



Sun N1 Service Provisioning System ユーザーズガイド (BEA WebLogic 6 と WebLogic 7 Plug-In 2.0)



Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 819-6496-10
2006年4月

本製品およびそれに関連する文書は著作権法により保護されており、その使用、複製、頒布および逆コンパイルを制限するライセンスのもとにおいて頒布されます。サン・マイクロシステムズ株式会社の書面による事前の許可なく、本製品および関連する文書のいかなる部分も、いかなる方法によっても複製することが禁じられます。

本製品の一部は、カリフォルニア大学からライセンスされている Berkeley BSD システムに基づいていることがあります。UNIX は、X/Open Company, Ltd. が独占的にライセンスしている米国ならびに他の国における登録商標です。フォント技術を含む第三者のソフトウェアは、著作権により保護されており、提供者からライセンスを受けているものです。

U.S. Government Rights Commercial software. Government users are subject to the Sun Microsystems, Inc. standard license agreement and applicable provisions of the FAR and its supplements.

本製品に含まれる HG-MinchoL、HG-MinchoL-Sun、HG-PMinchoL-Sun、HG-GothicB、HG-GothicB-Sun、および HG-PGothicB-Sun は、株式会社リコーがリョービマジクス株式会社からライセンス供与されたタイプフェースマスタをもとに作成されたものです。HeiseiMin-W3H は、株式会社リコーが財団法人日本規格協会からライセンス供与されたタイプフェースマスタをもとに作成されたものです。フォントとして無断複製することは禁止されています。

Sun、Sun Microsystems、docs.sun.com、および Java は、米国およびその他の国における米国 Sun Microsystems, Inc. (以下、米国 Sun Microsystems 社とします) の商標、登録商標もしくは、サービスマークです。

サンのロゴマーク、Solaris のロゴマーク、Java Coffee Cup のロゴマーク、および Solaris は、米国 Sun Microsystems 社の登録商標です。

すべての SPARC 商標は、米国 SPARC International, Inc. のライセンスを受けて使用している同社の米国およびその他の国における商標または登録商標です。SPARC 商標が付いた製品は、米国 Sun Microsystems 社が開発したアーキテクチャに基づくものです。

OPENLOOK、OpenBoot、JLE は、サン・マイクロシステムズ株式会社の登録商標です。

Wnn は、京都大学、株式会社アステック、オムロン株式会社で共同開発されたソフトウェアです。

Wnn6 は、オムロン株式会社、オムロンソフトウェア株式会社で共同開発されたソフトウェアです。Copyright OMRON Co., Ltd. 1995-2000. All Rights Reserved. © Copyright OMRON SOFTWARE Co., Ltd. 1995-2002 All Rights Reserved. ©

「ATOK」は、株式会社ジャストシステムの登録商標です。

「ATOK Server/ATOK12」は、株式会社ジャストシステムの著作物であり、「ATOK Server/ATOK12」にかかる著作権その他の権利は、株式会社ジャストシステムおよび各権利者に帰属します。

「ATOK Server/ATOK12」に含まれる郵便番号辞書(7桁/5桁)は日本郵政公社が公開したデータを元に制作された物です(一部データの加工を行なっています)。

「ATOK Server/ATOK12」に含まれるフェイスマーク辞書は、株式会社ビレッジセンターの許諾のもと、同社が発行する『インターネット・パソコン通信フェイスマークガイド』に添付のものを使用しています。

Unicode は、Unicode, Inc. の商標です。

本書で参照されている製品やサービスに関しては、該当する会社または組織に直接お問い合わせください。

OPEN LOOK および Sun Graphical User Interface は、米国 Sun Microsystems 社が自社のユーザおよびライセンス実施権者向けに開発しました。米国 Sun Microsystems 社は、コンピュータ産業用のビジュアルまたはグラフィカル・ユーザインタフェースの概念の研究開発における米国 Xerox 社の先駆者としての成果を認めるものです。米国 Sun Microsystems 社は米国 Xerox 社から Xerox Graphical User Interface の非独占的ライセンスを取得しており、このライセンスは、OPEN LOOK のグラフィカル・ユーザインタフェースを実装するか、またはその他の方法で米国 Sun Microsystems 社との書面によるライセンス契約を遵守する、米国 Sun Microsystems 社のライセンス実施権者にも適用されます。

本書は、「現状のまま」をベースとして提供され、商品性、特定目的への適合性または第三者の権利の非侵害の黙示の保証を含みそれに限定されない、明示的であるか黙示的であるかを問わない、なんらの保証も行われぬものとします。

本製品が、外国為替および外国貿易管理法(外為法)に定められる戦略物資等(貨物または役務)に該当する場合、本製品を輸出または日本国外へ持ち出す際には、サン・マイクロシステムズ株式会社の事前の書面による承諾を得ることのほか、外為法および関連法規に基づく輸出手続き、また場合によっては、米国商務省または米国所轄官庁の許可を得ることが必要です。

原典: Sun N1 Service Provisioning System User's Guide for BEA WebLogic 6 and WebLogic 7 Plug-In 2.0

Part No. 819-6391-10

目次

はじめに	7
1 WebLogic プラグインの概要	11
WebLogic プラグインの目的	11
BEA WebLogic プラグインに含まれるもの	11
BEA WebLogic プラグインを使用するための要件	12
2 WebLogic 6 および 7 プラグインのリリースノート	15
インストール上の問題	15
実行時の問題	15
WebLogic 7 Admin Server が正常に停止したあとエラーが表示される (6305069)	15
3 BEA WebLogic プラグインのインストールと構成	17
BEA WebLogic プラグインの入手	17
Solaris への BEA WebLogic プラグインの追加	18
▼ Solaris 用の BEA WebLogic プラグインを追加する	18
Linux への BEA WebLogic プラグインの追加	18
▼ Linux 用の BEA WebLogic プラグインを追加する	18
Windows への BEA WebLogic プラグインの追加	19
▼ Windows 用の BEA WebLogic プラグインを追加する	19
Sun N1 Service Provisioning System への BEA WebLogic プラグインのインポート	19
▼ ブラウザインタフェースを使用して BEA WebLogic プラグインをインポートする	19
▼ CLI を使用して BEA WebLogic プラグインをインポートする	20
環境に応じたソリューションのカスタマイズ	20
BEA WebLogic プラグインへのパッチの適用	20

4 BEA WebLogic プラグインの使用法	21
WebLogic サーバーおよびクラスタの作成と管理	21
▼ 7.0 用の WebLogic Admin Server を作成する	22
▼ 6.1 用の WebLogic Admin Server を作成する	23
▼ WebLogic クラスタを作成する	25
▼ WebLogic サーバーインスタンスを管理する	27
WebLogic アプリケーションの取得と編集	30
▼ WebLogic エンタープライズアプリケーションまたは EAR ファイルを取得する	30
▼ WebLogic Web アプリケーションまたは WAR ファイルを取得する	31
▼ WebLogic JAR ファイルを取得する	31
コンポーネントタイプ	32
エンタープライズアプリケーション (EAR) コンポーネントタイプ	32
Web アプリケーション (WAR) コンポーネントタイプ	33
EJB コンポーネントタイプを含む Java アーカイブファイル	34
プラン	35
リソース	35
索引	37

表目次

表 4-1	WebLogic エンタープライズアプリケーションのエラー状況	33
表 4-2	WebLogic Web アプリケーションのエラー状況	34
表 4-3	WebLogic EJB コンテナのエラー状態	35

はじめに

このマニュアルでは、Sun N1 Service Provisioning System ソフトウェアを使用して BEA WebLogic 6.1 および 7 のアプリケーションとファイルを取得して配備する方法について説明します。

対象読者

このマニュアルの主な対象読者は、BEA WebLogic 6.1 および 7 の機能を Sun N1 Service Provisioning System ソフトウェアに組み込む能力のある N1 Service Provisioning System ソフトウェアのシステム管理者およびオペレータです。このようなユーザーは、次のことに精通している必要があります。

- N1 Service Provisioning System 製品
- UNIX® と Windows の一般的なコマンドとユーティリティー
- BEA WebLogic 6.1 および 7 製品の一般的な概念と管理機能

お読みになる前に

Sun N1 Service Provisioning System ソフトウェアの使用法に関する十分な知識がないユーザーは、次のマニュアルを参照してください。

- 『Sun N1 Service Provisioning System 5.2 システム管理者ガイド』
- 『Sun N1 Service Provisioning System 5.2 オペレーションとプロビジョニングガイド』
- 『Sun N1 Service Provisioning System 5.2 リリースノート』

マニュアルの構成

第 1 章では、BEA WebLogic プラグインソリューションの概要を説明します。

第 3 章では、このプラグインのインストールと構成の方法について説明します。

第 4 章では、このプラグインを使用して BEA WebLogic アプリケーションとファイルを取得して配備する方法を説明し、さらに WebLogic 固有のコンポーネントタイプについて説明します。

Sun 以外の Web サイト

このマニュアルで参照している Sun 以外の URL は、追加の関連情報を提供します。

注 - このマニュアル内で引用する Sun 以外の Web サイトの可用性については、Sun は責任を負いません。Sun は、これらのサイトあるいはリソースに関する、あるいはこれらのサイト、リソースから利用可能であるコンテンツ、広告、製品、あるいは資料に関して一切の責任を負いません。こうしたサイトやリソース上で、またはこれらを経由して利用できるコンテンツ、製品、サービスを利用または信頼したことに伴って発生した (あるいは発生したと主張される) 実際の (あるいは主張される) 損害や損失についても、Sun は一切の責任を負いません。

マニュアル、サポート、およびトレーニング

Sun の Web サイトでは、以下の追加情報を提供しています。

- マニュアル (<http://jp.sun.com/documentation/>)
- サポート (<http://jp.sun.com/support/>)
- トレーニング (<http://jp.sun.com/training/>)

表記上の規則

このマニュアルでは、次のような字体や記号を特別な意味を持つものとして使用しません。

表 P-1 表記上の規則

字体または記号	意味	例
AaBbCc123	コマンド名、ファイル名、ディレクトリ名、画面上のコンピュータ出力、コード例を示します。	.login ファイルを編集します。 ls -a を使用してすべてのファイルを表示します。 machine_name% you have mail.
AaBbCc123	ユーザーが入力する文字を、画面上のコンピュータ出力と区別して示します。	machine_name% su Password:
<i>AaBbCc123</i>	変数を示します。実際に使用する特定の名前または値で置き換えます。	ファイルを削除するには、rm filename と入力します。

表 P-1 表記上の規則 (続き)

字体または記号	意味	例
『』	参照する書名を示します。	『コードマネージャ・ユーザーズガイド』を参照してください。
「」	参照する章、節、ボタンやメニュー名、強調する単語を示します。	第5章「衝突の回避」を参照してください。 この操作ができるのは、「スーパーユーザー」だけです。
\	枠で囲まれたコード例で、テキストがページ行幅を超える場合に、継続を示します。	sun% grep '^#define \ XV_VERSION_STRING'

コード例は次のように表示されます。

- C シェル

```
machine_name% command y|n [filename]
```

- C シェルのスーパーユーザー

```
machine_name# command y|n [filename]
```

- Bourne シェルおよび Korn シェル

```
$ command y|n [filename]
```

- Bourne シェルおよび Korn シェルのスーパーユーザー

```
# command y|n [filename]
```

[] は省略可能な項目を示します。上記の例は、*filename* は省略してもよいことを示しています。

| は区切り文字 (セパレータ) です。この文字で分割されている引数のうち 1 つだけを指定します。

キーボードのキー名は英文で、頭文字を大文字で示します (例: Shift キーを押します)。ただし、キーボードによっては Enter キーが Return キーの動作をします。

ダッシュ (-) は 2 つのキーを同時に押すことを示します。たとえば、Ctrl-D は Control キーを押したまま D キーを押すことを意味します。

WebLogic プラグインの概要

この章では、Sun N1 Service Provisioning System を使用した BEA WebLogic 6.1 および7アプリケーションのプロビジョニングに関する一般的な情報を説明します。この章では、次の内容について説明します。

- 11 ページの「WebLogic プラグインの目的」
- 11 ページの「BEA WebLogic プラグインに含まれるもの」
- 12 ページの「BEA WebLogic プラグインを使用するための要件」

WebLogic プラグインの目的

Sun N1 Service Provisioning System ソフトウェアは、BEA WebLogic 6.1 および7アプリケーションを難しい設定などは一切せずにサポートするための拡張機能を提供します。ユーザーは、BEA WebLogic アプリケーションをリファレンスサーバーから取得し、どのように構成するべきかを細かく選択した上で、スタンドアロンタイプ、管理タイプ、またはクラスタタイプの BEA WebLogic 環境にこのアプリケーションを配備できます。

注 - このプラグインを使用して BEA WebLogic 環境を管理する場合、プロビジョニングソフトウェアの外部への BEA WebLogic アプリケーションの配備やプロビジョニングソフトウェアの外部での BEA WebLogic アプリケーションの管理は避けてください。BEA WebLogic アプリケーションは、必ず Sun N1 Service Provisioning System ソフトウェアを使用して管理する必要があります。

BEA WebLogic プラグインに含まれるもの

BEA WebLogic プラグインには、WebLogic Enterprise アプリケーション、Enterprise JavaBeans (EJB) アプリケーション、および Web アプリケーションの取得、構成、配備を簡易化する定義済みコンポーネントとリソースファイルのほか、WebLogic 固有のコンポーネントタイプがいくつか含まれています。これらのコンポーネントタイプは次の3つのファミリに分類できます。

- 次に示す 3 種類の WebLogic 仮想ホストをサポートするコンポーネントタイプ
 - WebLogic Admin Server
 - WebLogic (Managed) Server
 - WebLogic クラスタ
- WebLogic 6.x および 7.x アプリケーションサーバー上への配備用アプリケーションの配備やプルを行う特殊なコンポーネント
- WebLogic プラットフォームの Enterprise JavaBeans™ 技術ベース API を使用して実装される特殊なコンポーネントロジック

BEA WebLogic プラグインを使用するための要件

BEA WebLogic プラグインの配備先となるホストは、次の要件を満たす必要があります。

WebLogic ソフトウェア

BEA Systems WebLogic バージョン 6.1 または 7.0 が Sun N1 Service Provisioning System 環境内に存在すること。これは、配備するアプリケーションとファイルを取得できるようにするためです。

オペレーティングシステム

BEA System で指定されている適切なバージョンの UNIX または Windows オペレーティングシステム

ディスク容量

Windows システムの場合、製品のインストール領域として約 236M バイトの空き容量と、インストーラが使用する一時的な格納領域として 170M バイト

UNIX システムの場合、製品のインストール領域として約 196M バイトの空き容量と、インストーラが使用する一時的な格納領域として 162M バイト

RAM

256M バイト (最小限); 512M バイト (推奨)

Java run-time environment (JRE)

WebLogic Server のインストールプログラムは、実行に JRE が必要です。Windows インストールプログラムと、一部の UNIX インストールプログラム (ファイル名が .bin で終了する) には、JRE を含む Java 2 Software Development Kit (SDK) が含まれています。ほかの UNIX プラットフォームについては、WebLogic Server インストールプログラム (ファイル名が .jar で終了する) に Java 2 SDK は含まれていません。 .jar インストールプログラムを実行するには、適切なバージョンの Java 2 SDK をシステムにインストールするとともに、その SDK の bin ディレクトリを PATH 変数の先頭に含める必要があります。SDK を使用する必要があるのは、このディレクトリを指すようにインストールプロセスによって JAVA_HOME 変数と関連変数が設定されるためです。

また、WebLogic アプリケーションを取得し、配備可能な最新の WebLogic 環境も用意する必要があります。この環境は、次の要件を満たすものでなければなりません。

WebLogic アプリケーション 必要に応じて構成された WebLogic 6.1 または 7.0 アプリケーションが稼働している

WebLogic 6 および 7 プラグインのリリースノート

この章では、WebLogic 6 および 7 プラグインの最新情報、および既知の問題について説明します。この章では、次の項目について説明します。

- 15 ページの「インストール上の問題」
- 15 ページの「実行時の問題」

インストール上の問題

インストールに関する問題はありません。

実行時の問題

WebLogic 6 および 7 プラグインの使用時には、次の問題が発生することが判明しています。

WebLogic 7 Admin Server が正常に停止したあとエラーが表示される (6305069)

説明: コンポーネントの stopServer コンポーネントプロシージャを使用して WebLogic 7 Admin Server を停止すると、プロビジョニングソフトウェアは作業が正常に完了したあとでエラーメッセージを生成します。

回避策: エラーメッセージを無視します。

BEA WebLogic プラグインのインストールと構成

この章では、BEA WebLogic プラグインをインストールして構成する方法について説明します。この章では、次の内容について説明します。

- 17 ページの「BEA WebLogic プラグインの入手」
- 19 ページの「Sun N1 Service Provisioning System への BEA WebLogic プラグインのインストール」
- 20 ページの「環境に応じたソリューションのカスタマイズ」
- 20 ページの「BEA WebLogic プラグインへのパッチの適用」

BEA WebLogic プラグインの入手

BEA WebLogic プラグインを入手する手順は2段階に分かれます。はじめに BEA WebLogic プラグイン JAR ファイルが含まれたパッケージファイルをシステムに追加する必要があります。そのあとに BEA WebLogic プラグイン JAR ファイルをインポートします。

BEA WebLogic プラグインは、Sun N1 Service Provisioning System ソフトウェアに対するプラグインとしてパッケージ化されています。プラグインは Java™ アーカイブ (JAR) ファイルとしてパッケージ化されています。BEA WebLogic プラグイン用のプラグインファイルは、Sun N1 Service Provisioning System Supplement CD または Sun Download Center から入手できます。

パッケージファイルをシステムに追加すると、BEA WebLogic 6.1 および 7 プラグインは2種類の JAR ファイルからインポートできます。状況に応じて、適切なファイルを選択してください。

- BEA WebLogic プラグインをはじめてインポートする場合は、`com.sun.weblogic_2.0.jar` ファイルを使用します。
- 旧バージョンの BEA WebLogic プラグインがすでにインポートされている場合は、`com.sun.weblogic_1.1_2.0.jar` ファイルを使用します。

1. JAR ファイルを含むパッケージを追加します。
 - 18 ページの「Solaris への BEA WebLogic プラグインの追加」

- 18 ページの「Linux への BEA WebLogic プラグインの追加」
 - 19 ページの「Windows への BEA WebLogic プラグインの追加」
2. JAR ファイルをインポートします。- 19 ページの「Sun N1 Service Provisioning System への BEA WebLogic プラグインのインポート」

Solaris への BEA WebLogic プラグインの追加

BEA WebLogic プラグインは、SUNWspswlg パッケージに含まれています。

▼ Solaris 用の BEA WebLogic プラグインを追加する

- 1 端末ウィンドウで、スーパーユーザーになります。
- 2 プラグインパッケージを含むディレクトリに移動します。
- 3 次のコマンドを入力し、Return キーを押します。

```
# pkgadd -d . SUNWspswlg
```

スタンドアロン用 JAR ファイルは /opt/SUNWn1sps/plugins/com.sun.weblogic/ ディレクトリにあります。アップグレード用 JAR ファイルは /opt/SUNWn1sps/plugins/com.sun.weblogic/Upgrade ディレクトリにあります。

Linux への BEA WebLogic プラグインの追加

BEA WebLogic プラグインは、sun-spswlg-2.0-1.noarch.rpm ファイルに含まれています。

▼ Linux 用の BEA WebLogic プラグインを追加する

- 1 端末ウィンドウで、スーパーユーザーになります。
- 2 sun-spswlg-2.0-1.noarch.rpm ファイルを含むディレクトリに移動します。
- 3 次のコマンドを入力し、Return キーを押します。

```
# rpm -i sun-spswlg-2.0-1.noarch.rpm
```

スタンドアロン用 JAR ファイルは /opt/sun/N1_Service_Provisioning_System/plugins/com.sun.weblogic/ ディレクトリにあります。アップグレード用 JAR ファイルは /opt/sun/N1_Service_Provisioning_System/plugins/com.sun.weblogic/Upgrade ディレクトリにあります。

Windows への BEA WebLogic プラグインの追加

BEA WebLogic プラグインは、sun-spswlg-2.0.msi ファイルに含まれています。

▼ Windows 用の BEA WebLogic プラグインを追加する

- 1 sun-spswlg-2.0.msi ファイルを含むディレクトリに移動します。
- 2 sun-spswlg-2.0.msi ファイルをダブルクリックします。
インストーラの GUI が起動されます。JAR ファイルは、C:\Program Files\N1 Service Provisioning System\plugins\com.sun.weblogic ディレクトリにコピーされます。

Sun N1 Service Provisioning System への BEA WebLogic プラグインのインポート

ある特定のプラグインを Sun N1 Service Provisioning System に認識させるには、そのプラグインをマスターサーバーにインポートする必要があります。旧バージョンの BEA WebLogic プラグインがすでにインポートされている場合は、新しいプラグインにアップグレードする必要があります。

▼ ブラウザインタフェースを使用して BEA WebLogic プラグインをインポートする

プラグインをインポートまたはアップグレードするには、『Sun N1 Service Provisioning System 5.2 システム管理者ガイド』の第 5 章「プラグインの管理」で詳しく説明されている手順に従います。

- 1 ブラウザインタフェースのメインウィンドウの「Administrative」セクションで、「Plug-ins」をクリックします。
- 2 「Plug-ins」ページの「Action」列で「Import」をクリックします。
- 3 JAR ファイルの場所に移動します。
 - BEA WebLogic プラグインをはじめてインポートする場合は、com.sun.weblogic_2.0.jar ファイルを選択します。
 - 旧バージョンの BEA WebLogic プラグインがすでにインポートされている場合は、com.sun.weblogic_1.1_2.0.jar ファイルを選択します。
- 4 「Continue to Import」ボタンをクリックします。
インポートが正常に終了すると、プラグインの詳細ページが開いて、そのプラグインが提供するオブジェクトが表示されます。

▼ CLI を使用して BEA WebLogic プラグインをインポートする

コマンド行を使用して、プラグインをインポートすることもできます。

- ▶ CLI からプラグインファイルをインポートするには、次のように入力します。
% `cr_cli -cmd plg.p.add -path plugin-filename -u username -p password`
 - BEA WebLogic プラグインをはじめてインポートする場合は、`plugin-filename` は `com.sun.weblogic_2.0.jar` になります。
 - 旧バージョンの BEA WebLogic プラグインがすでにインポートされている場合は、`plugin-filename` は `com.sun.weblogic_1.1_2.0.jar` になります。

環境に応じたソリューションのカスタマイズ

個々の環境に合わせて BEA WebLogic プラグインをカスタマイズする方法については、`com.sun.weblogic_2.0.jar` または `com.sun.weblogic_1.1_2.0.jar` の `readme.txt` ファイルで説明されています。

BEA WebLogic プラグインへのパッチの適用

BEA WebLogic プラグインに適用可能なパッチは、SunSolve (<http://sunsolve.sun.com>) サイトを確認します。パッチを適用するには、パッチの README ファイルの手順に従います。

◆ ◆ ◆ 第 4 章

BEA WebLogic プラグインの使用法

BEA WebLogic プラグインは特別なコンポーネントタイプを提供し、WebLogic アプリケーションの作業に必要な機能に簡単にアクセスできるようにします。この章では、次の内容について説明します。

- 21 ページの「WebLogic サーバーおよびクラスタの作成と管理」
- 30 ページの「WebLogic アプリケーションの取得と編集」
- 32 ページの「コンポーネントタイプ」
- 35 ページの「プラン」

注 - installPath 変数の値は、/opt や c:mydir などの絶対パスを指定しないかぎり、デフォルトのリモートエージェントディレクトリの相対パスとして扱われます。たとえば、Windows リモートエージェントで installPath 変数を c/mydir に設定し、デフォルトのリモートエージェントディレクトリ c:\Program Files\N1 Service Provisioning System\agent でエージェントにファイルを配備すると、ファイルは c:\Program Files\N1 Service Provisioning System\agent\c\mydir に配備されます。

WebLogic サーバーおよびクラスタの作成と管理

プロビジョニングソフトウェアで WebLogic サーバーと WebLogic クラスタの作成と管理を行うには、「WebLogic Common Tasks」ページを使用します。具体的には次のような作業を実行できます。

- 22 ページの「7.0 用の WebLogic Admin Server を作成する」
- 23 ページの「6.1 用の WebLogic Admin Server を作成する」
- 25 ページの「WebLogic クラスタを作成する」
- 27 ページの「WebLogic サーバーインスタンスを管理する」

▼ 7.0 用の WebLogic Admin Server を作成する

- 1 ブラウザインタフェースの「Common Tasks」セクションで、「WebLogic」をクリックします。
- 2 「WebLogic Common Tasks」ページで、「Manage 7.0 Admin Servers」をクリックします。
- 3 7.0 Admin Server コンポーネントの「Components Details」ページで、「default:install」の隣にある「Run action」をクリックします。
- 4 この Admin Server の変数を設定するには、ウィンドウの「Plan Parameters」セクションにある「Select from List」をクリックします。
 - デフォルト値とは異なる値を持つ新しい変数設定セットを作成するには、「Create Set」をクリックします。

「Select Variable Setting From List」ウィンドウが表示されます。

 - a. テーブルの最上部にあるテキストフィールドに、新しい変数設定セットの名前を入力します。
 - b. WebLogic Admin Server の名前を指定するには、「adminHost」の隣にあるボックスをクリックし、フィールド内に名前を入力します。
 - c. WebLogic Admin Server のデフォルトのポート番号値 7001 を変更するには、「adminPort」の隣にあるボックスをクリックし、フィールド内に新しい番号を入力します。
 - d. セキュリティ保護された HTTP を使用して WebLogic Admin Server に接続するよう指定するには、「secureConnect」の隣にあるボックスをクリックし、フィールド内に「True」と入力します。
 - e. WebLogic のインストール先となるパスを変更するには、「wlHomeDir」の隣にあるボックスをクリックし、フィールド内にパスを入力します。
 - f. WebLogic ドメイン名を変更するには、「domainName」の隣にあるボックスをクリックし、フィールド内にドメイン名を入力します。

ほかの変数はあらかじめ定義されていますが、必要に応じて変更することもできます。

 - targetRefName は、ホスト名の値とドメイン名の値に基づいて作成されます。この変数は、`: [target:sys.hostName]_admin_: [domainName]` という書式になります。
 - installPath は、ドメイン名の値に基づいて作成されます。この変数は、`admin_: [domainName]` という書式になります

- webLogicJARPath は、WebLogic ホームディレクトリの値に基づいて作成されます。この変数は、
:[wlHomeDir]:[/]weblogic700:[/]server:[/]lib:[/]weblogic.jar という書式になります。
 - domainRoot は user_projects です
- g. コンポーネント変数の値を更新したあと、「Save」をクリックします。
テーブル内に新しい変数設定が表示されます。
 - h. 「Select」をクリックします。
- 別のコンポーネントの変数設定を使用するには、「Import Set」をクリックします。
「Import Variable Settings」ウィンドウが表示されます。
 - a. 必要に応じ、インポートする変数設定がなされているコンポーネントが入ったフォルダに移動します。
 - b. コンポーネントのバージョンを選択します。
-
- 注-変数設定は、コンポーネントのバージョンごとに異なってもかまいません。
- c. 「Import Variable Settings」をクリックします。
インポートされた変数設定がテーブル内に表示されます。
 - d. 「Select」をクリックします。
- 5 ターゲットホストを選択します。
 - 6 「Run Plan (includes preflight)」をクリックします。

▼ 6.1 用の WebLogic Admin Server を作成する

- 1 ブラウザインタフェースの「Common Tasks」セクションで、「WebLogic」をクリックします。
- 2 「WebLogic Common Tasks」ページで、「Manage 6.1 Admin Servers」をクリックします。
- 3 6.1 Admin Server コンポーネントの「Components Details」ページで、「default:install」の隣にある「Run action」をクリックします。

- 4 この **Admin Server** の変数を設定するには、ウィンドウの「**Plan Parameters**」セクションにある「**Select from List**」をクリックします。
- デフォルト値とは異なる値を持つ新しい変数設定セットを作成するには、「**Create Set**」をクリックします。
 - 「**Select Variable Setting From List**」ウィンドウが表示されます。
 - a. テーブルの最上部にあるテキストフィールドに、新しい変数設定セットの名前を入力します。
 - b. **WebLogic Admin Server** の名前を指定するには、「**adminHost**」の隣にあるボックスをクリックし、フィールド内に名前を入力します。
 - c. **WebLogic Admin Server** のデフォルトのポート番号値 **7001** を変更するには、「**adminPort**」の隣にあるボックスをクリックし、フィールド内に新しい番号を入力します。
 - d. セキュリティー保護された **HTTP** を使用して **WebLogic Admin Server** に接続するよう指定するには、「**secureConnect**」の隣にあるボックスをクリックし、フィールド内に「**True**」と入力します。
 - e. **WebLogic** のインストール先となるパスを変更するには、「**wlHomeDir**」の隣にあるボックスをクリックし、フィールド内にパスを入力します。
 - f. **WebLogic** ドメイン名を変更するには、「**domainName**」の隣にあるボックスをクリックし、フィールド内にドメイン名を入力します。

ほかの変数はあらかじめ定義されていますが、必要に応じて変更することもできます。

 - **targetRefName** は、ホスト名の値とドメイン名の値に基づいて作成されます。この変数は、`:[target:sys.hostName]_admin_: [domainName]` という書式になります。
 - **installPath** は、ドメイン名の値に基づいて作成されます。この変数は、`admin_: [domainName]` という書式になります
 - **webLogicJARPath** は、**WebLogic** ホームディレクトリの値に基づいて作成されます。この変数は、`:[wlHomeDir]: [/]lib: [/]weblogic.jar` という書式になります。
 - g. コンポーネント変数の値を更新したあと、「**Save**」をクリックします。

テーブル内に新しい変数設定が表示されます。
 - h. 「**Select**」をクリックします。

- 別のコンポーネントの変数設定を使用するには、「**Import Set**」をクリックします。
「**Import Variable Settings**」ウィンドウが表示されます。
- a. 必要に応じ、インポートする変数設定がなされているコンポーネントが入ったフォルダに移動します。

- b. コンポーネントのバージョンを選択します。

注-変数設定は、コンポーネントのバージョンごとに異なってきかまいません。

- c. 「**Import Variable Settings**」をクリックします。
インポートされた変数設定がテーブル内に表示されます。
- d. 「**Select**」をクリックします。

- 5 ターゲットホストを選択します。

- 6 「**Run Plan (includes preflight)**」をクリックします。

▼ WebLogic クラスタを作成する

WebLogic サーバークラスタは複数の WebLogic サーバーインスタンスから構成されています。これらのサーバーインスタンスが同時に実行し、連携動作することで拡張性と信頼性が向上しています。クライアントから見ると、1つのクラスタが単一の WebLogic サーバーインスタンスのように見えます。クラスタを構成するサーバーインスタンスは、同一マシン上で実行したり、複数のマシンに配置することができます。クラスタ内の各サーバーインスタンスは、同じバージョンの WebLogic サーバーを使用する必要があります。

注 - Sun N1 Service Provisioning System 環境では、クラスタを作成してから、そのクラスタ内に配置する管理対象サーバーを作成する必要があります。変数リストの引数のうちクラスタホスト名は、管理対象サーバーを作成する際には省略可能な引数の1つです。このクラスタホスト名は、管理対象サーバーのインストール時にそのサーバーに対する依存関係を作成するために使用されるものです。管理対象サーバーをクラスタよりも先に作成した場合にはクラスタホストは存在せず、依存関係の作成は失敗します。

- 1 ブラウザインタフェースの「**Common Tasks**」セクションで、「**WebLogic**」をクリックします。
- 2 「**WebLogic Common Tasks**」ページで、「**Manage Clusters**」をクリックします。
- 3 **WebLogic** クラスタコンポーネントの「**Components Details**」ページで、「**default:install**」の隣にある「**Run action**」をクリックします。

- 4 このクラスタの変数を設定するには、ウィンドウの「**Plan Parameters**」セクションにある「**Select from List**」をクリックします。
 - デフォルト値とは異なる値を持つ新しい変数設定セットを作成するには、「**Create Set**」をクリックします。
 - 「**Select Variable Setting From List**」ウィンドウが表示されます。
 - a. テーブルの最上部にあるテキストフィールドに、新しい変数設定セットの名前を入力します。
 - b. クラスタの名前を指定するには、「**targetName**」の隣にあるボックスをクリックし、フィールド内にクラスタ名を入力します。
 - c. コンポーネント変数の値を更新したあと、「**Save**」をクリックします。
テーブル内に新しい変数設定が表示されます。
 - d. 「**Select**」をクリックします。
 - 別のコンポーネントの変数設定を使用するには、「**Import Set**」をクリックします。
 - 「**Import Variable Settings**」ウィンドウが表示されます。
 - a. 必要に応じ、インポートする変数設定がなされているコンポーネントが入ったフォルダに移動します。
 - b. コンポーネントのバージョンを選択します。

注-変数設定は、コンポーネントのバージョンごとに異なってもかまいません。

 - d. 「**Select**」をクリックします。
 - 5 クラスタを定義するターゲット **WebLogic Admin Server** を選択します。
 - 6 「**Run Plan (includes preflight)**」をクリックします。
 - 7 説明に従ってクラスタに含めるサーバーインスタンスを作成します。

▼ WebLogic サーバーインスタンスを管理する

存在する WebLogic 管理サーバーごとに、複数のサーバーインスタンスを実行できます。この作業では、WebLogic サーバーインスタンスの作成、起動、および停止の方法について説明します。

注 - Sun N1 Service Provisioning System の外部で WebLogic アプリケーションを管理することは避けてください。プロビジョニングソフトウェアを使用する場合、プロビジョニングソフトウェアのみを使用して WebLogic アプリケーションを管理する必要があります。

- 1 ブラウザインタフェースの「**Common Tasks**」セクションで、「**WebLogic**」をクリックします。
- 2 「**WebLogic Common Tasks**」ページで、「**Manage Server Instances**」をクリックします。
- 3 管理対象のサーバーインスタンスを作成するには、「**Default: Install in the Components Procedures list**」の隣にある「**Run action**」をクリックします。
 - a. この管理対象サーバーに適用する変数セットを選択するには、ウィンドウの「**Plan Parameters**」セクションにある「**Select from List**」をクリックします。
 - デフォルト値とは異なる値を持つ新しい変数設定セットを作成するには、「**Create Set**」をクリックします。

「**Select Variable Setting From List**」ウィンドウが表示されます。

 - i. テーブルの最上部にあるテキストフィールドに、新しい変数設定セットの名前を入力します。
 - ii. この管理対象のサーバーインスタンスが実行される **WebLogic Admin Server** の名前を指定するには、「**adminServerHostName**」の隣にあるボックスをクリックし、フィールド内に名前を入力します。これは、既存の **WebLogic Admin Server** 名でなければなりません。
 - iii. 管理対象サーバーインスタンスの名前を指定するには、「**targetName**」の隣にあるボックスをクリックし、フィールド内に名前を入力します。

この名前は、**targetRefName** 内で管理対象サーバーインスタンスの実際の名前を作成するために使用されます。
 - iv. 管理対象サーバーインスタンスのフルネームを指定するには、「**targetRefName**」の隣にあるボックスをクリックし、フィールド内に名前を入力します。

デフォルトでは、ホスト名、ホストのタイプ("server" など)、先の手順で指定した **targetName** の順に並んだものが **targetRefName** となります。

- v. 管理対象サーバーインスタンスのインストール先となるパスを指定するには、「installPath」の隣にあるボックスをクリックし、完全パス名を指定します。デフォルトでは、ホストのタイプに基づいて installPath が生成され、そのあとに adminServerHostName またはターゲットとなることのできるその他のコンポーネントのドメイン名が続き、さらに targetName が続きます。

注 - installPath 変数の値は、/opt や c:mydir などの絶対パスを指定しないかぎり、デフォルトのリモートエージェントディレクトリの相対パスとして扱われます。たとえば、Windows リモートエージェントで installPath 変数を c/mydir に設定し、デフォルトのリモートエージェントディレクトリ c:\Program Files\N1 Service Provisioning System\agent でエージェントにファイルを配備すると、ファイルは c:\Program Files\N1 Service Provisioning System\agent\c\mydir に配備されます。

- vi. この管理対象サーバーをクラスタに含める場合は、clusterHostName を指定しません。

注 - 管理対象サーバーは、このクラスタの存在に依存します。クラスタがまだ存在しない場合は、まずクラスタを作成し、そのあと管理対象サーバーインスタンスの作成に戻する必要があります。

- vii. WebLogic 起動スクリプトの完全パスを指定します。

適切なスクリプトが存在しないと、WebLogic サーバーインスタンスを起動することはできません。必要に応じて、デフォルトの WebLogic 起動スクリプトを変更してください。具体的には、次に示す変数の値を必ず指定する必要があります。

- SERVER_NAME=server_name — WebLogic 管理サーバーの名前。managed2 など。
- ADMIN_URL=URL_for_server — WebLogic サーバーの URL。
http://myplace.domain.me.com:7001/ など。
- WLS_USER=\${1} — WebLogic サーバーに使用するユーザー名。\${1} という表記は、スクリプトの呼び出し時にユーザーが提供する最初の引数を WebLogic ユーザー名に設定することを意味します。このように指定すると、サーバーの起動時にユーザー名の入力を求めるプロンプトが表示されなくなります。
- WLS_PW=\${2} — WebLogic ユーザー名のパスワード。\${2} という表記は、スクリプトの呼び出し時に2つめの引数としてユーザーが指定するパスワードを WebLogic パスワードに設定することを意味します。このように指定すると、サーバーの起動時にパスワードの入力を求めるプロンプトが表示されなくなります。

注-起動スクリプトにこれらの変更を加える場合は、スクリプト内の古い情報を必ず削除してください。古い情報が残っていると、ユーザーが定義する値は無効になります。また、スクリプトはバックグラウンドで実行されるように設定し、スクリプトの入出カストリームはリダイレクトしてください。

- viii. コンポーネント変数の値を更新したあと、「**Save**」をクリックします。
テーブル内に新しい変数設定が表示されます。
- ix. 「**Select**」をクリックします。
管理対象サーバーインスタンスの「**Components Details**」ページに戻ります。
- 別のコンポーネントの変数設定を使用するには、「**Import Set**」をクリックします。
「**Import Variable Settings**」ウィンドウが表示されます。
 - i. 必要に応じ、インポートする変数設定がなされているコンポーネントが入ったフォルダに移動します。
 - ii. コンポーネントのバージョンを選択します。

注-変数設定は、コンポーネントのバージョンごとに異なってもかまいません。

- iii. 「**Import Variable Settings**」をクリックします。
インポートされた変数設定がテーブル内に表示されます。
 - iv. 「**Select**」をクリックします。
管理対象サーバーインスタンスの「**Components Details**」ページに戻ります。
 - b. 管理対象サーバーインスタンスの「**Components Details**」ページで、この管理対象サーバーインスタンスを実行するターゲットホストを選択します。
 - c. 「**Run Plan (includes preflight)**」をクリックします。
- 4 WebLogic 管理対象サーバーインスタンスを起動するには、管理対象サーバーインスタンスの「**Components Details**」ページにある「**Component Procedures**」リストの「**Start**」の隣にある「**Run action**」をクリックします。
- a. 管理対象サーバーを選択します。
 - b. 「**Run Selected Installations**」をクリックします。
 - c. 「**Run Plan (includes preflight)**」をクリックします。

- 5 **WebLogic** 管理対象サーバーインスタンスを停止するには、管理対象サーバーインスタンスの「**Components Details**」ページにある「**Component Procedures**」リストの「**Stop**」の隣にある「**Run action**」をクリックします。
 - a. 管理対象サーバーを選択します。
 - b. 「**Run Selected Installations**」をクリックします。
 - c. 「**Run Plan (includes preflight)**」をクリックします。

WebLogic アプリケーションの取得と編集

「WebLogic Common Tasks」ページでは、WebLogic Enterprise アプリケーション、EJB™ アプリケーション、および Web アプリケーションの取得、構成、配備が可能になります。

具体的には次のような作業を実行できます。

- 30 ページの「WebLogic エンタープライズアプリケーションまたは EAR ファイルを取得する」
- 31 ページの「WebLogic Web アプリケーションまたは WAR ファイルを取得する」
- 31 ページの「WebLogic JAR ファイルを取得する」

注 - Sun N1 Service Provisioning System の外部では、WebLogic アプリケーションの配備や管理を行わないでください。プロビジョニングソフトウェアを使用する場合、プロビジョニングソフトウェアのみを使用して WebLogic アプリケーションを管理する必要があります。

▼ WebLogic エンタープライズアプリケーションまたは EAR ファイルを取得する

- 1 ブラウザインタフェースの「**Common Tasks**」セクションで、「**WebLogic**」をクリックします。
- 2 「**WebLogic Common Tasks**」ページの「**Enterprise Applications (EARs)**」セクションで、「**Create New**」をクリックします。
- 3 このコンポーネントのラベルを入力します。
- 4 このコンポーネントの説明を入力します。
- 5 このエンタープライズアプリケーションが存在する **WebLogic Admin Server** を選択します。
- 6 階層をナビゲートして、取得するエンタープライズアプリケーションを検索します。

- 7 「**Check-in Selected Item**」をクリックします。
- 8 「**check-in**」ページの情報を確認して、「**Continue to Check-in**」をクリックします。

▼ **WebLogic Web アプリケーションまたは WAR ファイルを取得する**

- 1 ブラウザインタフェースの「**Common Tasks**」セクションで、「**WebLogic**」をクリックします。
- 2 「**WebLogic Common Tasks**」ページの「**Web Applications (WARs)**」セクションで、「**Create New**」をクリックします。
- 3 このコンポーネントのラベルを入力します。
- 4 このコンポーネントの説明を入力します。
- 5 この **Web** アプリケーションが存在する **WebLogic Admin Server** を選択します。
- 6 階層をナビゲートして、取得する **Web** アプリケーションを検索します。
- 7 「**Check-in Selected Item**」をクリックします。
- 8 「**check-in**」ページの情報を確認して、「**Continue to Check-in**」をクリックします。

▼ **WebLogic JAR ファイルを取得する**

- 1 ブラウザインタフェースの「**Common Tasks**」セクションで、「**WebLogic**」をクリックします。
- 2 「**WebLogic Common Tasks**」ページの「**Java Archives Containing EJBs (JARs)**」セクションで、「**Create New**」をクリックします。
- 3 このコンポーネントのラベルを入力します。
- 4 このコンポーネントの説明を入力します。
- 5 この **JAR** ファイルが存在する **WebLogic Admin Server** を選択します。
- 6 階層をナビゲートして、取得する **JAR** ファイルを検索します。
- 7 「**Check-in Selected Item**」をクリックします。

- 8 「check-in」 ページの情報を確認して、「Continue to Check-in」をクリックします。

コンポーネントタイプ

BEA WebLogic プラグインには WebLogic に固有の複数のコンポーネントタイプが含まれています。これらのコンポーネントタイプを使用すると、最も一般的な WebLogic アプリケーションコンポーネントの多くを素早くモデル化し、特定のリソースとインストール、アンインストール、エクスポート、およびスナップショット動作を自動的に関連付けることができます。サーバーやクラスタを作成、管理する際には、コンポーネントタイプの多くがソフトウェアにより使用されます。このため、ユーザーが直接使用するコンポーネントタイプはわずかです。

- エンタープライズアプリケーション (EAR)
- Web アプリケーション (WAR)
- Enterprise JavaBeans (EJB) を含む Java Archive (JAR) ファイル

エンタープライズアプリケーション (EAR) コンポーネントタイプ

このタイプのコンポーネントは、エンタープライズアプリケーションをモデル化します。このコンポーネントは、エンタープライズアプリケーションアーカイブ (EAR) またはパッケージとして EAR の拡張版を含むことができます。

エンタープライズアプリケーションのブラウズ

次のブラウザのいずれかを使用して、このコンポーネントタイプのエンタープライズアプリケーションを選択できます。

- インストール済みアプリケーションの 1 つとその関連設定を選択できる管理サーバーブラウザ。管理サーバーブラウザは、管理サーバー上のエンタープライズアプリケーションをアルファベット順に並べたリストを表示します。あるコンポーネント用に 1 つのエンタープライズアプリケーションを選択できます。
- EAR ファイルを選択して、設定が含まれないコンポーネントを作成できる、ファイルシステムブラウザ。ファイルシステムブラウザは、ユーザーが選択できるファイルとディレクトリのリストを提供します。これらのファイルまたはタイプ *.ear のみをブラウザに表示させることを選択できます。

エンタープライズアプリケーションのインストール

エンタープライズアプリケーションのコンポーネントをインストールするには、WebLogic サーバーまたはクラスタをターゲットホストとして使用する必要があります。インストールプロセスにより、次の 3 つの主要タスクが実行されます。

1. ターゲットへの EAR ファイルのインストール。

2. WebLogic 管理サーバーへの EAR の登録。
3. ドメインのブラウザで以前に取得されたプロパティの適用。

エンタープライズアプリケーションのアンインストール

エンタープライズアプリケーションのコンポーネントをアンインストールすると、そのエンタープライズアプリケーションはターゲットから外れます。エンタープライズアプリケーションをターゲットにしているサーバーまたはクラスタが存在しない場合には、管理サーバーから登録コンポーネントが削除されます。

エラー状態

表 4-1 WebLogic エンタープライズアプリケーションのエラー状況

アクション	条件	結果
インストール	トポロジが正しく構成されていない (ターゲットホストが正しいドメインホストをポイントしていない)	ターゲット化が失敗する
インストール	ターゲットホストが有効な WebLogic ターゲットではない	インストールが禁止される
ブラウザ/インストール/アンインストール	資格が適切に構成されていない。	操作が失敗する
ブラウザ	ドメインホストでパスが正しく構成されていない。	ブラウザが失敗する

Web アプリケーション (WAR) コンポーネントタイプ

このタイプのコンポーネントには Web アプリケーションが含まれます。コンポーネントには、Web アプリケーションとその関連設定を含めることも、あるいは設定は含めずに Web アプリケーションだけを含めることもできます。

Web アプリケーションのブラウザ

次のブラウザのいずれかを使用して、このコンポーネントタイプの Web アプリケーションを選択できます。

- インストール済みアプリケーションの 1 つとその関連設定を選択できる管理サーバーブラウザ。管理サーバーブラウザは、管理サーバー上の Web アプリケーションをアルファベット順に並べたリストを表示します。1 つのコンポーネントに対して 1 つの Web アプリケーションを選択できます。

- WAR ファイルを選択して、設定が含まれないコンポーネントを作成できる、ファイルシステムブラウザ。ファイルシステムブラウザは、ユーザーが選択できるファイルとディレクトリのリストを提供します。これらのファイルまたはタイプ*.warのみをブラウザに表示させることを選択できます。

Web アプリケーションのインストール

Web アプリケーションのコンポーネントをインストールするには、WebLogic サーバーまたはクラスタをターゲットホストとして使用する必要があります。このインストールでは、コンポーネントがこのターゲットにインストールされ、含まれている登録コンポーネントがターゲットの管理サーバーにインストールされます。

Web アプリケーションのアンインストール

Web アプリケーションのコンポーネントをアンインストールすると、その Web アプリケーションはターゲットからターゲット解除されます。Web アプリケーションをターゲットにしているサーバーまたはクラスタが存在しない場合には、管理サーバーから登録コンポーネントが削除されます。

エラー状態

表 4-2 WebLogic Web アプリケーションのエラー状況

アクション	条件	結果
インストール	トポロジが正しく構成されていない(ターゲットホストが正しいドメインホストをポイントしていない)	ターゲット化が失敗する。
インストール	ターゲットホストが有効な WebLogic ターゲットではない。	インストールが禁止される。
ブラウズ/インストール/アンインストール	資格が適切に構成されていない。	操作が失敗する。
ブラウズ	ドメインホストでパスが正しく構成されていない。	ブラウズが失敗する。

EJB コンポーネントタイプを含む Java アーカイブファイル

ブラウズ

次のブラウザのいずれかを使用して、このコンポーネントタイプの JAR または EJB ファイルを選択できます。

- インストール済みアプリケーションの1つとその関連設定を選択できる管理サーバーブラウザ。管理サーバーブラウザは、管理サーバー上のアプリケーションをアルファベット順に並べたリストを表示します。1つのコンポーネントに対して1つのEJBを選択できます。
- JARファイルを選択して、設定が含まれないコンポーネントを作成できる、ファイルシステムブラウザ。ファイルシステムブラウザは、ユーザーが選択できるファイルとディレクトリのリストを提供します。これらのファイルまたはタイプ*.jarのみをブラウザに表示させることを選択できます。

インストール

このタイプのコンポーネントをインストールする場合は、ファイルまたはディレクトリはインストールパスに基づいてファイルシステムにコピーされます。コピーが終了すると、そのファイルまたはディレクトリは WebLogic 管理サーバーに登録されます。

アンインストール

このタイプのコンポーネントをアンインストールすると、マスターサーバーからファイルまたはディレクトリが削除されます。

エラー状態

表 4-3 WebLogic EJB コンテナのエラー状態

アクション	条件	結果
アンインストール	依存する WebLogic EJB が依然としてインストールされている。	依存関係を示し、アンインストールが失敗する。

プラン

BEA WebLogic プラグインには、定義済みプランは含まれていません。

リソース

BEA WebLogic プラグインには、追加リソースは含まれていません。

索引

A

- Admin Server
 - WebLogic 6.1, 23-25
 - WebLogic 7.0, 22-23

E

- EAR
 - コンポーネントタイプ, 32-33
 - 取得, 30-31
- EJB
 - コンポーネントタイプ, 34-35
 - 取得, 31-32
- EJBの取得, 31-32

J

- JAR
 - コンポーネントタイプ, 34-35
 - 取得, 31-32
- JAR ファイル, 17
- JAR ファイルの取得, 31-32
- Java アーカイブ, 「JAR ファイル」を参照

W

- WAR
 - コンポーネントタイプ, 33-34
 - 取得, 31

WebLogic 6.1

- Admin Server, 23-25
- 管理対象サーバーインスタンス, 27-30
- クラスタ, 25-26

WebLogic 7.0

- Admin Server, 22-23
- 管理対象サーバーインスタンス, 27-30
- クラスタ, 25-26

WebLogic プラグイン

- インポート, 19-20
- 構成, 20
- 構成要素, 11-12
- 配備サーバーの要件, 12-13

Web アプリケーション

- コンポーネントタイプ, 33-34
- 取得, 31

Web アプリケーションの取得, 31

え

- エンタープライズアプリケーション
 - コンポーネントタイプ, 32-33
 - 取得, 30-31
- エンタープライズアプリケーションの取得, 30-31

か

- 管理対象サーバーインスタンス, 27-30

く

クラスタ, 25-26

こ

コンポーネントタイプ

EAR, 32-33

EJB, 34-35

JAR, 34-35

WAR, 33-34

Web アプリケーション, 33-34

エンタープライズアプリケーション, 32-33

は

配備サーバーの要件, 12-13

ふ

プラグインのインポート, 19-20

プラグインの構成, 20

プラグインファイル

インポート, 19-20

場所, 17