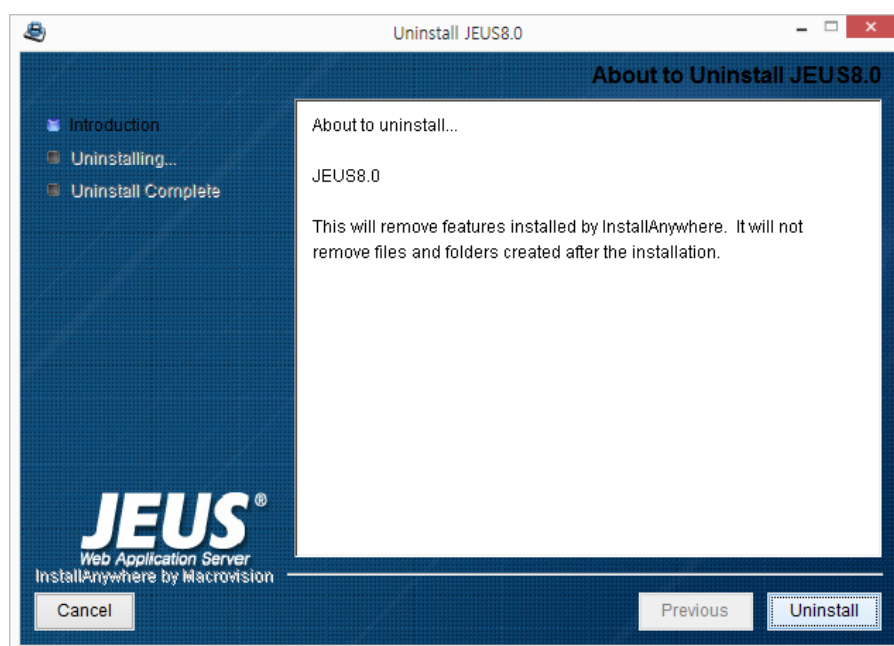
6.5. アンインストール

以下は、JEUSをアンインストールする手順です。

1. 「startDomainAdminServer、startManagedServer」スクリプトやWebAdminなど、すべてのJEUSプログラムを終了します。
2. JEUSホーム・ディレクトリーのUninstallerData\jeus8.0ディレクトリーでUninstall.exeを実行するか、(例: C:\TmaxSoft\JEUS8.0\UninstallerData\jeus8.0)、Windowsの[スタート] > [プログラム] > [TmaxSoft] > [JEUS8.0] > [Uninstall]を選択します。

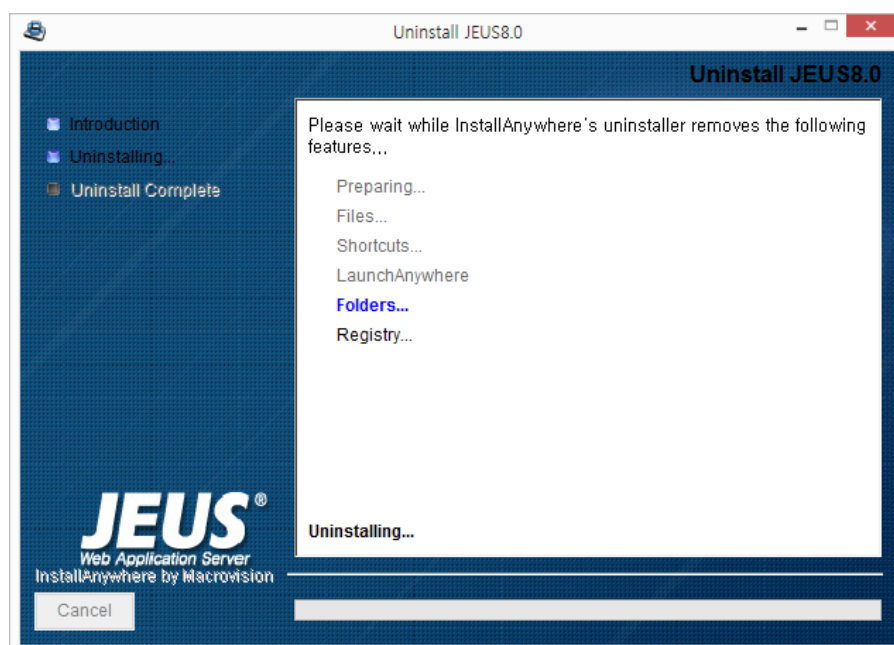
3. 以下のようにアンインストールの初期画面が表示されます。

[図 6.18] Windows環境でのアンインストール - アンインストールの初期画面



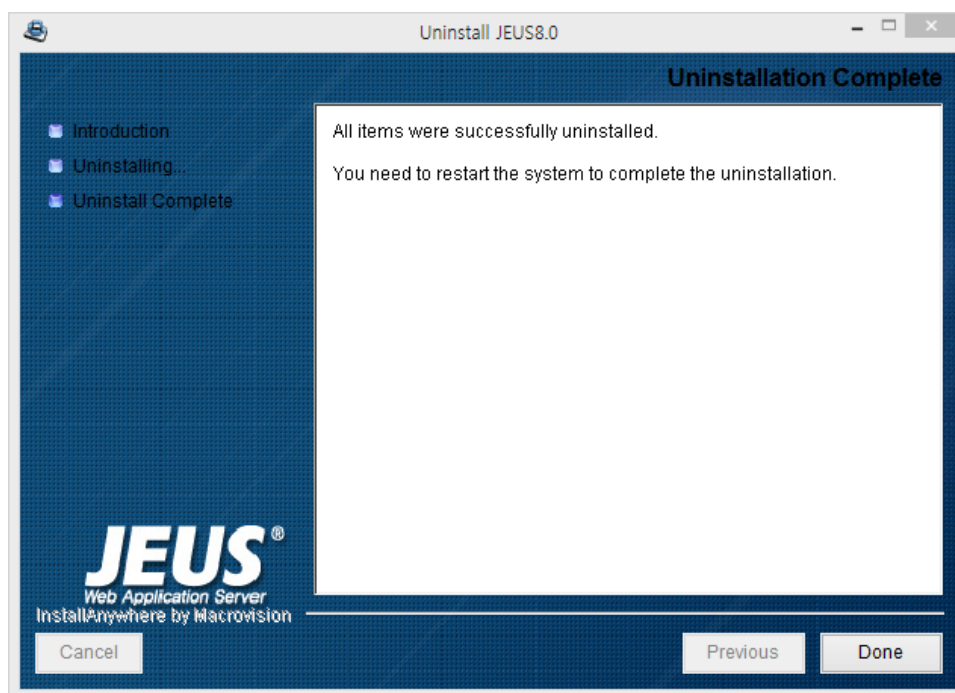
4. アンインストール画面のメッセージを確認した後、[Uninstall]ボタンをクリックします。JEUSのアンインストールを中止する場合は、[Cancel]ボタンをクリックします。
5. [Uninstall JEUS]画面でJEUSアンインストールの進行状況が確認できます。

[図 6.19] Windows環境でのアンインストール - Uninstall JEUS



6. **[Uninstall Complete]**画面が表示されるまで待ちます。

[図 6.20] Windows環境でのアンインストール - Uninstall Complete



参考

JEUSのインストールにより生成されたファイルはアンインストーラーで削除されません。これらのファイルはそのまま残すか、直接削除します。

7. **[Done]**ボタンをクリックして、JEUSのアンインストーラーを終了します。

8. 削除された内容を適用するために、サーバーを再起動するかログオフしてから再ログインします。

6.6. アンインストールの確認

以下の手順で正常にアンインストールされたのかを確認します。

• ファイルとディレクトリー

%JEUS_HOME%ディレクトリー(デフォルト値: C:\TmaxSoft\JEUS8.0)の下位ファイルとフォルダーが削除されたのかを確認します。JEUSのインストールにより作成されたファイルは削除されないため、直接削除します。

- ショートカット

インストール時に生成された「ショートカット」(プログラム・メニューの項目など)が削除されたのを確認します。

- 環境変数

環境変数が削除されたのを確認し、必要に応じてWindowsの[スタート] > [設定] > [コントロール・パネル] > [システム] > [詳細設定] > [環境変数]で直接削除します。Windowsコマンド・プロンプトで「set」コマンドを使って変数を確認することができます。JEUSのアンインストール後、サーバーを再起動した場合は更新された内容が表示されます。

- PATH

システムPATHからJEUSパスが削除されます。(例: C:\TmaxSoft\JEUS8.0).

Part III. JEUS スタートガイド

第7章 概要

本章は、TmaxSoftのWeb Application Server(WAS)のJEUSを使用してプログラムを初めて作成するユーザーを対象としており、その構成は以下のとおりです。

- **システムの設定**

JEUSの基本的な環境設定と運用方法について説明します。

- **Web層の使用**

Webアプリケーションのパッケージング方法について説明し、例を使用してサーブレット、JSP、JSTL、JSFページを実行します。

- **EJBの使用**

EJBモジュールのパッケージングおよびデプロイ方法、ステートレス・セッションBeanとJava Persistence APIの使用方法について説明します。

- **Webサービスの使用**

Webサービスのパッケージングおよびデプロイ方法、サーブレット・ベースのエンドポイントとEJBベースのエンドポイントの使用方法について説明します。

JEUS Webアプリケーション・サーバーを完全に理解するには、説明内容を直接実行してみることをお勧めします。

参考

さらに詳しい技術的な内容については、『JEUS サーバガイド』、『JEUS EJBガイド』、『JEUS Webエンジンガイド』などを参照してください。WebAdminツールをより効果的に使用するには、『JEUS WebAdminガイド』を参照してください。

第8章 システムの設定

本章では、JEUSの環境設定および起動方法について説明します。

8.1. 概要

WebAdminを使用してシステムを構成する方法について、以下の順で説明します。

- 基本環境の設定
- MS(Managed Server)の追加と設定方法
- データソースの追加

8.2. 基本環境の構成

JEUS WebAdminは、Webを通じてJEUSのすべての要素が管理できるサービスを提供します。ユーザーはJEUSにアクセスし、システム設定、モニタリング、アプリケーションを容易に管理することができます。

以下の順にWebAdminを実行します。

1. コマンド・プロンプトでstartDomainAdminServerファイルを実行し、JEUS DASを実行します。

参考

startDomainAdminServerスクリプトはJEUS_HOME/bin/ディレクトリーに存在しており、システム・パスに設定されている必要があります。WebAdminを使用するには、domain.xmlに<enable-webadmin>が設定されている必要があります。詳細内容については、『JEUS WebAdminガイド』の「2.2. ログイン」を参照してください。

以下は、JEUS DASを実行する例です。

```
C:\TmaxSoft\JEUS8.0\bin>startDomainAdminServer -u administrator -p <password>
*****
- JEUS Home           : C:\TmaxSoft\JEUS8.0
- Added Java Option  :
- Java Vendor        : Sun
*****

C:\Users\paper>"D:\java\jdk1.7.0_80\jre\..\bin\java" -server -Xmx512m
-Xbootclasspath/p:"C:\TmaxSoft\JEUS8.0\lib\system\extension.jar"
-classpath "C:\TmaxSoft\JEUS8.0\lib\system\bootstrap.jar" -Dsun.rmi.dgc.client.gcInterval=3600000
-Dsun.rmi.dgc.server.gcInterval=3600000
-Djeus.jvm.version=hotspot -Djeus.home="C:\TmaxSoft\JEUS8.0" -Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
```

```

-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url -Djava.library.path="C:\TmaxSoft\JEUS8.0\lib\system"
-Djava.endorsed.dirs="C:\TmaxSoft\JEUS8.0\lib\endorsed" -Djava.util.logging.manager=jeus.util.logging.JeusLogManager
-Djava.util.logging.config.file="C:\TmaxSoft\JEUS8.0\bin\logging.properties"
-Djeus.properties.replicate=jeus,java.util.logging,sun.rmi.dgc
-Djava.net.preferIPv4Stack=true jeus.server.LauncherBootstrapper -u administrator -p <password>

===== JEUS LICENSE INFORMATION =====
=== VERSION : JEUS 8.0 (Fix#0) (8.0.0.0-b27)
=== EDITION: Enterprise (Trial License)
=== NOTICE: This license restricts the number of allowed clients.
=== Max. Number of Clients: 5
=====

[2016.08.24 15:39:19][2] [launcher-1] [Launcher-0012] Starting the server [server1] with the command
D:\java\jdk1.7.0_80\jre\..\bin\java -Dserver1 -Xmx512m -XX:MaxPermSize=256m -Djeus.ejb.enable.configDeleteOption=true
-server
-Xbootclasspath/p:C:\TmaxSoft\JEUS8.0\lib\system\extension.jar -classpath C:\TmaxSoft\JEUS8.0\lib\system\bootstrap.jar
-Djava.security.policy=C:\TmaxSoft\JEUS8.0\domains\jeus8\config\security\policy
-Djava.library.path=C:\TmaxSoft\JEUS8.0\lib\system
-Djava.endorsed.dirs=C:\TmaxSoft\JEUS8.0\lib\endorsed -Djeus.properties.replicate=jeus,java.util.logging,sun.rmi.dgc
-Djeus.jvm.version=hotspot
-Djava.util.logging.config.file=C:\TmaxSoft\JEUS8.0\bin\logging.properties -Dsun.rmi.dgc.server.gcInterval=3600000
-Djava.util.logging.manager=jeus.util.logging.JeusLogManager -Djeus.home=C:\TmaxSoft\JEUS8.0
-Dsun.rmi.dgc.client.gcInterval=3600000
-Djeus.domain.name=jeus8 -Djava.naming.factory.initial=jeus.jndi.JNSContextFactory
-Djava.naming.factory.url.pkgs=jeus.jndi.jns.url
-Djeus.server.protectmode=false -XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput
-XX:LogFile=C:\TmaxSoft\JEUS8.0\domains\jeus8\servers\server1\logs\jvm.log jeus.server.admin.DomainAdminServerBootstrapper
-u administrator -domain jeus8 -server server1 .
[2016.08.24 15:39:19][2] [launcher-1] [Launcher-0014] The server[server1] is being started ...
[2016.08.24 15:39:21][2] [server1-1] [SERVER-0248] The JEUS server is STARTING.
[2016.08.24 15:39:21][0] [server1-1] [SERVER-0000] Version information - JEUS 8.0 (Fix#0) (8.0.0.0-b27)

... 中略

[2016.08.24 15:39:31][2] [server1-1] [WEB-3413] The web engine is ready to receive requests.
[2016.08.24 15:39:31][2] [server1-1] [UNIFY-0100] Listener information
                                BASE (plain, 0.0.0.0 : 9736) - VIRTUAL - SecurityServer
                                                                - ClassFTP
                                                                - JMSServiceChannel-channel_oneport
                                                                - JMXConnectionServer/JeusMBeanServer
                                                                - JMSServiceChannel-channel_noconf
                                                                - JMXConnectionServer/JEUSMP_server1
                                                                - FileTransfer
                                                                - JNDI
                                                                - TransactionManager
                                                                - BootTimeFileTransfer
                                                                - JMSServiceChannel-channel_security
                                - RMI
                                - HTTP
                                http-server (plain, 0.0.0.0 : 9099) - HTTP
                                jms-jmstest (plain, 0.0.0.0 : 9931) - VIRTUAL - JMSServiceChannel-jmstest
jms-channel_serverport (plain, 0.0.0.0 : 39742) - VIRTUAL - JMSServiceChannel-channel_serverport
                                                                - JMSServiceChannel-channel_serverurl
                                jms-channel_port0 (plain, 0.0.0.0 : 0) - VIRTUAL - JMSServiceChannel-channel_port0
                                jms-channel_nio (plain, 0.0.0.0 : 9932) - VIRTUAL - JMSServiceChannel-channel_nio
                                jms-channel_clientlimit (plain, 0.0.0.0 : 9933) - VIRTUAL - JMSServiceChannel-channel_clientlimit

[2016.08.24 15:39:31][0] [server1-1] [SERVER-0242] Successfully started the server.
[2016.08.24 15:39:31][2] [server1-1] [SERVER-0248] The JEUS server is RUNNING.
[2016.08.24 15:39:31][2] [server1-1] [SERVER-0401] The elapsed time to start: 12254ms.
[2016.08.24 15:39:31][2] [launcher-10] [Launcher-0034] The server[server1] initialization completed successfully[pid :
11216].
[2016.08.24 15:39:31][0] [launcher-1] [Launcher-0040] Successfully started the server. The server state is now RUNNING.

```

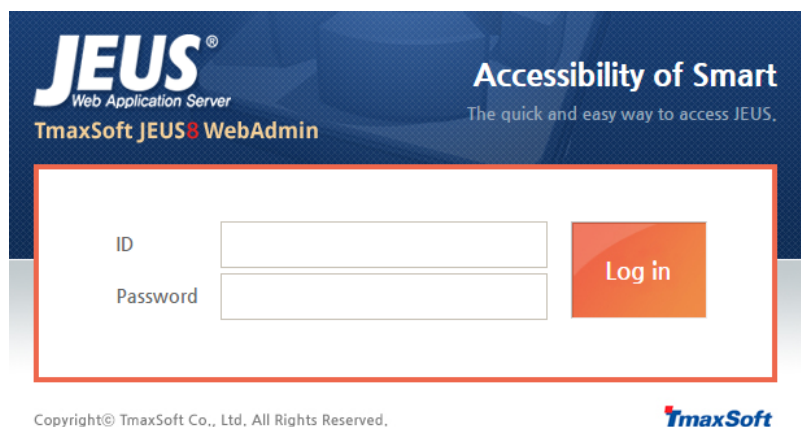
2. Webブラウザーを開いてWebAdminに接続し、JEUSのインストール時に設定したログインIDとパスワードを入力します。

```
http://<localhost>:9736/webadmin
```

参考

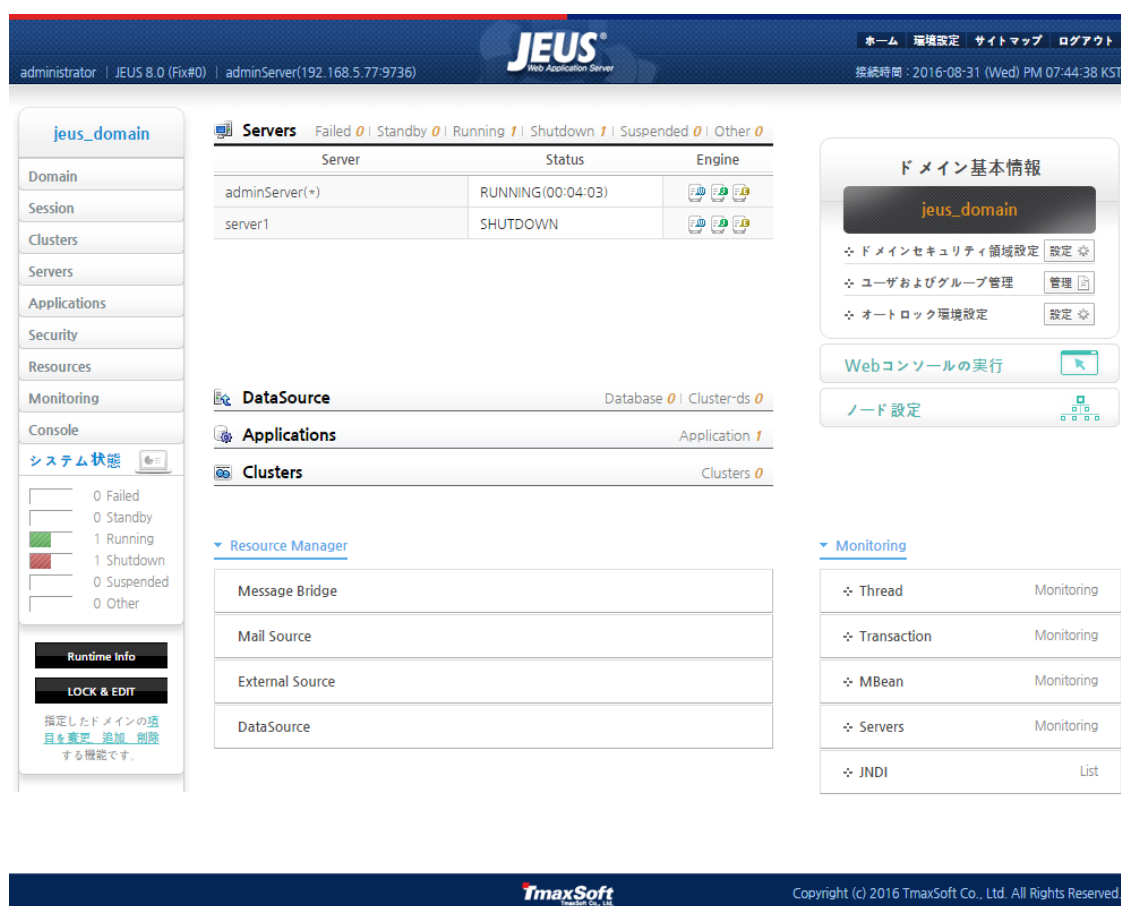
DASのベース・ポートは9736であり、WebAdminはベース・ポートを使って接続します。

[図 8.1] ログイン画面



3. 認証に成功したら、以下のようなWebAdminのメイン画面が表示されます。

[図 8.2] メイン画面



8.3. MSの追加と設定

MSとは、実際にアプリケーションをサービスするためのエンジンと様々なサービスを管理するサーバー・インスタンスのことをいいます。MSはドメインに複数存在することができます。MSの主な役割はユーザーがデプロイするアプリケーションをサービスし、アプリケーションが必要としているリソースやサービスを提供することです。

MSの追加

以下は、新しいMSを追加し、追加されたMSにリスナーを追加する方法です。

1. WebAdmin画面の左側のメニューから[**Servers**]を選択すると、サーバー・リストの照会画面が表示されます。サーバーを追加するときは、[**Add**]ボタンをクリックします。

【図 8.3】 MSの追加 - サーバー・リストの照会画面



参考

設定内容を動的に反映するには、[**LOCK & EDIT**]ボタンをクリックして設定変更モードに切り替えます。設定変更モードに変更しないと[**Add**]ボタンは有効になりません。

2. [Server]画面で各項目を設定してから、[確認]ボタンをクリックします。

[図 8.4] MSの追加 - 基本情報の設定

The screenshot shows the JBoss JMS console interface. On the left is a sidebar with a tree view containing 'jeus_domain' and various sub-items like Domain, Session, Clusters, Servers, Applications, Security, Resources, Monitoring, and Console. Below the tree is a 'システム状態' (System Status) section with a legend for states: 0 Failed, 0 Standby, 1 Running, 1 Shutdown, 0 Suspended, and 0 Other. The main area is titled 'Server' and contains a message: 'ドメイン内で使用するJEUSサーバの詳細設定を定義します。' (Define the detailed settings for the JEUS server to be used in the domain). Below this is a tabbed interface with 'Basic', 'Resource', and 'Engine' tabs. The 'Basic' tab is active, showing fields for 'Name' (set to 'server2'), 'Log Home', 'Node Name', 'Group', 'Action On Resource Leak', 'Data Source Remote Lookup', and 'Engine Init On Startup'. A '確認' (Confirm) button is visible at the top right of the form.

3. 設定内容を動的に反映するため、画面の左側の[Activate Changes]ボタンをクリックします。
4. サーバーに反映されると、以下のような結果メッセージが出力されます。動的設定でMSが生成されたことが確認できます。

[図 8.5] MSの追加 - サーバー追加結果の確認

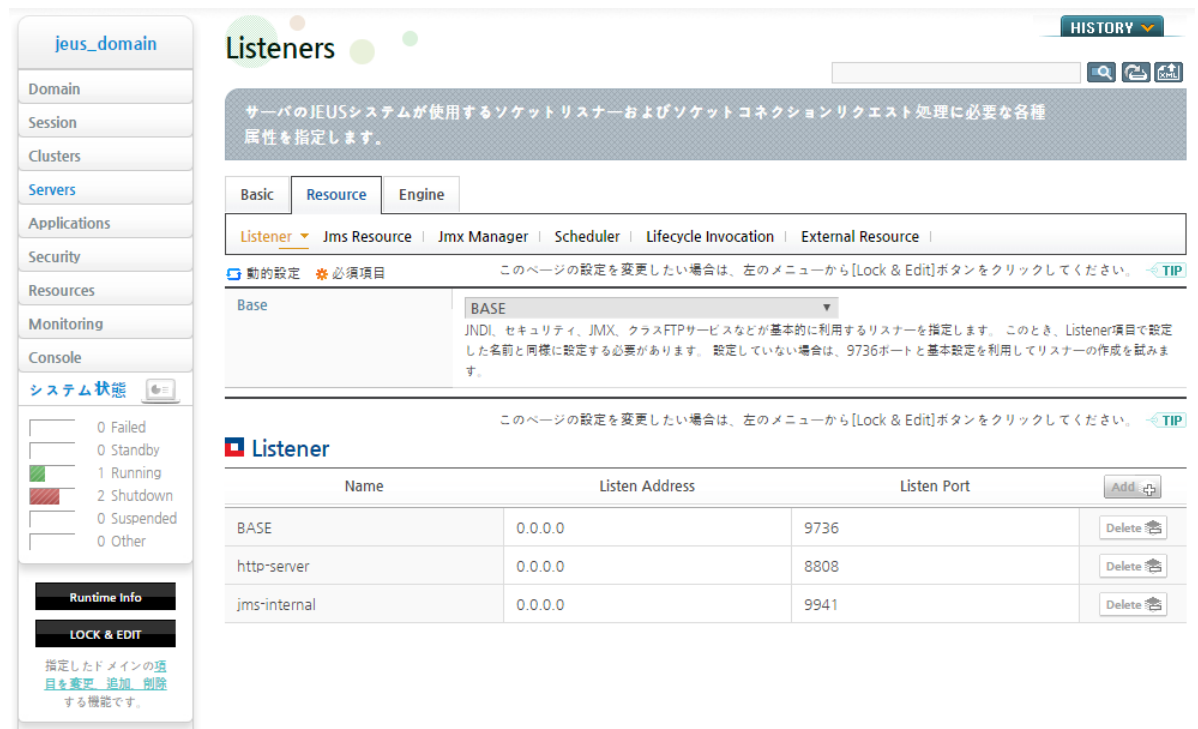
This screenshot shows the same JBoss JMS console interface as Figure 8.4, but after the configuration has been applied. The 'Basic' tab is still active. A message box at the top of the main area displays the following log output: 'domain.xmlの設定を動的に適用しました。', 'domain.xml : ACTIVATED', 'servers.server : ACTIVATED', 'servers.server:{? name == 'server2'} : ACTIVATED', 'previous value : null, edited value : jeus.xml.binding.jeusDD.ServerType@1d4fd4a, result value : jeus.xml.binding.jeusDD.ServerType@1d4fd4a'. Below the message box, the 'Auto Generated' checkbox is checked. The 'Name' field is still set to 'server2'. The '確認' (Confirm) button is no longer visible, indicating the configuration is now active.

5. MSを起動するとき「BASE」というリスナーを使用しますが、デフォルト値が「9736」に設定されます。しかし、この値はDASの値と同じであるため、正常に起動されない場合があるので変更する必要があります。

WebAdminの左側のメニューから[**Servers**]を選択すると、サーバー・リストの照会画面が表示されます。

照会されたサーバー・リストから目的のサーバーを選択し、[**Resource**] > [**Listener**]メニューを選択します。[**Listeners**]画面で「BASE」リスナーをクリックすると、詳細情報ページに移動します。

[図 8.6] MSの追加 - リスナーの選択



jeus_domain

Domain

Session

Clusters

Servers

Applications

Security

Resources

Monitoring

Console

システム状態

0 Failed

0 Standby

1 Running

2 Shutdown

0 Suspended

0 Other

Runtime Info

LOCK & EDIT

指定したドメインの項目を変更、追加、削除する機能です。

Listeners

HISTORY

サーバのJEUSシステムが使用するソケットリスナーおよびソケットコネクションリクエスト処理に必要な各種属性を指定します。

Basic Resource Engine

Listener Jms Resource Jmx Manager Scheduler Lifecycle Invocation External Resource

動的設定 必須項目 このページの設定を変更したい場合は、左のメニューから[Lock & Edit]ボタンをクリックしてください。TIP

Base BASE

JNDI、セキュリティ、JMX、クラスFTPサービスなどが基本的に利用するリスナーを指定します。このとき、Listener項目で設定した名前と同様に設定する必要があります。設定していない場合は、9736ポートと基本設定を利用してリスナーの作成を試みます。

このページの設定を変更したい場合は、左のメニューから[Lock & Edit]ボタンをクリックしてください。TIP

Listener

Name	Listen Address	Listen Port	Add
BASE	0.0.0.0	9736	Delete
http-server	0.0.0.0	8808	Delete
jms-internal	0.0.0.0	9941	Delete

6. 「Listen Port」項目を「9736」以外の値に設定して、[確認]ボタンをクリックします。

[図 8.7] MSの追加 - リスナー設定の変更

The screenshot shows the JBoss WebAdmin interface for configuring a Listener. The left sidebar has a menu with items like Domain, Session, Clusters, Servers (selected), Applications, Security, Resources, Monitoring, and Console. Below the menu is a 'システム状態' (System Status) section showing various status indicators. The main content area is titled 'Listener' and has tabs for Basic, Resource (selected), and Engine. Below the tabs is a breadcrumb trail: Listener > Jms Resource > Jmx Manager > Scheduler > Lifecycle Invocation > External Resource. A status bar indicates '動的設定' (Dynamic Configuration) and '必須項目' (Required Fields). The form contains the following fields:

Field	Value	Description
Name	BASE	統合ポートリスナーを示す固有のIDを設定します。別の設定でこのリスナーを参照するときに使用します。
Listen Address	0.0.0.0	複数のIPが割り当てられている場合、統合ポートリスナーがバインドされるサービスIPアドレスを設定します。設定していない場合は、すべてのIPアドレスを利用してサービスが行われます。仮想マルチキャストを使用する場合は、ベースリスナーのIPアドレスを必ず指定します。
Listen Port	39736	[デフォルト: 9736] 統合ポートリスナーのポートを設定します。設定されていない場合は常に9736を使用するため、複数のサーバを同じマシンで使用する場合には必ず指定します。
Selectors	1	[デフォルト: 1] 統合ポートリスナーのI/O処理に使用するセクタ数を設定します。一般的にCPUのコア数で設定します。デフォルト値は1です。
Use Dual Selector	<input type="checkbox"/>	[デフォルト: false] 統合ポートリスナーの読み書きに使用するセクタを分離するか否かを設定します。大量の読み書きが同時に必要な場合に設定してください。
Backlog	128	[デフォルト: 128] 統合ポートリスナーのバックログ値を設定します。

At the bottom right of the form are two buttons: '確認' (Confirm) and '再設定' (Reset).

7. 画面左側の[Activate Changes]ボタンをクリックして、サーバーに設定内容を反映します。

HTTPリスナーおよびコネクタの設定

新たに追加されたMSを起動するための設定は基本設定だけで十分です。Webエンジンを利用してサービスする場合は、追加的にリスナーおよびコネクタの設定を行います。

1. WebAdminの左側のメニューから[Servers]を選択すると、サーバー・リストの照会画面が表示されます。

照会されたサーバー・リストから目的のサーバーを選択し、[Resource] > [Listener]メニューを選択します。[Listeners]画面で[Add]ボタンをクリックします。

【図 8.8】HTTPリスナーの設定 - リスナーの追加

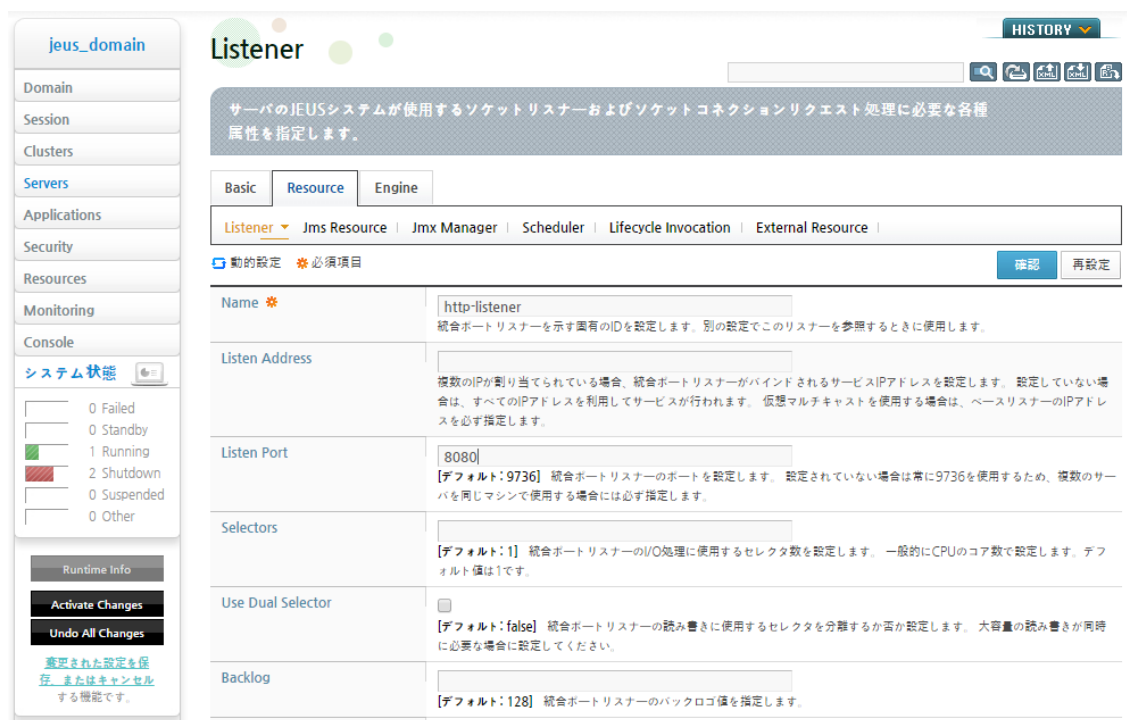


参考

設定内容を動的に反映するには、**[LOCK & EDIT]**ボタンをクリックして設定変更モードに切り替えます。設定変更モードに変更しないと**[Add]**ボタンは有効になりません。

2. **[Listener]**画面で「Name」と「Listen Port」などの値を設定してから、**[確認]**ボタンをクリックします。

【図 8.9】HTTPリスナーの設定 - 基本情報の設定

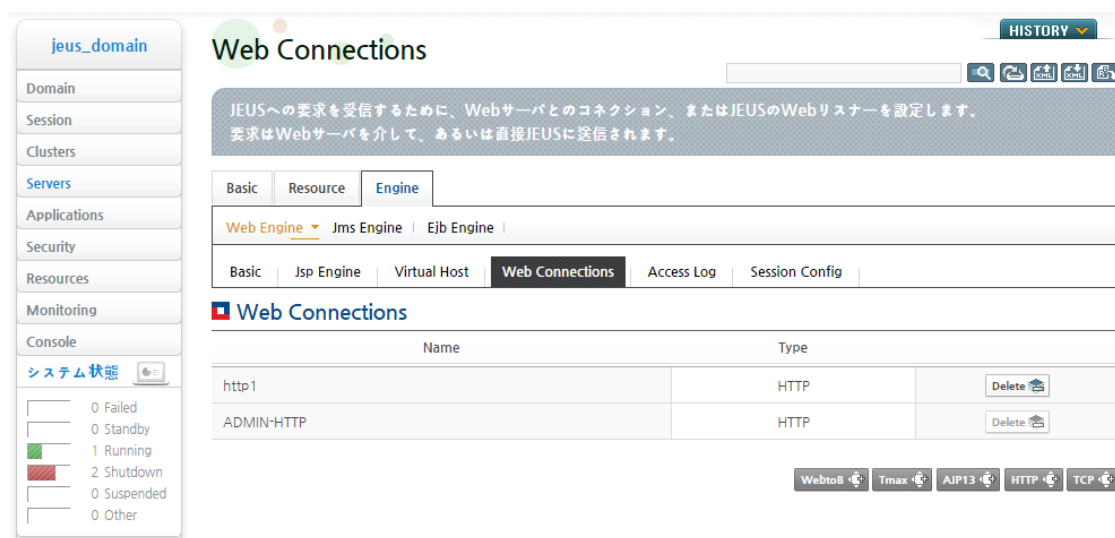


参考

別のリスナーで使用しているポートと重複されないように設定します。

3. **[Engine] > [Web Engine] > [Web Connections]**メニューを選択し、**[Web Connections]**画面へ移動します。本例では、HTTPリスナーの設定について記述しているので、**[HTTP]**ボタンをクリックします。

[図 8.10] HTTPリスナーの設定 - Webコネクションの追加



4. **[Http Listener]**画面で「Name」と「Server Listener Ref」などを設定します。「Server Listener Ref」は、以前設定した「http-listener」を選択します。設定が完了したら、**[確認]**ボタンをクリックします。

[図 8.11] HTTPリスナーの設定 - Webコネクションの基本設定



5. 設定内容を動的に反映するため、画面の左側の[**Activate Changes**]ボタンをクリックします。

WebtoBコネクターの設定

JEUS 8Iには、JEUS 6で組み込みWebtoBを起動していたWSエンジンが存在しません。したがって、組み込みWebtoBを使用するためには、WebtoB環境ファイルをコンパイルし、組み込みWebtoBを直接起動する手順と、WebtoBコンテナを追加して組み込みWebtoBと接続する手順が必要です。

1. 組み込みWebtoBの環境ファイル(\$JEUS_HOME/webserver/config/ws_engine.m)をコンパイルします。

参考

デフォルトでインストールされている組み込みWebtoBの環境ファイルはコンパイルされていないため、コンパイルの後に起動します。

```
C:\TmaxSoft\JEUS8.0\webserver\config>wscfl -i ws_engine.m

Current configuration:
    Number of client handler(HTH) = 1
    Supported maximum user per node = 2018
    Supported maximum user per handler = 2018

Successfully created the configuration file
(C:/TmaxSoft/JEUS8.0/webserver/config/wsconfig) for node paper-PC.
The host name of the running machine is paper-PC.

C:\TmaxSoft\JEUS8.0\webserver\config>
```

2. 組み込みWebtoBを起動します。

```
C:\TmaxSoft\JEUS8.0\webserver\config>wsboot

Booting WebtoB on node (paper-PC)
Starting WSM at 04/30/13 11:06:56
Starting HTL at 04/30/13 11:06:56
Starting HTH at 04/30/13 11:06:56
    Current WebtoB Configuration:
        Number of client handlers (HTH) = 1
        Supported maximum user per node = 2018
        Supported maximum user per handler = 2018
Starting SVR(C:/TmaxSoft/JEUS8.0/webserver/bin/htmls.exe) at 04/30/13 11:06:56
Starting SVR(C:/TmaxSoft/JEUS8.0/webserver/bin/cgis.exe) at 04/30/13 11:06:56
Starting SVR(C:/TmaxSoft/JEUS8.0/webserver/bin/ssis.exe) at 04/30/13 11:06:56
```

```
C:\TmaxSoft\JEUS8.0\webserver\config>
```

3. WebAdminの左側のメニューから[**Servers**]を選択すると、サーバー・リストの照会画面が表示されます。

照会されたサーバー・リストから目的のサーバーを選択し、[**Engine**] > [**Web Engine**] > [**Web Connections**]メニューを選択して[**Web Connections**]画面に移動します。本例では、WebtoBコネクタの設定について説明するので、[**WebtoB**]ボタンをクリックします。

[図 8.12] WebtoBコネクタの設定 - Webコネクションの追加

The screenshot displays the 'Web Connections' configuration page. On the left, a sidebar menu has 'Servers' highlighted. The main content area has a 'Web Connections' title and a 'HISTORY' dropdown. Below this is a message box explaining the purpose of the connections. A tabbed interface shows 'Basic', 'Resource', and 'Engine' tabs, with 'Engine' selected. Under 'Engine', there are sub-tabs for 'Web Engine', 'Jms Engine', and 'Ejb Engine'. The 'Web Engine' sub-tab is active, showing further sub-tabs: 'Basic', 'Jsp Engine', 'Virtual Host', 'Web Connections', 'Access Log', and 'Session Config'. The 'Web Connections' sub-tab is selected, displaying a table with columns 'Name' and 'Type'. The table lists three connections: 'http1' (HTTP), 'ADMIN-HTTP' (HTTP), and 'http' (HTTP). Each row has a 'Delete' button. Below the table are buttons for 'WebtoB', 'Tmax', 'AJP13', 'HTTP', and 'TCP'. At the bottom left, a 'Runtime Info' section shows system status (0 Failed, 0 Standby, 1 Running, 2 Shutdown, 0 Suspended, 0 Other) and buttons for 'Activate Changes' and 'Undo All Changes'. A note at the bottom states: '変更された設定を保存、またはキャンセルする機能です。' (Function to save or cancel changed settings.)

参考

設定内容を動的に反映するには、[**LOCK & EDIT**]ボタンをクリックして設定変更モードに切り替えます。設定変更モードに変更しないと[**WebtoB**]ボタンは有効になりません。

4. [WebtoB Connector]画面で「Name」、「WebtoB Home」、「Thread Pool Number」などを設定します。設定が完了したら、[確認]ボタンをクリックします。

【図 8.13】 WebtoBコネクターの設定 - Webコネクションの基本設定

jeus_domain

Domain

Session

Clusters

Servers

Applications

Security

Resources

Monitoring

Console

システム状態

0 Failed

0 Standby

1 Running

2 Shutdown

0 Suspended

0 Other

Runtime Info

Activate Changes

Undo All Changes

変更された設定を保存、またはキャンセルする機能です。

WebtoB Connector

HISTORY

WebtoBと連動するためのコネクタです。コネクションを作成するとき、JEUSがクライアントになります。

Basic Resource Engine

Web Engine Jms Engine Ejb Engine

Basic Jsp Engine Virtual Host Web Connections Access Log Session Config

動的設定 * 必須項目

確認 再設定

Name * webtoB1
リスナーまたはコネクタを識別できる一意の名前を指定します。

Output Buffer Size byte
out.write()メソッドにより出力されるJSP/Servlet出力を一時保存する出力バッファのサイズを指定します。バッファがいっぱいになると、自動でクライアントにデータを送信します。デフォルト値は、AJP13の場合は8192です。他のプロトコルの場合は、new Socket().getSendBufferSize()に従いますが、最小8192、最大16384の範囲になります。

Wjvp Version 1
[デフォルト: 2] WebtoBと接続するときに使用するWJPプロトコルのバージョンを決定します。1、2のみ設定できます。WebtoB 4.1.6以下の場合はバージョン2をサポートしないため、1に設定する必要があります。

Registration Id *
WebtoBサーバと接続するときに使用する登録IDを設定します。WebtoBと約束されたプロトコルにより、最大15文字の制限があります。

Hth Count
[デフォルト: 1] HTHプロセスの数を設定します。WebtoBサーバのHTHプロセス数と同一に設定します。

Read Timeout ms
[デフォルト: 120000] 設定時間の間WebtoBからメッセージが送信されなかった場合は、コネクションに問題があると見なし、新たにコネクションを確立します。WebtoBとJEUS間にファイアウォールがある場合は、WebtoBが送信するPingメッセージの到着の可否をこの設定によりチェックできます。WebtoBのsvrchtimelimitより大きく設定する必要があります。0より大きく、1秒より小さい値を設定した場合は、無条件で1秒と見なします。-1の場合は、OSの設定に従います。

Network Address
WebtoBのTCP/IPアドレス情報を設定します。

Port *
WebtoBに接続するポートを指定します。このポート番号は、WebtoB設定ファイルのJSVPORT値と一致する必要があります。

Ip Address
[デフォルト: localhost] WebtoBのIPアドレスを指定します。

Domain Socket Address
UNIXドメインソケット情報、あるいはWindowsでHTHプロセスとのIPC通信を行うための情報を設定します。WebtoBがWebエンジンと同じマシンにある場合は、UNIXドメインソケット(パイプ)を介して相互通信します。ただし、Windowsでは一般のソケット通信を行います。

WebtoB Ipcbaseport
[デフォルト: 6666] WindowsでWebtoB HTHプロセスとIPC通信を行うためのポートを指定します。この設定値はOS環境変数のWEBTOB_IPCBASEPORTをオーバーライドします。

WebtoB Home
[C:\WtmaxSoft\WJEUS8.0\wwebserver] WebtoBのホームディレクトリを設定します。この設定値はWebtoBホームディレクトリを示すOS環境変数(JEUS_WSDIRまたはWEBTOBDIR)をオーバーライドします。

参考

「Thread Pool」項目の「Number」を設定するときは、WebtoBの設定を参考にして設定します。インストーラーを使ってインストールした後、組み込みWebtoBの設定を変更していない場合、スレッド・プール数はデフォルト値の5を使用します。

5. 設定内容を動的に反映するため、画面の左側の[Activate Changes]ボタンをクリックします。

6. WebtoBコネクターの設定を適用するためにJEUSを再起動します。

8.4. データソースの追加

データソースは、データベースとJava EEプログラムを接続するために使用されます。

以下の例では、JEUSにデフォルトで含まれているApache Derbyデータベースを使用します。Apache Derbyは、JEUS_HOME/derbyに含まれています。Derbyが実行されていない場合は、以下のように実行します。

UNIX/LinuxまたはWindowsのコマンド・プロンプトで以下のように実行します。

```
JEUS_HOME\bin> startderby
```

以降、Derbyを終了するときは以下のとおり実行します。

```
JEUS_HOME\bin> stopderby
```

参考

JEUSでDerbyを使用するには、DerbyのJDBCドライバー・ファイルであるderbyclient.jarがJEUS_HOME\lib\datasourceに存在する必要があります。(デフォルトで含まれています)

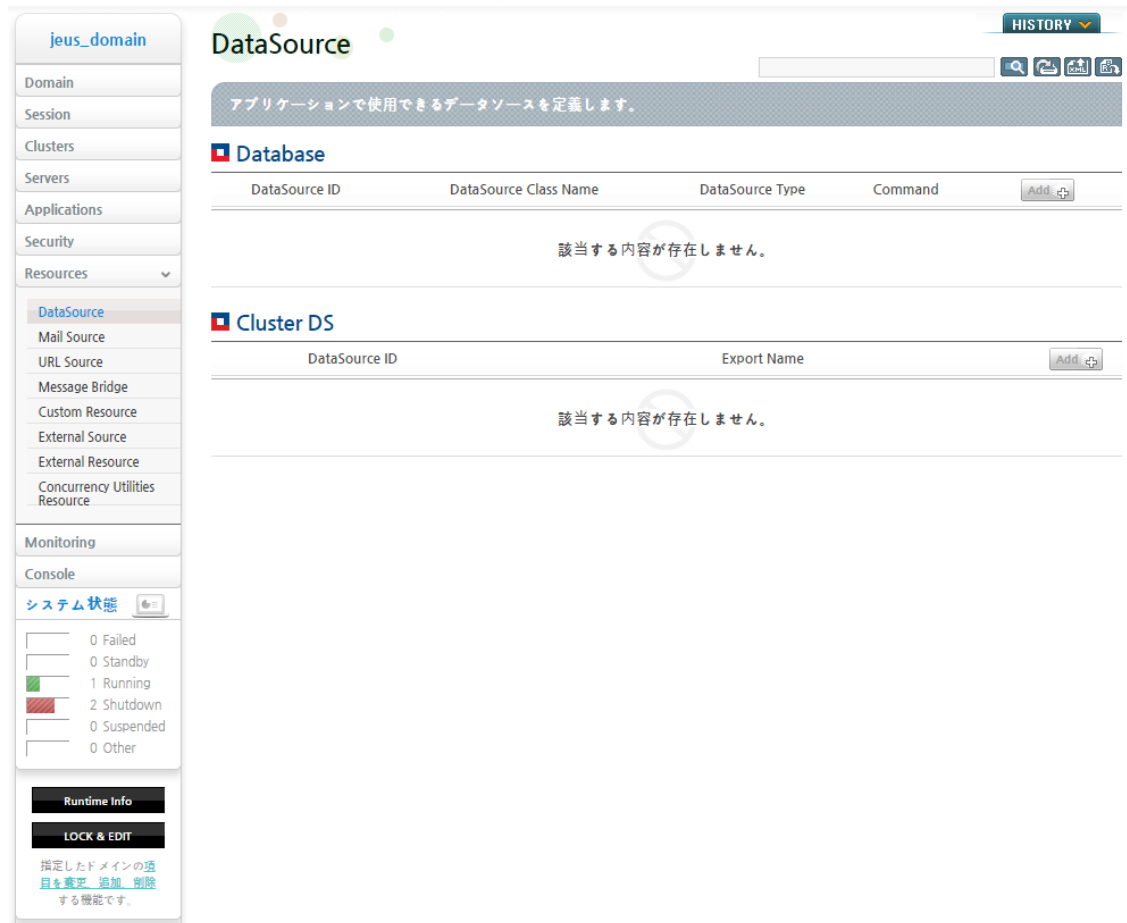
Derbyについての詳細内容は、<http://db.apache.org/derby/>を参照してください。

例では、sampleというデータベースをjdbc/sampleというデータソース名として使用します。

以下は、データソースを追加する方法についての説明です。

1. WebAdminの左側のメニューから[Resource] > [Data Source]を選択すると、[Data Source]画面が表示されます。データソースを追加するため、「Database」メニューの[Add]ボタンをクリックします。

[図 8.14] データソースの追加 - JDBCデータ・リスト



参考

設定内容を動的に反映するには、[LOCK & EDIT]ボタンをクリックして設定変更モードに切り替えます。設定変更モードに変更しないと[Add]ボタンは有効になりません。

2. [Database]画面の基本情報を設定してから、[確認]ボタンをクリックします。

[図 8.15] データソースの追加 - 基本情報の設定

jeus_domain

Domain

Session

Clusters

Servers

Applications

Security

Resources

DataSource

Mail Source

URL Source

Message Bridge

Custom Resource

External Source

External Resource

Concurrency Utilities Resource

Monitoring

Console

システム状態

0 Failed

0 Standby

1 Running

2 Shutdown

0 Suspended

0 Other

Database

JDBCドライバのデータソースインスタンスを作成し、コネクションプールを構成するための特性が含まれています。

Basic Connection Pool

動的設定 * 必須項目

Data Source ID * datasource1
データソースのIDを設定します。1つのドメインにおいてデータソースIDは、データソースの一意の識別子として動作するように設定します。

Export Name jdbc/sample
データソースのJNDI名を設定します。異なる2つのデータソースが異なるサーバのJNDIにバインドされることが保証できるなら、該当するデータソースは同じJNDI名が指定可能です。これは、任意のサーバにおいて同じJNDI名を持つ異なるデータソースを許可しないことを意味します。設定していない場合は、データソースIDをJNDI名として使用します。

Vendor others
JDBCドライバベンダの名前を指定します。

Data Source Class Name * org.apache.derby.jdbc.ClientConnectionPoolDataSource
JDBCドライバデータソースクラスの名前を指定します。パッケージ名を含む完全修飾名で記述します。

Data Source Type * ConnectionPoolDataSource
データソースのタイプを指定します。DATA_SOURCEに設定すると、コネクションプーリングサービスは提供されません。CONNECTION_POOL_DATA_SOURCEに設定するとコネクションプーリングサービスが提供されます。XA_DATA_SOURCEに設定した場合は、コネクションプーリングサービスと共にXA運動がサポートされます。

Server Name
DBが実行されるホスト名、またはIPを設定します。

Port Number
DBリスナーのポート番号を設定します。

Database Name

確認 再設定

3. 以下のように、生成されたデータソースがデータソース・リストに表示されます。

[図 8.16] データソースの追加 - データソース・リストの照会

jeus_domain

Domain

Session

Clusters

Servers

Applications

Security

Resources

DataSource

Mail Source

URL Source

Message Bridge

Custom Resource

External Source

External Resource

Concurrency Utilities Resource

Monitoring

Console

DataSource

アプリケーションで利用できるデータソースを定義します。

追加されました。

Database

DataSource ID	DataSource Class Name	DataSource Type	Command	Add
datasource1	org.apache.derby.jdbc.ClientConnectionPoolDataSource	CONNECTION_POOL_DATA_SOURCE	Test	Delete Duplicate

Cluster DS

DataSource ID	Export Name	Add
該当する内容が存在しません。		

- WebAdminの左側のメニューから[Servers]をクリックし、照会されたサーバー・リストから目的のサーバーを選択すると、[Server]画面へ移動します。「Data Source」項目に追加したデータソースが表示されるので、適用するデータソースを選択します。

[図 8.17] データソースの追加 - データソースの選択



8.5. WebAdminを使用したサーバーの起動および終了

WebAdminを使用してMSを起動および終了するには、DASが実行されている状態である必要があります。ただし、サーバーにノード・マネージャが実行されているか、SSHが設定されている必要があります。詳細内容については、『JEUS ノードマネージャガイド』を参照してください。

参考

Windowsの場合はSSHが動作しないため、ノード・マネージャを実行した後にMSを起動します。

以下は、WebAdminでMSを起動する手順についての説明です。

- WebAdminの左側のメニューから[Servers]を選択すると、サーバー・リストの照会画面が表示されます。
照会されたサーバー・リストから目的のサーバーを選択し、[Resource] > [Listener]メニューを選択します。[Listeners]画面で[Add]ボタンをクリックします。
- 照会されたサーバー・リストから目的のサーバーの[start]、[stop]ボタンを使って当該サーバーを起動、終了することができます。

[図 8.18] JEUS WebAdmin - MSの制御画面



第9章 Web層の使用

本章では、例を使用してサーブレット、JSP、JSTL、JSFアプリケーションのデプロイとWAR(Web Application ARchive)モジュールのパッケージングおよびデプロイについて説明します。

9.1. 例

本節では、Webアプリケーションの簡単なサンプル・コードを作成し、当該ソースのコンパイルとデプロイ手順について説明します。

参考

詳しい内容は、『JEUS サーバガイド』、『JEUS Webエンジンガイド』、『JEUS Webサービスガイド』を参照してください。

以下のHelloWorldServlet.javaは、Webブラウザで「Hello World!」というメッセージを出力するサンプル・サーブレットです。

[例 9.1] <<HelloWorldServlet.java>>

```
import java.io.*;

import javax.servlet.*;
import javax.servlet.http.*;

public class HelloWorldServlet extends HttpServlet
{
    public void doGet(HttpServletRequest req, HttpServletResponse res)
        throws IOException, ServletException
    {
        res.setContentType("text/html");

        PrintWriter out = res.getWriter();
        out.println("<HTML>");
        out.println("<HEAD>");
        out.println("<TITLE>Hello World Sample</TITLE>");
        out.println("</HEAD>");
        out.println("<BODY>");
        out.println("<CENTER><H1>Hello World!</H1></CENTER>");
        out.println("</BODY>");
    }
}
```

```
        out.println("</HTML>");
        out.close();
    }
}
```

例は以下のディレクトリーに格納されています。

```
JEUS_HOME/samples/getting_started/helloservlet/src/java
```

以下は、要求を受けたとき、リクエストに関するいくつかの情報を表示する「snoop.jsp」というサンプルJSPプログラムです。

[例 9.2] <<snoop.jsp>>

```
<html>

<body bgcolor="white">

<h2> Request Information </h2>
<font size="4">
JSP Request Method: <%= request.getMethod() %>
<br>
Request URI: <%= request.getRequestURI() %>
<br>
Request Protocol: <%= request.getProtocol() %>
<br>
Servlet path: <%= request.getServletPath() %>
<br>
Path info: <%= request.getPathInfo() %>
<br>
Path translated: <%= request.getPathTranslated() %>
<br>
Query string: <%= request.getQueryString() %>
<br>
Content length: <%= request.getContentLength() %>
<br>
Content type: <%= request.getContentType() %>
<br>
Server name: <%= request.getServerName() %>
<br>
Server port: <%= request.getServerPort() %>
<br>
Remote user: <%= request.getRemoteUser() %>
<br>
Remote address: <%= request.getRemoteAddr() %>
<br>
Remote host: <%= request.getRemoteHost() %>
```

```

<br>
Authorization scheme: <%= request.getAuthType() %>
<hr>
The browser you are using is <%= request.getHeader("User-Agent") %>
<hr>
</font>
</body>
</html>

```

例は以下のディレクトリーに格納されています。

```
JEUS_HOME/samples/getting_started/helloservlet/web
```

以下はsnoop.jspと同様な仕事をする、JSTLとJSFを使用した「snoop-jstl.jsp」というサンプルJSPプログラムです。

[例 9.3] <<snoop-jstl.jsp>>

```

<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<html>
<body>
<h2> Request Information </h2>
<font size="4">
    <c:set var="req" value="${pageContext.request}"/>
    JSP Request Method: <c:out value="${req.method}"/>
    <br/>
    Request Protocol: <c:out value="${req.protocol}"/>
    <br/>
    Servlet path: <c:out value="${req.servletPath}"/>
    <br/>
    Path info: <c:out value="${req.pathInfo}"/>
    <br/>
    Path translated: <c:out value="${req.pathTranslated}"/>
    <br/>
    Query string: <c:out value="${req.queryString}"/>
    <br/>
    Content length: <c:out value="${req.contentLength}"/>
    <br/>
    Content type: <c:out value="${req.contentType}"/>
    <br/>
    Server name: <c:out value="${req.serverName}"/>
    <br/>
    Server port: <c:out value="${req.serverPort}"/>
    <br/>
    Remote user: <c:out value="${req.remoteUser}"/>
    <br/>

```

```
Remote address: <c:out value="\${req.remoteAddr}"/>
<br/>
Remote host: <c:out value="\${req.remoteHost}"/>
<br/>
Authorization scheme: <c:out value="\${req.authType}"/>
<hr/>
<f:view>
The browser you are using is <h:outputText value=
"\#{header['User-Agent']}"/>
</f:view>
<hr/>
</font>
</body>
</html>
```

上記の例はJSPなのでユーザーがコンパイルする必要がありません。サーブレット・エンジンが自動でコンパイルします。

9.2. コンパイル

作成された例はjantを使用して以下のとおりビルドできます。

```
%JEOS_HOME%/samples/getting_started/helloservlet>jant build
```

正常にビルドされると、distフォルダーの下位に「hello-servlet.war」アプリケーションWARファイルが生成されます。

9.3. デプロイ

パッケージングされたWARモジュールは、WebAdminまたはコンソールを使用してデプロイします。

WebAdminを使用したWARモジュールのデプロイ

WebAdminを使用してアプリケーションをデプロイ(インストールおよびデプロイ)する手順について説明します。

参考

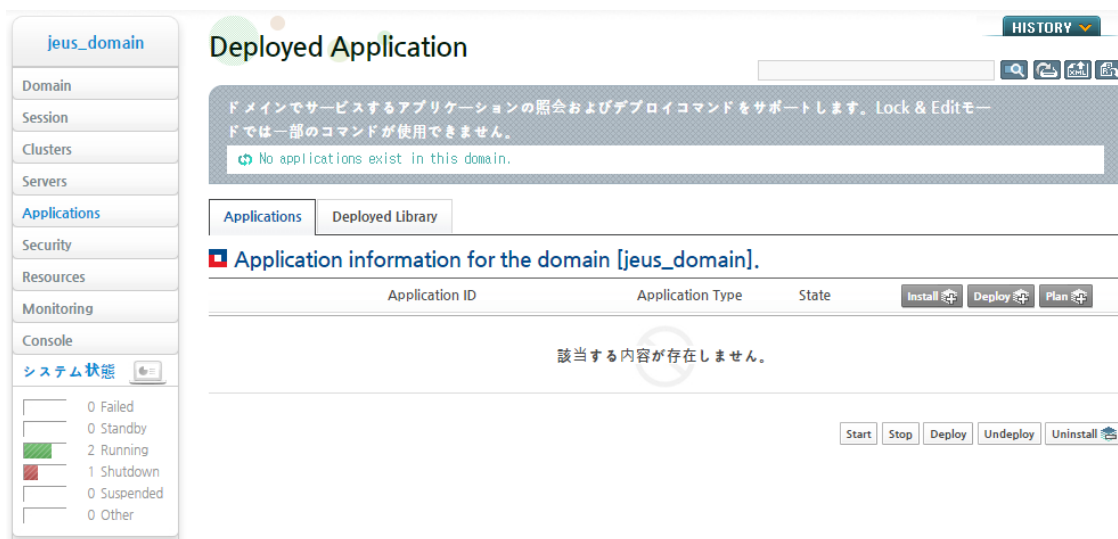
デプロイについての詳細内容は、『JEUS アプリケーション&デプロイメントガイド』を参照してください。

本節では、以前使用した方法と同様にWARモジュールをデプロイする方法について説明します。

1. Webブラウザに接続してWebAdminにログインします。

2. WebAdminの左側のメニューから[Applications]を選択すると、アプリケーション・リストの照会画面が表示されます。以下のように[Deployed Application]画面で[install]ボタンをクリックします。

[図 9.1] DASにインストールされているアプリケーション・リスト



3. [Install]画面の「Id」項目に使用するID値を入力します。「Path」項目の[ファイル選択]ボタンをクリックして、生成されたアプリケーションを選択した後、[確認]ボタンをクリックします。生成されたアプリケーションは、「JEUS_HOME/samples/getting_started/helloservlet/dis/」に格納されています。

[図 9.2] Webモジュールのインストール



4. アプリケーション・リストの照会画面が表示され、DASにインストールされたアプリケーションが表示されます。MSIにデプロイするためにアプリケーションの[**deploy**]ボタンをクリックします。

【図 9.3】 Webアプリケーションのインストールの成功

The screenshot shows the 'Deployed Application' page for the 'jeus_domain'. The left sidebar contains navigation links: Domain, Session, Clusters, Servers, Applications (selected), Security, Resources, Monitoring, and Console. Below these is a 'システム状態' (System Status) section with a bar chart and counts: 0 Failed, 0 Standby, 2 Running, 1 Shutdown, 0 Suspended, and 0 Other. The main content area has a 'Deployed Application' title and a 'HISTORY' dropdown. A message box states: 'ドメインでサービスするアプリケーションの照会およびデプロイコマンドをサポートします。Lock & Editモードでは一部のコマンドが使用できません。' Below this, a green message box says: 'helloworld : Successfully installed the application [/home/michael/jeus8000_install13/domains/jeus_domain/servers/adminServer/.workspace/deployed/webadmin/1472639711716/webadmin_war____/tempdir/helloworld.war]'. The 'Applications' tab is selected, showing 'Application information for the domain [jeus_domain]'. A table lists the application 'helloworld' with Application ID, Application Type, and State (INSTALLED). Action buttons (Install, Deploy, Plan) are available for each application. At the bottom, there are buttons for Start, Stop, Deploy, Undeploy, and Uninstall.

5. 「**Server**」項目からデプロイするサーバーを選択し、[**確認**]ボタンをクリックします。DASを選択することも可能ですが、MSIにデプロイしてアプリケーションをサービスすることをお勧めします。

6. 正常にデプロイされると、以下のように結果メッセージが出力されます。

【図 9.4】 Webモジュールのデプロイ

The screenshot shows the 'Deployed Application' page for the 'jeus_domain'. The left sidebar is the same as in Figure 9.3. The main content area shows a green message box: 'deploy the application for the application [helloworld] succeeded.'. The 'Applications' tab is selected, showing 'Application information for the domain [jeus_domain]'. A table lists the application 'helloworld' with Application ID, Application Type (WAR), and State (RUNNING). Action buttons (Redeploy, Add Target, Remove Target) are available for each application. At the bottom, there are buttons for Start, Stop, Deploy, Undeploy, and Uninstall.

コンソール・ツールを使用したWARモジュールのデプロイ

コンソール・ツール(jeusadmin)を使用してWebモジュールをデプロイする方法は以下のとおりです。

1. jeusadminを使用してJEUSに接続します。

```
jeusadmin -u administrator -p <password>
```

2. アプリケーションをDASにインストールします。

```
[DAS]jeus_domain.adminServer>install-application -id helloworld  
C:\TmaxSoft\JEUS8.0\samples\getting_started\helloservlet\dist\hello-servlet.war  
Successfully installed application[helloworld].
```

3. アプリケーションをMS(server1)にデプロイします。

```
[DAS]jeus_domain.adminServer>deploy helloworld -servers server1  
Succeeded to deploy the application : helloworld
```

4. モジュールが正常にデプロイされたのかを確認します。

9.4. 実行および結果

本節では、デプロイされたJSPとサーブレットの使用方法について説明します。

デプロイされたJSPの使用方法

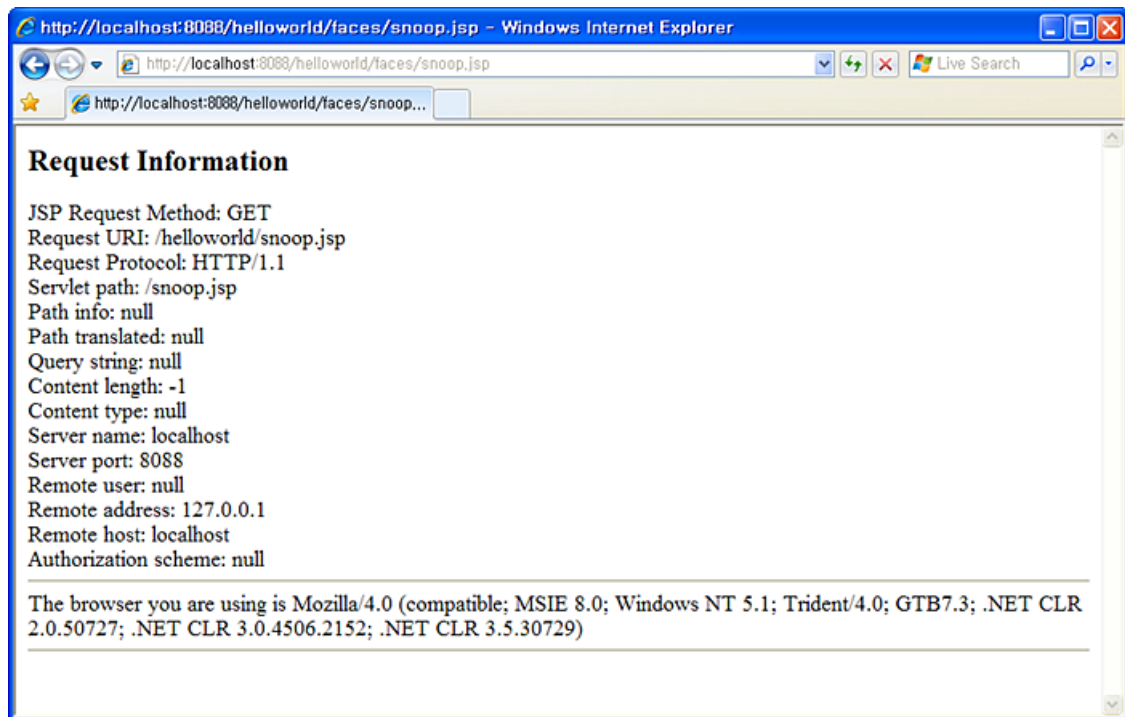
デプロイされたJSPの使用方法は以下のとおりです。

- snoop.jspページの呼び出し

以下のアドレスをWebブラウザのアドレスバーに入力してsnoop.jspページを呼び出します。(JSPの場合、最初の呼び出しでサーブレット・エンジンが自動でコンパイルを行うので、実行が若干遅れます)

```
http://localhost:8088/helloworld/faces/snoop.jsp
```

[図 9.5] WARモジュールのJSP呼び出し



- snoop-jstl.jspページの呼び出し

以下のアドレスを入力してsnoop-jstl.jspページを呼び出します。画面結果はsnoop.jspの呼び出し時と同様です。

```
http://localhost:8088/helloworld/snoop-jstl.jspf
```

デプロイされたサーブレットの使用方法

ブラウザーに以下のアドレスを入力して「helloworld」サーブレットを呼び出します。

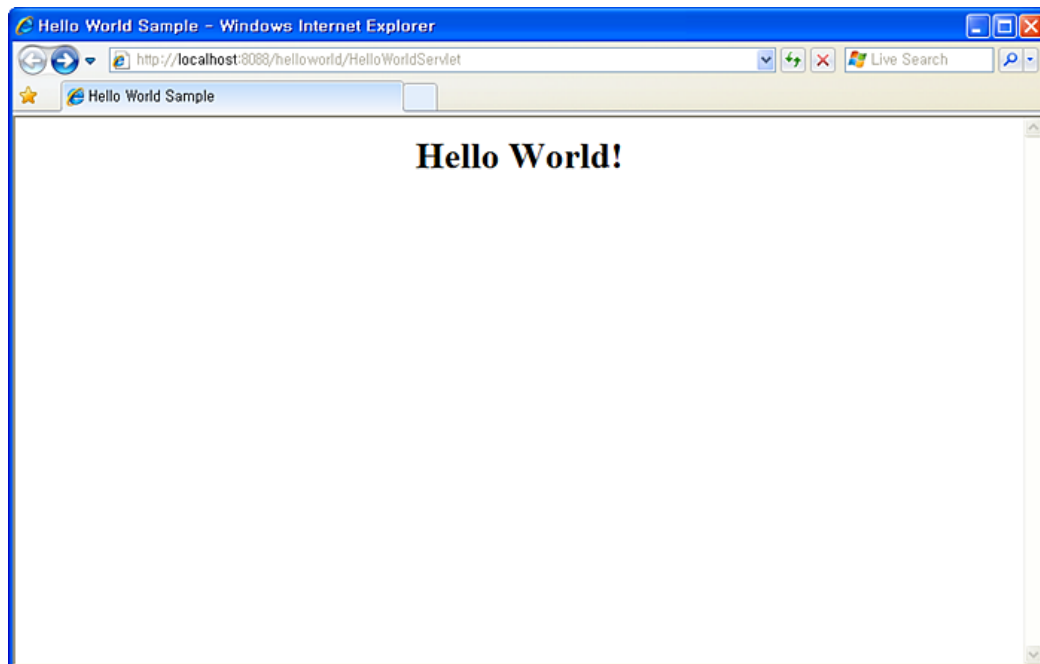
```
http://localhost:8088/helloworld/HelloWorldServlet
```

項目	説明
http	JEUSに接続する際に使用するHTTPプロトコルです
localhost	サービスを提供するサーバーがブラウザーと同様な自身のアドレスに存在することを意味します
8088	MS内に生成したHTTPリスナーのポート番号です
helloworld	Webアプリケーションのコンテキスト用のリクエスト・パスです。このリクエスト・パスは、jeus-web-dd.xmlの<context-path> elementに設定されます。設定しない場合、WARモジュール・ファイルの名前と同様です

項目	説明
HelloWorldServlet	サーブレットに定義されたURLパターンです

サーブレット・エンジンが正常に起動されており、「Hello World」サーブレットが正常にデプロイされると、以下のような画面が表示されます。

[図 9.6] WARモジュール・サーブレットの呼び出し



第10章 EJBの使用

本章では、ステートレス・セッションBeanとJava Persistence APIを使ってエンティティを開発する方法について例を使用して説明します。

10.1. セッションBeanの例

セッションBeanは、基本的にビジネス・インターフェースとBeanクラスで構成されます。

10.1.1. 例

EJBの例

以下のサンプル・コードは「helloejb」を出力するビジネス・メソッドを持つ簡単なステートレス・セッションBeanの例です。

- ビジネス・インターフェース

[例 10.1] <<Hello.java>>

```
package helloejb;

import javax.ejb.Remote;

@Remote
public interface Hello {
    String sayHello();
}
```

- Beanクラス

[例 10.2] <<HelloBean.java>>

```
package helloejb;

import javax.ejb.Stateless;

@Stateless(mappedName="helloejb.Hello")
public class HelloBean implements Hello {

    public String sayHello() {
```

```

        return "Hello EJB!";
    }

}

```

サンプル・コードは以下のディレクトリーに格納されています。

```
JEUS_HOME/samples/getting_started/helloejb/helloejb-ejb/src/java/helloejb
```

サーブレット・クライアントの例

helloejbを呼び出すサーブレット・クライアントを以下のように実装します。

[例 10.3] <<HelloClient.java>>

```

package helloejb;

import java.io.*;
import javax.ejb.EJB;

import javax.servlet.*;
import javax.servlet.http.*;

public class HelloClient extends HttpServlet {
    @EJB(mappedName="helloejb.Hello")
    private Hello hello;

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        PrintWriter out = response.getWriter();
        try {
            // Call session bean business method.
            String msg = hello.sayHello();

            response.setContentType("text/html");
            out.println("<html>");
            out.println("<head>");
            out.println("<title>HelloClient</title>");
            out.println("</head>");
            out.println("<body>");
            out.println("<center><h1>" + msg + "</h1></center>");
            out.println("</body>");
            out.println("</html>");
            out.close();
        } catch (Exception ex) {

```

```

        response.setContentType("text/plain");
        ex.printStackTrace(out);
    }
}

protected void doGet(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}

protected void doPost(HttpServletRequest request,
    HttpServletResponse response)
    throws ServletException, IOException {
    processRequest(request, response);
}
}

```

サンプル・コードは以下のディレクトリーに格納されています。

```
%JEUS_HOME%/samples/getting_started/helloejb/helloejb-war/src/java/helloejb
```

10.1.2. コンパイル

作成された例は、Antを使用して以下のとおりビルドできます。

```
%JEUS_HOME%/samples/getting_started/helloejb>jant build
```

正常にビルドされると、%JEUS_HOME%/samples/getting_started/helloejb/dist/helloejb.earアプリケーションEARファイルが生成されます。

10.1.3. デプロイ

パッケージングしたEJBアプリケーションをWebAdminまたはコンソール・ツールを使用してデプロイできます。

WebAdminを使用したデプロイ

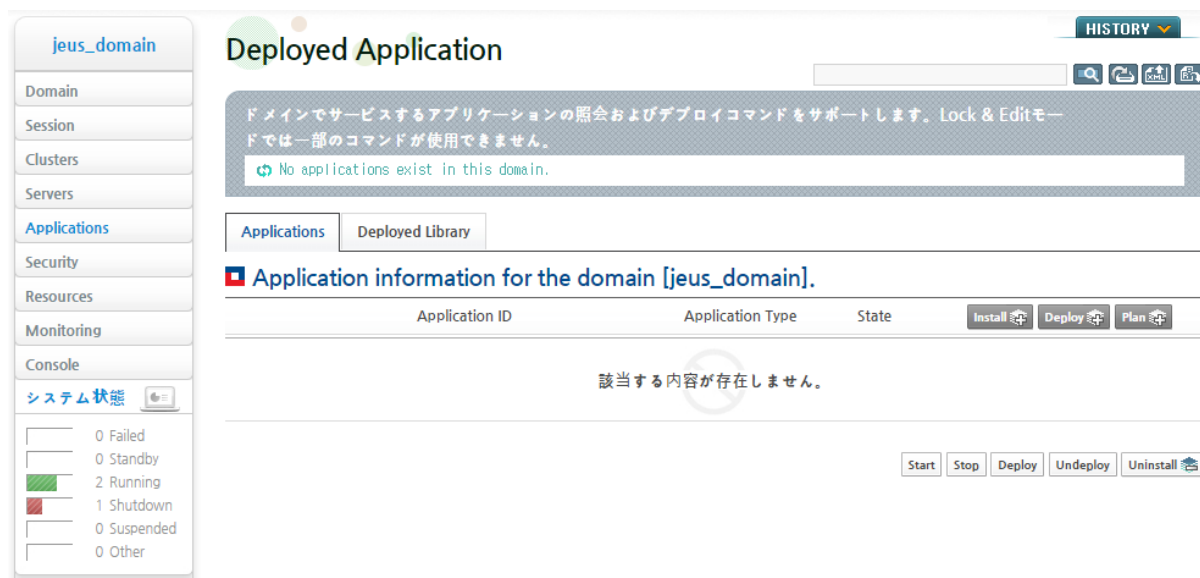
WebAdminを使用してデプロイする手順は以下のとおりです。

1. Webブラウザに接続してWebAdminにログインします。

2. WebAdminの左側のメニューから[Applications]を選択すると、アプリケーション・リストの照会画面が表示されます。

[Deployed Applications]画面で[Install]ボタンをクリックします。

【図 10.1】 DASにインストールされたアプリケーション・リスト



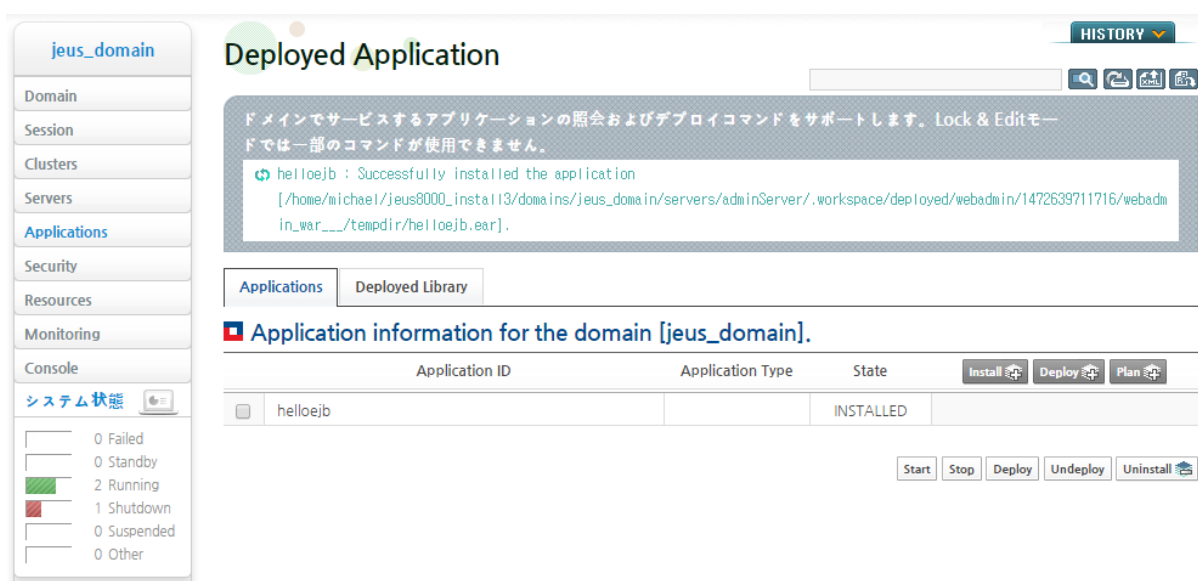
3. [Install]画面の「Id」項目に使用するID値を入力します。「Path」項目の[ファイル選択]ボタンをクリックして、生成されたアプリケーションを選択した後、[確認]ボタンをクリックします。

【図 10.2】 HelloEJBのデプロイ



4. アプリケーション・リストの照会画面が表示され、DASにインストールされたアプリケーションが照会されます。MSIにデプロイするためにアプリケーションの[deploy]ボタンをクリックします。

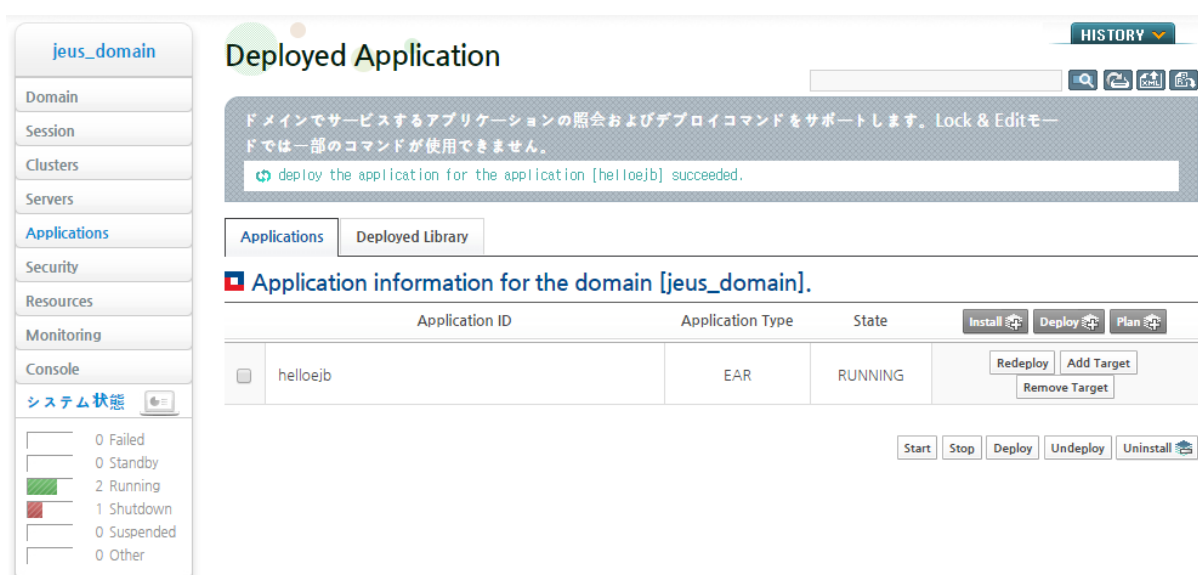
[図 10.3] HelloEJBのデプロイ選択事項



5. 「Server」項目からデプロイするサーバーをチェックし、[確認]ボタンをクリックします。DASを選択することも可能ですが、MSIにデプロイしてアプリケーションをサービスすることをお勧めします。

6. 正常にデプロイされると、以下のように結果メッセージが出力されます。

[図 10.4] HelloEJBのデプロイ結果



コンソール・ツールを使用したデプロイ

コンソール・ツールを使用してデプロイする手順は以下のとおりです。

1. 上記で作成したhelloejb.earファイルを任意のパスに格納します。

2. jeusadminを使用してJEUSに接続します。

```
jeusadmin johan -u administrator -p password>
```

3. アプリケーションをDASにインストールします。

```
[DAS]jeus_domain.adminServer>install-application -id helloejb  
C:\TmaxSoft\JEUS8.0\samples\getting_started\helloejb\dist\helloejb.ear  
Successfully installed application[helloejb].
```

4. アプリケーションをMS(server1)にデプロイします。

```
[DAS]jeus_domain.adminServer>deploy helloejb -servers server1  
Succeeded to deploy the application : helloejb
```

10.1.4. 実行および結果

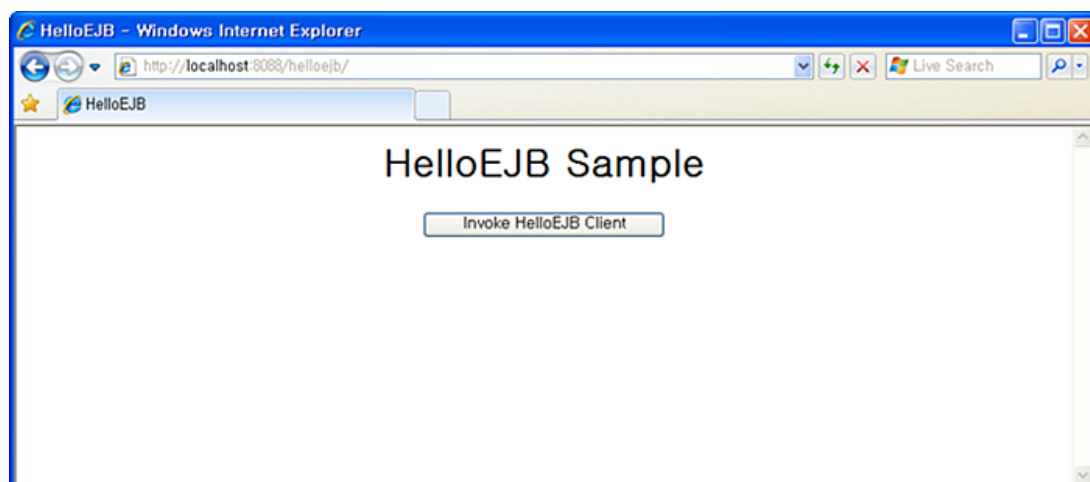
デプロイが完了したら、作成された画面を実行して結果を確認します。

以下は、Webクライアントを実行する手順です。

1. Webブラウザを利用して以下のURLに接続すると、HelloEJBのWebクライアントに接続できます。

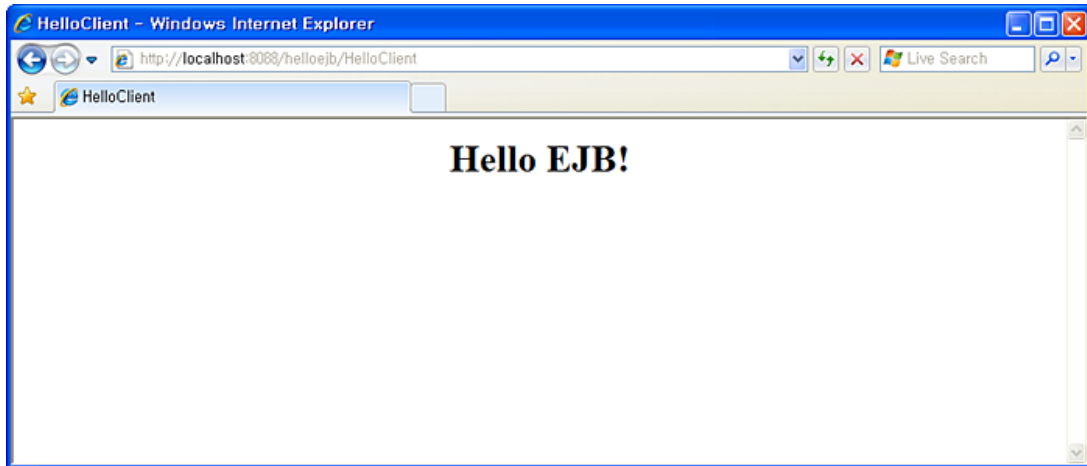
```
http://localhost:8088/helloejb/
```

[図 10.5] HelloEJBのクライアント画面



2. 画面の[Invoke HelloEJB Client]ボタンをクリックすると、EJBを呼び出すHelloClientサーブレットを実行させ、以下のような結果画面が出力されます。

[図 10.6] HelloEJBのサーブレット・クライアントの実行結果



10.2. Java Persistence APIの例

本節では、Java Persistence APIを使用してエンティティを作成、コンパイル、デプロイする手順について説明します。

10.2.1. 例

EJBの例

例は、Product Entityと、これを使用するEJBであるProductManager Session Beanで構成されています。

- エンティティ

[例 10.4] <<Product.java>>

```
package hellojpa;

import java.io.Serializable;
import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.NamedQuery;

@Entity
@NamedQuery(name="findAllProducts", query="SELECT p FROM Product p")
public class Product implements Serializable {
    @Id
    private String productId;
```

```

private double price;
private String description;

public Product() {
}

public Product(String productId, double price,
    String description){
    this.productId = productId;
    this.price = price;
    this.description = description;
}

public String getProductId() {
    return productId;
}

public void setProductId(String id) {
    this.productId = id;
}

public double getPrice() {
    return price;
}

public void setPrice(double price) {
    this.price = price;
}

public String getDescription() {
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

public String toString() {
    return "Product[productId=" + productId + ", price=" +
        price + ", description=" + description + "]";
}
}

```

- ビジネス・インターフェース

[例 10.5] <<ProductManager.java>>

```
package hellojpa;

import java.util.Collection;
import javax.ejb.Local;

@Local
public interface ProductManager {
    Product createProduct(String productId, double price, String desc);

    Product getProduct(String productId);

    Collection findAllProducts();

    Collection findProductsByDescription(String desc);

    Collection findProductsInRange(double low, double high);

    void updateProduct(Product product);

    void removeProduct(Product product);

    void removeAllProducts();
}
```

- Beanクラス

[例 10.6] <<ProductManagerBean.java>>

```
package hellojpa;

import java.util.Collection;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;

@Stateless(mappedName="hellojpa.ProductManager")
public class ProductManagerBean implements ProductManager {
    @PersistenceContext
    private EntityManager em;

    public ProductManagerBean() {
    }
}
```

```

public Product createProduct(String productId, double price, String desc){
    Product product = new Product(productId, price, desc);
    em.persist(product);
    return product;
}

public Product getProduct(String productId){
    return (Product)em.find(Product.class, productId);
}

public Collection findAllProducts() {
    return em.createNamedQuery("findAllProducts").getResultList();
}

public Collection findProductsByDescription(String desc){
    Query query = em.createQuery("SELECT p FROM Product p WHERE
        p.description=:desc");
    query.setParameter("desc", desc);
    return query.getResultList();
}

public Collection findProductsInRange(double low, double high){
    Query query = em.createQuery("SELECT p FROM Product p WHERE
        p.price between :low and :high");
    query.setParameter("low", low).setParameter("high", high);
    return query.getResultList();
}

public void updateProduct(Product product){
    Product managed = em.merge(product);
    em.flush();
}

public void removeProduct(Product product){
    Product managed = em.merge(product);
    em.remove(managed);
}

public void removeAllProducts(){
    em.createQuery("DELETE FROM Product p").executeUpdate();
}
}

```

サンプル・コードは以下のディレクトリーに格納されています。

```
JEUS_HOME/samples/getting_started/hellojpa/hellojpa-ejb/src/java/hellojpa
```

サーブレット・クライアントの例

ProductManager EJBを使用してデータベースにデータを保存し、データを使用するサーブレット・クライアントを以下のとおり実装します。

[例 10.7] <<ProductManagerClient.java>>

```
package hellojpa;

import java.io.*;
import java.util.Collection;
import javax.ejb.EJB;

import javax.servlet.*;
import javax.servlet.http.*;

public class ProductManagerClient extends HttpServlet {
    @EJB
    private ProductManager productManager;

    protected void processRequest(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        response.setContentType("text/plain");
        PrintWriter out = response.getWriter();
        out.println("SERVLET CLIENT CONSOLE OUTPUT:\n");

        productManager.removeAllProducts();
        out.println("Cleaned up existing products.\n");

        out.println("Creating products...");
        Product p1 = productManager.createProduct("1", 10.00, "Ceramic Dog");
        Product p2 = productManager.createProduct("2", 13.00, "Wooden Duck");
        Product p3 = productManager.createProduct("3", 19.00, "Ivory Cat");
        Product p4 = productManager.createProduct("4", 33.00, "Ivory Cat");
        Product p5 = productManager.createProduct("5", 22.00, "Chrome Fish");

        Collection products;

        out.println("Created products:");
        products = productManager.findAllProducts();
        for(Object product : products){
            out.println(product);
        }
        out.println();
    }
}
```

```

        out.println("Find product with productId 1:");
        Product pp1 = productManager.getProduct("1");
        out.println("Found = " + pp1.getDescription() + " $" + pp1.getPrice());

        out.println("Update the price of this product to 12.00");
        pp1.setPrice(12.00);
        productManager.updateProduct(pp1);

        Product pp2 = productManager.getProduct("1");
        out.println("Product " + pp2.getDescription() + " is now $" +
pp2.getPrice());
        out.println();

        out.println("Find products with description:");
        products = productManager.findProductsByDescription("Ivory Cat");
        for(Object product : products){
            out.println(product);
        }
        out.println();

        out.println("Find products with price range between 10.00 and 20.00");
        products = productManager.findProductsInRange(10.00, 20.00);
        for(Object product : products){
            out.println(product);
        }
        out.println();

        out.println("Removed all products.");
        productManager.removeProduct(p1);
        productManager.removeProduct(p2);
        productManager.removeProduct(p3);
        productManager.removeProduct(p4);
        productManager.removeProduct(p5);

        out.close();
    }

    protected void doGet(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

    protected void doPost(HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException {
        processRequest(request, response);
    }

```



```
}  
}
```

サンプル・コードは以下のディレクトリーに格納されています。

```
JEUS_HOME/samples/getting_started/hellojpa/hellojpa-war/src/java/hellojpa
```

10.2.2. コンパイル

jantを使用してサンプル・コードを以下のとおりビルドできます。

1. 例が作成されたパスで**ant build**コマンドを実行します。

```
C:\TmaxSoft\JEUS8.0\samples\getting_started\hellojpa>jant build
```

2. 正常にビルドされると、dist/hellojpa.earアプリケーションEARファイルが生成されます。

同例はデータベースが必要なのでDerbyを開始します。また、データベースがjdbc/sampleデータソースに設定されている必要があります。詳細内容は、「[第8章 システムの設定](#)」を参照してください。

Derbyが開始されたら、以下のようにデータベース・テーブルを生成します。本例では「sample」というデータベースを使用します。

```
CREATE TABLE PRODUCT (PRODUCTID VARCHAR(255) NOT NULL, PRICE FLOAT,  
DESCRIPTION VARCHAR(255), PRIMARY KEY (PRODUCTID));
```

3. **ant setup**コマンドを実行してデータベース・テーブルを生成します。

```
C:\TmaxSoft\JEUS8.0\samples\getting_started\hellojpa>jant setup
```

以上で(ビルドとデータベース設定の完了)パッケージングされたモジュールをデプロイする準備が整いました。

10.2.3. デプロイ

パッケージングしたEJBアプリケーションをWebAdminまたはコンソール・ツールを使用してデプロイできます。

WebAdminを使用したデプロイ

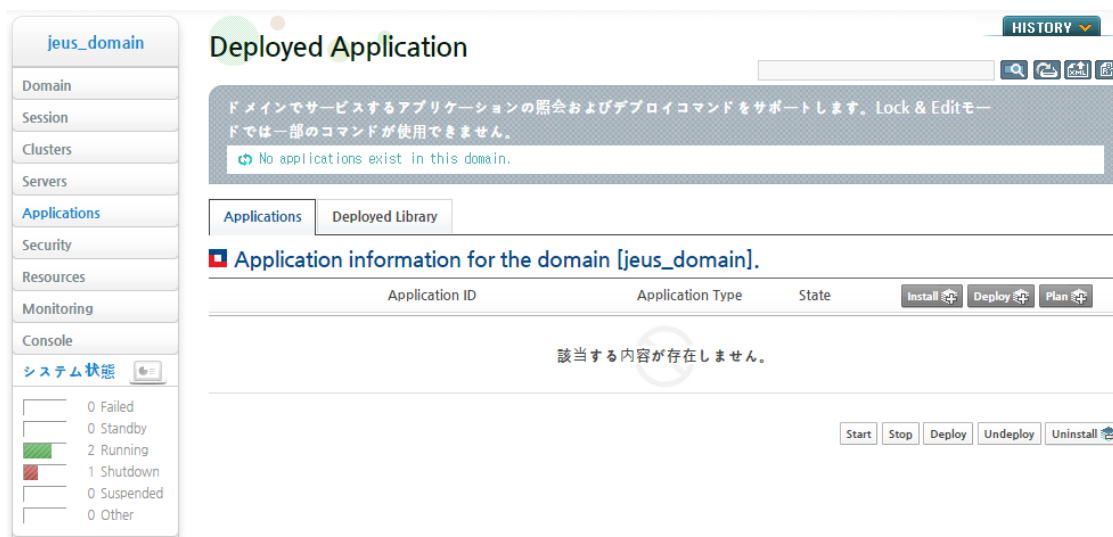
WebAdminを使用してパッケージングされたモジュールをデプロイします。

1. Webブラウザに接続してWebAdminにログインします。

2. WebAdminの左側のメニューから[Applications]を選択すると、アプリケーション・リストの照会画面が表示されます。

[Deployed Application]画面で[install]ボタンをクリックします。

[図 10.7] DASにインストールされたアプリケーション・リスト



3. [Install]画面の「Id」項目に使用するID値を入力します。「Path」項目の[ファイル選択]ボタンをクリックして、生成されたアプリケーションを選択した後、[確認]ボタンをクリックします。

生成されたアプリケーションは、「JEUS_HOME/samples/getting_started/hellojpa/dist」に格納されています。

[図 10.8] インストール対象の選択



4. アプリケーション・リストの照会画面に移動され、DASにインストールされたアプリケーションが照会されます。MSにデプロイするためにアプリケーションの[deploy]ボタンをクリックします。

[図 10.9] Webアプリケーションのインストールの成功

The screenshot shows the JBoss Web Console interface. On the left is a sidebar with a navigation menu including 'Domain', 'Session', 'Clusters', 'Servers', 'Applications' (selected), 'Security', 'Resources', 'Monitoring', and 'Console'. Below the menu is a 'システム状態' (System Status) section showing various status indicators. The main content area is titled 'Deployed Application' and contains a message box stating: 'hellojpa : Successfully installed the application'. Below this, there is a table titled 'Application information for the domain [jeus_domain]'. The table has columns for 'Application ID', 'Application Type', and 'State'. It lists one application, 'hellojpa', with a state of 'INSTALLED'. To the right of the table are buttons for 'Start', 'Stop', 'Deploy', 'Undeploy', and 'Uninstall'.

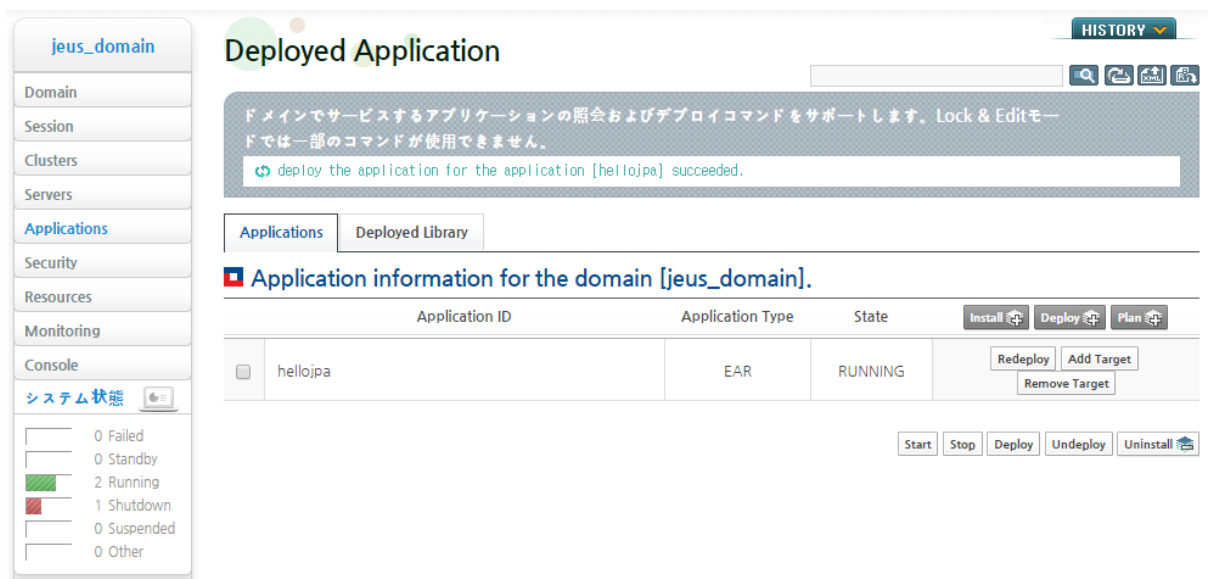
5. 「Server」項目からデプロイするサーバーをチェックし、[確認]ボタンをクリックします。DASを選択することも可能ですが、MSにデプロイしてアプリケーションをサービスすることをお勧めします。

[図 10.10] HelloJPAのデプロイ・オプション

The screenshot shows the 'Deploy' dialog box. It has a title bar with a close button. Below the title bar is a section for '動的設定' (Dynamic Configuration) and '必須項目' (Required Items). The main section is titled 'Targets' and contains four rows: 'Target All Servers' with a checkbox, 'Target Server' with a list of servers and checkboxes, 'Target Cluster' with a message, and 'Virtual Host' with a dropdown menu. The 'server1' checkbox is checked.

6. 正常にデプロイされると、以下のように結果メッセージが出力されます。

[図 10.11] HelloJPAのデプロイ結果



参考

「[8.4. データソースの追加](#)」のデータソース設定の部分で説明しましたが、データソース設定において、パスワードやIPアドレスが正しくない場合はデプロイに失敗します。

コンソール・ツールを使用したデプロイ

コンソール・ツールを使用してデプロイする手順は以下のとおりです。

1. jeusadminを使用してJEUSに接続します。

```
jeusadmin -u administrator -p <password>
```

2. アプリケーションをDASにインストールします。

```
[DAS]jeus_domain.adminServer>install-application -id hellojpa
C:\TmaxSoft\JEUS8.0\samples\getting_started\hellojpa\dist\hellojpa.ear
Successfully installed application[hellojpa].
```

3. アプリケーションをMS(server1)にデプロイします。

```
[DAS]jeus_domain.adminServer>deploy hellojpa -servers server1
Succeeded to deploy the application : hellojpa
```

4. モジュールが正常にデプロイされたのを確認します。

10.2.4. 実行および結果

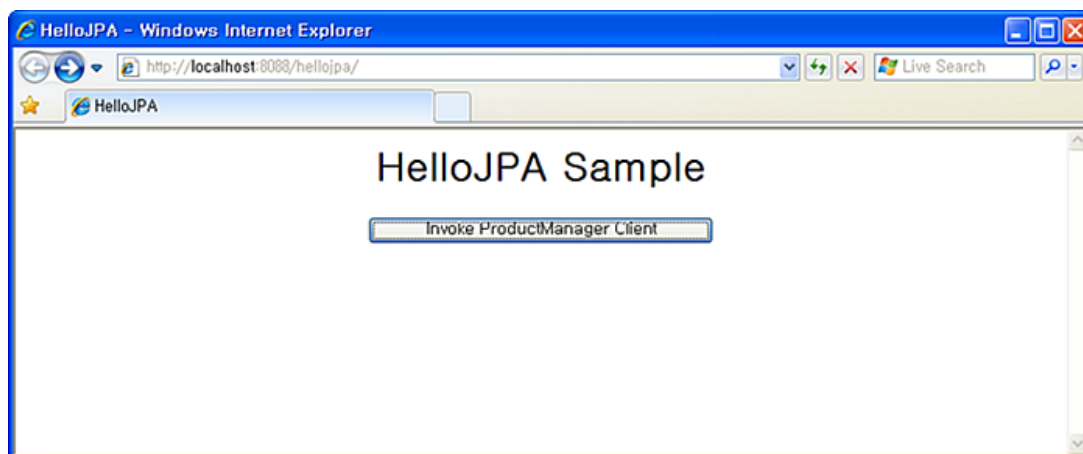
デプロイが完了したら、作成された画面を実行して結果を確認します。

以下は、Webクライアントを実行する手順です。

1. Webブラウザを利用して以下のURLに接続すると、HelloJPAのWebクライアントに接続できます。

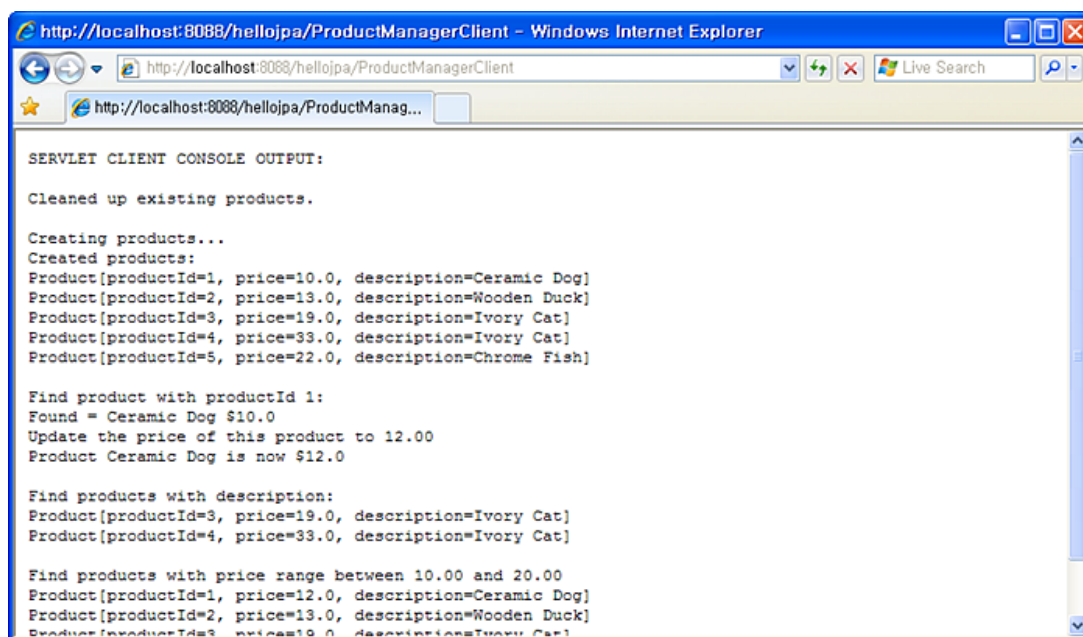
```
http://localhost:8088/hellojpa/
```

[図 10.12] HelloJPAのクライアント画面



2. 画面の[Invoke ProductManager Client]ボタンをクリックすると、EJBを呼び出すサーブレット・クライアントを実行させ、以下のような結果画面が出力されます。

[図 10.13] HelloJPAのサーブレット・クライアントの実行結果



第11章 Webサービスの使用

本章では、JAX-WS 2.2を利用したWebサービスの生成とクライアントの作成およびWebサービスを呼び出す方法について説明します。

11.1. Webサービスの生成

JEUS 8では、Java EE 7のWebサービスをサポートしており、JAX-WS 2.2はJava EE 6 Webサービスの中核です。JAX-WSは既存のJAX-RPCを代替する手法として設計されました。新しく登場したJAX-WSの背景にはJAXB 2.0があり、JAXB 2.2からはすべてのXMLスキーマ・タイプを完全サポートするため、既存のJavaタイプとXMLタイプ間のマッピングをより明確に定義することができます。これは、JAX-RPC仕様に存在していたJavaタイプとXMLタイプ間の依存性が解消できる原動力となりました。なお、SOAP 1.2メッセージを直接使用できるようにするSAAJ 1.3まで含め、Java Web Services 2.0という新しいWebサービスのモデルが登場されます。

11.1.1. From Java方式

From Java方式の基本的なWebサービスの生成手順は以下のとおりです。

1. Webサービス・アノテーションが含まれたサービスを実装するBeanの作成

以下は、サービス実装Beanを作成する際の制約です。

- `javax.jws.WebService` Annotationを含ませ、当該クラスがサービス実装Beanであることを明示します。
- Webサービス・メソッドのパラメータとリターン・タイプは、JAXB 2.0のJavaとXMLスキーマ間のマッピング定義と互換される必要があります。
- Webサービス・メソッドのパラメータとリターン・タイプは、`java.rmi.Remote`インターフェースを直接または間接的に実装してはいけません。

以外にも、`javax.jws`パッケージに定義されている様々なアノテーションを利用すると、メソッドのパラメータとリターン・タイプ、バインディング方式などのカスタマイジングが可能です。簡単なサンプル・コードを使用して実際のサービス実装Beanの作成方法について説明します。

以下は、簡単なWebサービスを実装したクラスの例です。サンプル・コードは以下のディレクトリに格納されています。

```
JEUS_HOME/samples/getting_started/webservices/from_java/src/java/fromjava/server
```

[例 11.1] <<AddNumbersImpl.java>>

```
package fromjava.server;

import javax.xml.ws.WebService;

@WebService
public class AddNumbersImpl {
    public int addNumbers(int number1, int number2) {
        return number1 + number2;
    }
}
```

JAX-WSでは、既存のJavaクラスのWebサービスへの公開が容易です。上記の<AddNumbersImpl.java>で確認できるように@WebService Annotationを追加することで、既存のJavaクラスを簡単にWebサービスに変換して作成することができます。

2. 他ベンダー間に移植可能なポータブル・アーティファクトの生成

サービス実装Beanを作成してコンパイルまで実行したら、JAX-WSランタイムで使用する他ベンダー間に移植可能なポータブル・アーティファクトを生成する必要があります。他ベンダー間に移植可能なポータブル・アーティファクトは、JAX-WS仕様に準拠するすべてのベンダーが使用できるJAX-WSツールを使って生成されたものを意味します。Javaのパラメータを、実際のWSDLのメッセージで正確にマッピングするための情報などが含まれているJavaクラスやWSDLなどがこれに該当します。

JEUSでは、**wsgen**というコンソール・スクリプトを提供しており、JEUS_HOME/binディレクトリーの下位に格納されています。

```
wsgen -cp <classpath> -d <destination_dir> fromjava.server.AddNumbersImpl
```

上記のコマンドを実行すると、指定したパスに移植可能なポータブル・アーティファクトが生成されたことが確認できます。このスクリプトを実行するときに-wsdlオプションを設定するとWSDLまで生成されますが、JAX-WSではWebサービスのエンドポイントにWSDLを含ませる必要がないので、-wsdlオプションを設定しないように注意します。以下は、ポータブル・アーティファクトをまとめてサーバーにデプロイする作業です。

3. Webサービスのパッケージングおよびデプロイ

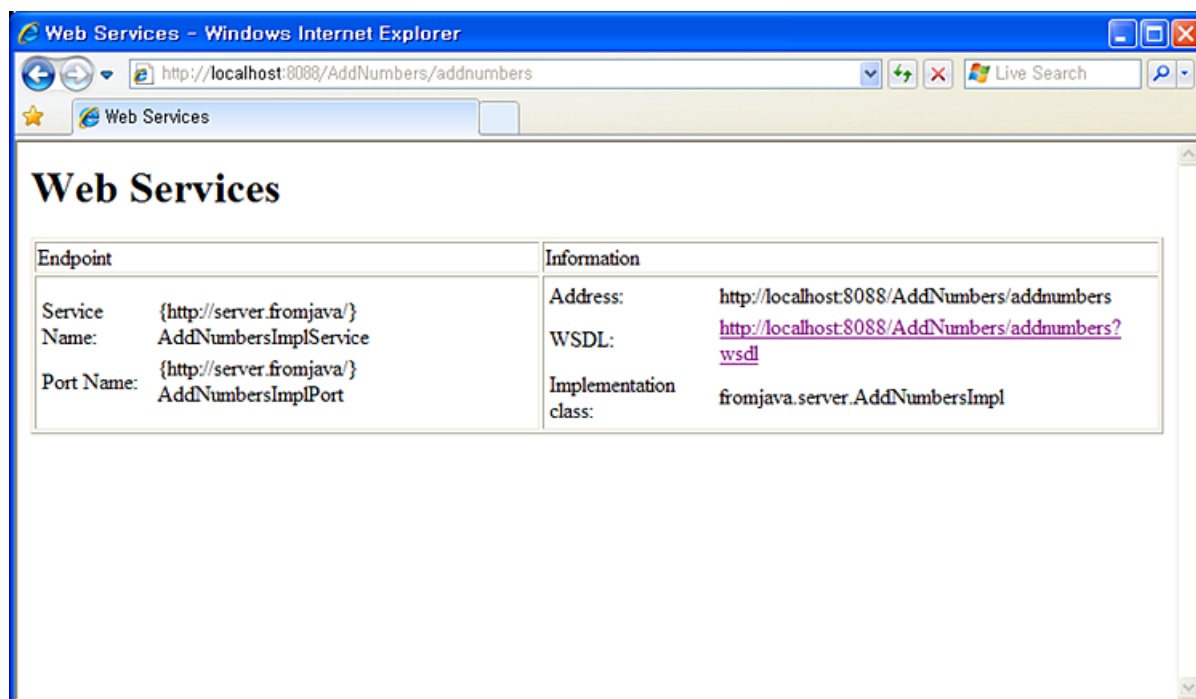
Webサービスのパッケージングとは、サービス実装Beanおよびサービス実装Beanが参照しているJavaクラスと付加的なDDをWAR形式にまとめることです。こちらでは、上記で作成したfromjava.server.AddNumbersImplクラスと、wsgenスクリプトを使用して生成したfromjava.server.jaxws.AddNumbers、fromjava.server.jaxws.AddNumbersResponseクラスのみ含まれます。これらのクラスをWEB-INF/classesの下位に格納します。パッケージングするファイル名をAddNumbers.warに指定し、WARパッケージングを実行してからJEUS 7にデプロイします。

実際にこのサービスが呼び出せるHTTPアドレスは以下のとおりです。


```
http://localhost:8088/AddNumbers/addnumbers
```

Webブラウザで上記のアドレスを呼び出すと、以下のとおりWebサービスが正常にデプロイされたことが確認できます。

【図 11.1】 正常にデプロイされたAddNumbers



サンプルを簡単に実行

JEUS 8では、Webサービスのサンプルが簡単に実行できる方法を提供しています。

JEUS_HOME/samples/getting_started/webservices/from_javaの下位でjantコマンドを実行すると、サービス・パッケージングからクライアントまで簡単に実行できます。

作成された例は、Antを使用してビルドできます。

```
%JEUS_HOME%/samples/getting_started/webservices/from_java> jant
```

正常にビルドされるとWebブラウザが開かれ、結果を確認できます。

11.1.2. From WSDL方式

From Java方式のWebサービスがすでに作成したJava RPCモデルをWebサービスに公開することならば、From WSDL方式のWebサービスは、互いに通信するSOAPメッセージを先に定義し、WSDLを介してその情報を共有した後、定義されたメッセージ・タイプに合わせてJavaクラスを生成することが目的です。

一般的なFrom WSDL方式のWebサービスの生成は、以下の手順で行われます。

1. サービス・エンドポイント・インターフェースの生成

すでに公開されているWSDLをベースにし、JEUS 8が提供するwsimportコンソール・スクリプトを使って、WebサービスのJavaインターフェース・ファイルとJavaクラス・ファイルを生成します。wsimportスクリプトはJEUS_HOME/binに格納されています。

JEUS_HOME/samples/getting_started/webservices/from_wsdlディレクトリーで以下のように実行します。

```
wsimport -keep -p fromwsdl.server -d ./build/classes
./web/WEB-INF/wsdl/AddNumbers.wsdl
```

上記のコマンドを実行すると、指定したパスにサービス・エンドポイント・インターフェースとサービス定義クラスが含まれた様々な結果が生成されます。

生成されたサービス・エンドポイント・インターフェースには、JAX-WSランタイムに使用される情報がアノテーション形式で含まれています。

[例 11.2] <<AddNumbersImpl.java>>

```
package fromwsdl.server;

@javax.jws.WebService(endpointInterface = "fromwsdl.server.AddNumbersPortType",
    wsdlLocation = "WEB-INF/wsdl/AddNumbers.wsdl",
    targetNamespace = "urn:AddNumbers", serviceName = "AddNumbersService",
    portName = "AddNumbersPort")
public class AddNumbersImpl {

    public int addNumbers(int number1, int number2) {
        return number1 + number2;
    }
}
```

2. サービス・エンドポイント・インターフェースの実装

サービス・エンドポイント・インターフェースが生成されたら、エンドポイント・インターフェースを実装する実際のビジネス・ロジックを持つサービス実装Beanを作成します。このサービス実装Beanを作成するときは、@javax.jws.WebService Annotationを追加する必要があり、このアノテーションはサービス・エンドポイント・インターフェースを明示したエンドポイント・インターフェース・メンバーを属性として有する必要があります。

3. Webサービスのパッケージングおよびデプロイ

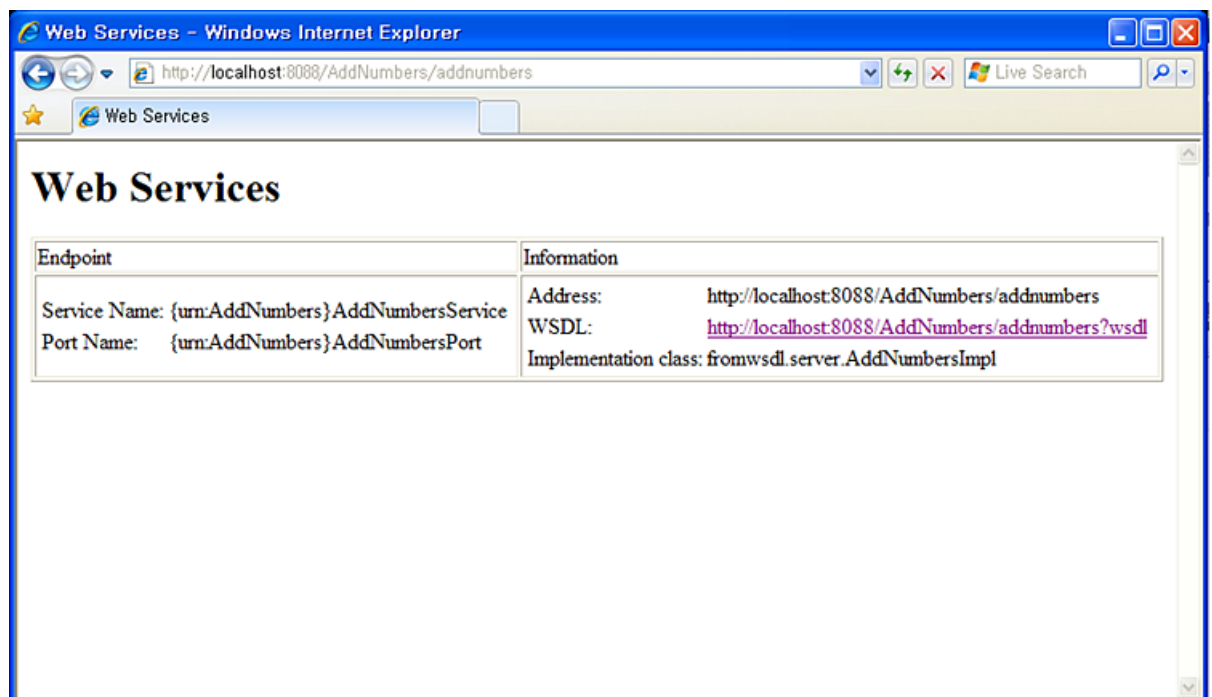
Webサービスに公開するため、WAR形式にパッケージングする作業は、From Java方式のWebサービス・パッケージングおよびデプロイ作業と類似しています。

wsimportスクリプトを使用して生成したサービス・エンドポイント・インターフェースを含むいくつかの結果とサービス実装BeanをWEB-INF/classesに格納し、WAR形式にパッケージングします。WARパッケージ名をAddNumbers.warに指定し、JEUS 8にデプロイした場合、以下のアドレスでアクセスできます。すでにFrom Java方式でデプロイした場合はコンテキスト名が重複されるので、削除してから再デプロイします。

```
http://localhost:8088/AddNumbers/addnumbers
```

Webブラウザで上記のアドレスを呼び出すと、以下のとおりWebサービスが正常にデプロイされたことが確認できます。

[図 11.2] 正常にデプロイされたFrom WSDL方式のWebサービス



サンプルを簡単に実行

JEUS 8では、Webサービスのサンプルが簡単に実行できる方法を提供しています。

JEUS_HOME/samples/getting_started/webservices/from_java/の下位で以下のように実行すると、サービス・パッケージングからクライアントまで簡単に実行できます。

作成された例は、Antを使用してビルドできます。

```
%JEUS_HOME%/samples/getting_started/webservices/from_wsd1> ant
```

正常にビルドされるとWebブラウザが開かれ、結果を確認できます。

11.2. Webサービス・クライアントの作成

Webサービスにアクセスするには、Webサービス・クライアントを作成する必要があります。Webサービス・クライアント・プログラムの動作環境によって、Java SEクライアントとJava EEクライアントに分けられます。

本節では、一般のJavaプログラムと同様に動作されるJava SEクライアントについてのみ記述します。

11.2.1. Java SEクライアントの作成

JAX-WSでは、基本的にWebサービス・エンドポイントに対応される動的プロキシを内部的に生成し、このサービスをエンドポイント・インターフェースの実装であるかのように使用することができます。したがって、クライアント・プログラムの開発者は、生成されたプロキシを利用してサービス・エンドポイント・インターフェースとして定義されたWebサービス・メソッドを呼び出すことが可能です。

同作業を行うには、サービス・エンドポイント・インターフェースのようないくつかの結果が必要ですが、その結果をWSDLから生成する場合は、From WSDL方式のWebサービス生成と同様、**wsimport**ツールを使用します。

以下は、wsimportの使用例です。JEUS_HOME/samples/getting_started/webservices/from_wsdlディレクトリで以下のとおり実行します。

```
wsimport -p fromwsdl.client -d ./build/classes
http://localhost:8088/AddNumbers/addnumbers?wsdl
```

サービス・エンドポイント・インターフェースとサービス・クラスが生成されたら、これを使用するクライアントJavaプログラムを作成します。

以下は、Webサービス・クライアント・プログラミングの一例です。同例は、以前JAX-RPCで提供していたWebサービス・クライアント・プログラミングの例と類似しています。基本的にサービス・インターフェースを生成した後、同サービス・インターフェースからサービス・エンドポイント・インターフェースを実装したプロキシ・オブジェクトを取得するのが主な作業です。クライアント・ランタイムは内部的には大きく変わりましたが、Java SEクライアントは類似していると言えます。

サンプル・コードは、JEUS_HOME/samples/getting_started/webservices/from_wsdl/src/java/fromwsdl/clientに格納されています。

[例 11.3] <<AddNumbersClient.java>>

```
package fromwsdl.client;

public class AddNumbersClient {

    public static void main(String[] args) {
        AddNumbersPortType port = new AddNumbersService().getAddNumbersPort();
```

```

int number1 = 10;
int number2 = 20;

System.out.println("#####");
System.out.println("### JAX-WS Webservises examples - fromwsdl ###");
System.out.println("#####");
System.out.println("Testing Java class webservises from WSDL...");
int result = port.addNumbers(number1, number2);
if (result == 30) {
    System.out.println("Success!");
}
}
}

```

上記のようにクライアント・プログラムを作成した後、実際にコンパイルして実行すると、以下のような結果が出力されます。

```

[java] #####
[java] ### JAX-WS Webservises examples - fromwsdl ###
[java] #####
[java] Testing Java class webservises from WSDL...
[java] Success!

```


付録 A. 各プラットフォームのJDKインストール

本付録では、各プラットフォームのJDK(Java SE Development Kit)のインストールおよび使用方法について説明します。

参考

本付録は、インターネットで使用可能なJDKのインストール・ガイドを要約したものです。(2016年基準)

A.1. Sun Solaris

Solaris SPARCおよびx86のためのパッチファイルは、以下の「Solaris OS Patch」よりダウンロードできます。

<http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html>

JDKを使用する前にSolarisバージョンの必須パッチがインストールされているのかを確認します。JDKを実行するための最も簡単な方法は、ユーザーのシェルに存在するPATH環境変数をアップデートして「/usr/bin」の頭にJava実行可能ファイルのパスを入力します。

たとえば、JDKが/usr/j2seにインストールされている場合、以下のとおり設定します。

```
(csh) setenv PATH "/usr/j2se/bin:$PATH"
```

```
(sh or ksh) PATH="/usr/j2se/bin:$PATH"
```

また、実行可能ファイルは全体パス名(例: /usr/j2se/bin/javac, /usr/j2se/bin/java)を指定して実行することもできます。

JDKを実行するとき、別の環境変数(例: JDK_HOME, JAVA_HOMEまたはCLASSPATH)は不要です。しかし、既存設定の影響で正常に動作しない場合は、衝突有無を確認する必要があります。

A.2. IBM AIX

以下のパスよりJDKをダウンロードします。

<http://www.ibm.com/developerworks/java/jdk/aix/service.html>

JDKをインストールするには、システム・パッチが必要です。

インストール中にシステム要件を要求する場合、以下のパスから必要なパッケージをダウンロードしてインストールします。

```
ftp://service.software.ibm.com
```

上記のWebサイトにアクセスした後、anonymous/e-mailにログインします。リストを確認するには、必ずls -lを使用します。

以下のディレクトリー・パスから必要なファイルが取得できます。

- AIX update path

```
/aix/fixes/<該当するOSバージョン>
```

- Other update file

```
/aix/fixes/v4/other
```

A.3. HP-UX

Webサイトより、HP-UX JDK for the Java™ 2 Platform Standard Edition (J2SE)を/tmpディレクトリーにダウンロードします。

```
https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPUXJAVAHOME
```

同ソフトウェアをインストールするにはシステム・パッチが必要です。これについての情報は上記のWebサイトに記述されています。インストール方法およびコマンドについての情報は以下のパスの文書を参照してください。

```
https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=HPUXJAVAINFOLIB
```

A.4. Linux

以下のWebサイトより、JDKをダウンロードできます。

```
http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html
```


付録 B. IPv6の設定

本付録では、IPv6の設定について説明します。

B.1. 紹介

IPv6環境においてのJEUSの設定は以下の手順で行われます。

1. jeus.properties、jeusadmin、startDomainAdminServer、startDomainAdminServerNM、startManagedServer、stopServer、startNodeManager、stopNodeManager、startderby、stopderby、mcastReceiver、mcastSenderスクリプトの変更
2. domain.xmlの変更
3. 動作の確認

参考

設定するサーバーのhostsファイルのループバック・アドレスは「::1」に設定されている必要があります。

hostsファイルは、OSに応じて以下のパスに存在します。

- Windows

```
C:\WINDOWS\system32\drivers\etc
```

- UNIX系

```
/etc/hosts
```

hostsファイルの例は以下のとおりです。

[例 B.1] <<hosts>>

```
[jeusqa@ipv6linux /home/jeusqa]$ cat /etc/hosts
# Do not remove the following line, or various programs
# that require network functionality will fail.
#127.0.0.1        localhost
#::1             localhost6.localdomain6 localhost6
::1             localhost
```

注意

windowsでJDK 6を使用すると、JavaバグによりIPv6環境が正常に動作しない場合があるため、JDK 7の使用をお勧めします。詳しい内容は、http://bugs.java.com/view_bug.do?bug_id=6230761を参照してください。

B.2. IPv6の環境設定

IPv6を使用するための環境設定について説明します。IPv6の環境はファイル別に設定方法が異なります

JEUS_HOME/binに存在するファイルの変更

- ファイル名 : startDomainAdminServer、startManagedServer、startNodeManager

-classpath が作成されている箇所に、-Djava.net.preferIPv6Addresses=trueと
-Djava.net.preferIPv4Stack=falseを追加します。

– 例

```
-classpath ..... \
-Djava.net.preferIPv6Addresses=true \
-Djava.net.preferIPv4Stack=false \
```

- ファイル名 : mcastReceiver, mcastSender

– 既存

```
-Djava.net.preferIPv4Stack=true \
```

– 変更

```
-Djava.net.preferIPv6Addresses=true \
-Djava.net.preferIPv4Stack=false \
```

- ファイル名 : jeusadmin, startDomainAdminServerNM, stopNodeManager, stopServer

– 既存

```
"${JAVA_HOME}/bin/java" -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION} \
```

– 変更

```
"${JAVA_HOME}/bin/java" -classpath "${BOOTSTRAP_CLASSPATH}" ${TOOL_OPTION}
\
-Djava.net.preferIPv6Addresses=true \
-Djava.net.preferIPv4Stack=false \
```

● ファイル名 : jeus.properties

– 既存

```
TOOL_OPTION="-Djeus.tm.not_use=true -Djava.net.preferIPv4Stack=true"
```

– 変更

```
TOOL_OPTION="-Djeus.tm.not_use=true -Djava.net.preferIPv6Addresses=true
-Djava.net.preferIPv4Stack=false"
```

● ファイル名 : startderby, stopderby

– 既存

```
-Dderby.system.home="${JEUS_HOME}/derby/databases" \
```

– 変更

```
-Dderby.system.home="${JEUS_HOME}/derby/databases" \
-Djava.net.preferIPv6Addresses=true \
-Djava.net.preferIPv4Stack=false \
```

domain.xmlの変更

JEUS_HOME/domains/DOMAIN_NAME/config/domain.xmlファイルの内容を以下のように変更および追加します。

1. listen-addressのアドレスがIPv4に設定されている場合はIPv6に変更します。

– 例

```
<listeners>
  <base>BASE</base>
  <listener>
    <name>BASE</name>
    <listen-address>0:0:0:0:0:0:0:0</listen-address>
```

```

<listen-port>9736</listen-port>
<selectors>1</selectors>
<use-dual-selector>false</use-dual-selector>
<backlog>128</backlog>
<select-timeout>120000</select-timeout>
<read-timeout>30000</read-timeout>
<reserved-thread-num>0</reserved-thread-num>
</listener>

```

2. heartbeat-addressのアドレスがIPv4に設定されている場合は、以下のとおりIPv6に変更する必要があります。設定がない場合は以下の内容を追加します。

– 例

```

<group-communication-info>
  <heartbeat-address>FF02:0:0:0:0:0:0:0</heartbeat-address>
  <heartbeat-port>3030</heartbeat-port>
  <use-virtual-multicast>false</use-virtual-multicast>
</group-communication-info>

```

3. 各サーバーのvm-optionにjava.net.preferIPv4Stack=falseとjava.net.preferIPv6Addresses=trueを追加します。

– 例

```

<jvm-config>
  <jvm-option>-Xmx1024m -XX:MaxPermSize=256m</jvm-option>
  <jvm-option>java.net.preferIPv4Stack=false</jvm-option>
  <jvm-option>java.net.preferIPv6Addresses=true</jvm-option>
</jvm-config>

```

付録 C. Windowsサービスの設定

本付録では、Windowsサービスの登録時に必要なファイル設定について説明します。

以下は、サービス名が「jeus」の場合、Windowsサービス登録ファイルの例です。(サービス名に応じてファイル名が設定されます。)

[例 C.1] <<jeus.ini>>

```
[jeus]
jeus.home=C:\TmaxSoft\JEUS8.0
jeus.description=JEUS8 Windows service
java.home=C:\Java\jdk1.7.0_21
java.vm.type=hotspot
java.vm.option=-server
jeus.nm.classpath=C:\TmaxSoft\JEUS8.0\lib\system\jeus-nodemanager.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\jeusapi.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\jeusutil.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\jeus-network.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\jmxremote.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\jeusjaxb.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\jmx-description.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\javaee.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\tmaxjce_jdk15x.jar;
                  C:\TmaxSoft\JEUS8.0\lib\system\commons-cli.jar
jeus.nm.boot.param=-domain jeus_domain -server adminServer
jeus.nm.down.param=-stopServer
shutdown.previous.nodemanager=true
-Xmx1024m
```

以下は、各設定項目についての説明です。

項目	説明
jeus.home	JEUSがインストールされたディレクトリー・パスです
jeus.description	Windowsサービスについての説明です
java.home	Javaがインストールされたディレクトリー・パスです
java.vm.type	Java HotSpot JVMの使用有無を設定します。 以下の値に設定します

項目	説明
	<ul style="list-style-type: none"> – hotspot : Java HotSpot JVMを使用する場合に設定するオプションです。java.vm.optionの-serverオプションで動作します – old : Java HotSpot JVMを使用しない場合に設定するオプションです。vmオプションが入りません。vmオプションが動作しないサーバーは、「old」に設定する必要があります
java.vm.option	<p>Java HotSpot JVMの使用有無に応じたオプションを設定します。</p> <p>以下の値に設定します</p> <ul style="list-style-type: none"> – HotSpot JVMを使用する場合 : -server – HotSpot JVMを使用しない場合 : -Djeus.dispatcher.blocking=true
jeus.nm.classpath	<p>ノード・マネージャーを起動するときに必要なclasspathのパスを設定します</p>
jeus.nm.boot.param	<p>Windowsサービスの開始と同時にノード・マネージャーが起動されますが、そのとき使用する追加的なオプションを設定します。</p> <p>現在は、Windowsサービスを開始すると同時にJEUSで起動するDASサーバーを設定することができます。ドメイン名とサーバー名を一緒に設定します。</p> <p>(例: -domain jeus_domain -server adminServer)</p> <p>何にも設定しない場合は、ノード・マネージャーのみ起動されます</p>
jeus.nm.down.param	<p>Windowsサービスを終了するときに使用する追加的なオプションを設定します。</p> <p>現在は、Windowsサービスの終了と同時にJEUSを終了するオプションを設定することができます</p> <p>(例: -stopServer)</p>
shutdown.previous.nodemanager	<p>Windowsサービスを開始するとき、すでに実行されているJEUSを終了してから、Windowsサービスを実行するかどうかを設定します。</p> <p>falseに設定されている場合、すでにJEUSが実行されている状態でWindowsサービスを開始することはできません</p>