



BEA WebLogic ServerTM

パートナーズガイド

BEA WebLogic Server バージョン 6.1
マニュアルの日付：2002年6月24日

著作権

Copyright © 2002 BEA Systems, Inc. All Rights Reserved.

限定的権利条項

本ソフトウェアおよびマニュアルは、BEA Systems, Inc. 又は日本ビー・イー・エー・システムズ株式会社（以下、「BEA」といいます）の使用許諾契約に基づいて提供され、その内容に同意する場合にのみ使用することができ、同契約の条項通りにのみ使用またはコピーすることができます。同契約で明示的に許可されている以外の方法で同ソフトウェアをコピーすることは法律に違反します。このマニュアルの一部または全部を、BEA からの書面による事前の同意なしに、複写、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

米国政府による使用、複製もしくは開示は、BEA の使用許諾契約、および FAR 52.227-19 の「Commercial Computer Software-Restricted Rights」条項のサブパラグラフ (c)(1)、DFARS 252.227-7013 の「Rights in Technical Data and Computer Software」条項のサブパラグラフ (c)(1)(ii)、NASA FAR 補遺 16-52.227-86 の「Commercial Computer Software--Licensing」条項のサブパラグラフ (d)、もしくはそれらと同等の条項で定める制限の対象となります。

このマニュアルに記載されている内容は予告なく変更されることがあり、また BEA による責務を意味するものではありません。本ソフトウェアおよびマニュアルは「現状のまま」提供され、商品性や特定用途への適合性を始めとする（ただし、これらには限定されない）いかなる種類の保証も与えません。さらに、BEA は、正当性、正確さ、信頼性などについて、本ソフトウェアまたはマニュアルの使用もしくは使用結果に関していかなる確約、保証、あるいは表明も行いません。

商標または登録商標

BEA、WebLogic、Tuxedo、および Jolt は BEA Systems, Inc. の登録商標です。How Business Becomes E-Business、BEA WebLogic E-Business Platform、BEA Builder、BEA Manager、BEA eLink、BEA WebLogic Commerce Server、BEA WebLogic Personalization Server、BEA WebLogic Process Integrator、BEA WebLogic Collaborate、BEA WebLogic Enterprise、および BEA WebLogic Server は、BEA Systems, Inc. の商標です。

その他の商標はすべて、関係各社がその権利を有します。

BEA WebLogic Server パートナース ガイド

パート番号	マニュアルの日付	ソフトウェアのバージョン
なし	2002 年 6 月 24 日	WebLogic Server バージョン 6.1 サービス パック 3

目次

このマニュアルの内容

対象読者	v
e-docs Web サイト	vi
このマニュアルの印刷方法	vi
関連情報	vi
サポート情報	vii
表記規則	viii

1. WebLogic Server の配布

WebLogic Server の新しいリリースを配布するための ISV ライセンスのアップグレード	1-2
BEA Star Partner Program への登録	1-2
パートナー開発キットのインストール	1-3
ISV ライセンスのインストール	1-3
手順 1: ISV ライセンスをインストールする準備	1-4
手順 2: ライセンス データの抽出と WebLogic Server ファイルへのリンク	1-5
手順 3: WebLogic Server ライセンスの更新	1-6
次の手順: アプリケーションと WebLogic Server のコンフィグレーション	1-6
ファイルの配布	1-7

2. カスタム WebLogic Server インストールの作成

概要	2-1
config.xml ファイルのカスタマイズ	2-2
アプリケーション リソースのプリコンフィグレーション	2-3
アプリケーション コンポーネントのデプロイメント	2-4
fileRealm.properties ファイルのカスタマイズ	2-5
コンフィグレーションの例	2-6
ドメインのコンフィグレーション	2-6
基本的なサーバの設定	2-7

アプリケーション JDBC 要件	2-8
セキュリティ レルムの定義	2-10
アプリケーション コンポーネント	2-10
スタートアップ クラス	2-11
fileRealm.properties の例	2-12

3. WebLogic Server MBean の使用

概要	3-1
JDBC プロファイリング MBean の使用	3-2
JDBC プロファイリングの有効化	3-2
JDBC プロファイルのアクセス	3-3

このマニュアルの内容

このマニュアルでは、独立系ソフトウェアベンダ（ISV）用のライセンスを入手およびインストールする方法について説明します。ISV ライセンスを入手すると、開発したアプリケーションに BEA のコア技術をバンドルし、両方をまとめて単独の製品として配布できるようになります。このマニュアルでは、BEA WebLogic Server™ をアプリケーションとバンドルするための開発手法についても説明します。

このマニュアルの構成は次のとおりです。

- 第 1 章「WebLogic Server の配布」では、ISV ライセンスの入手方法と、配布キットに収めなければならない WebLogic Server ファイルについて説明します。
- 第 2 章「カスタム WebLogic Server インストールの作成」では、アプリケーションとともに配布する WebLogic Server コンフィグレーション ファイルに対して、パートナーおよび ISV が行う一般的な変更について説明します。
- 第 3 章「WebLogic Server MBean の使用」では、JDBC プロファイリングを有効にし、使用する方法について説明します。

対象読者

このマニュアルは、BEA WebLogic Server のコアテクノロジーを使用するカスタムアプリケーションの作成に関心がある独立系ソフトウェアベンダ（ISV）とその他の開発者を対象としています。BEA WebLogic Server プラットフォーム、WebLogic Server ドキュメントセット内のその他のガイドの内容、および Java プログラミング言語に読者が精通していることを前提として書かれています。

e-docs Web サイト

BEA 製品のドキュメントは、BEA の Web サイトで入手できます。BEA ホームページの [製品のドキュメント] をクリックするか、WebLogic Server 製品ドキュメント ページ (<http://edocs.beasys.co.jp/e-docs/index.html>) を直接表示してください。

このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルのメイン トピックを一度に 1 つずつ印刷できます。

このマニュアルの PDF 版は、Web サイトで入手できます。PDF を Adobe Acrobat Reader で開くと、マニュアルの全体 (または一部分) を書籍の形式で印刷できます。PDF を表示するには、WebLogic Server ドキュメントのホームページを開き、[ドキュメントのダウンロード] をクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader は、Adobe の Web サイト (<http://www.adobe.co.jp>) から無料で入手できます。

関連情報

BEA の Web サイトでは、WebLogic Server の全マニュアルを提供しています。以下の BEA WebLogic Server のマニュアルには、WebLogic Server を拡張する方法を理解するための関連情報が含まれています。

- BEA WebLogic Server のマニュアル (オンラインで入手可能)
 - [管理者ガイド](#)
 - [プログラミング ガイド](#)
 - [WebLogic Server API](#)

-
- Sun Microsystems, Inc. の Java サイト (<http://java.sun.com/>)

BEA WebLogic Server と Java の詳細については、<http://edocs.beasys.co.jp/e-docs/>の参考文献を参照してください。

サポート情報

BEA のドキュメントに関するユーザからのフィードバックは弊社にとって非常に重要です。質問や意見などがあれば、電子メールで docsupport-jp@bea.com までお送りください。寄せられた意見については、ドキュメントを作成および改訂する BEA の専門の担当者が直に目を通します。

電子メールのメッセージには、ご使用のソフトウェア名とバージョン名、およびマニュアルのタイトルと作成日付をお書き添えください。本バージョンの BEA WebLogic Server について不明な点がある場合、または BEA WebLogic Server のインストールおよび動作に問題がある場合は、BEA WebSUPPORT (www.bea.com) を通じて BEA カスタマ サポートまでお問い合わせください。カスタマ サポートへの連絡方法については、製品パッケージに同梱されているカスタマ サポート カードにも記載されています。

カスタマ サポートでは以下の情報をお尋ねしますので、お問い合わせの際はあらかじめご用意ください。

- お名前、電子メール アドレス、電話番号、ファクス番号
- 会社の名前と住所
- お使いの機種とコード番号
- 製品の名前とバージョン
- 問題の状況と表示されるエラー メッセージの内容

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
{ Ctrl } + { Tab }	同時に押すキーを示す。
斜体	強調または本のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、Java クラス、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 例： <pre>import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float</pre>
斜体の等幅テキスト	コード内の変数を示す。 例： <pre>String CustomerName;</pre>
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 例： <pre>LPT1 BEA_HOME OR</pre>
{ }	構文内の複数の選択肢を示す。

表記法	適用
[]	<p>構文内の任意指定の項目を示す。</p> <p>例：</p> <pre>java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]</pre>
	<p>構文の中で相互に排他的な選択肢を区切る。</p> <p>例：</p> <pre>java weblogic.deploy [list deploy undeploy update] password {application} {source}</pre>
...	<p>コマンドラインで以下のいずれかを示す。</p> <ul style="list-style-type: none"> ■ 引数を複数回繰り返すことができる。 ■ 任意指定の引数が省略されている。 ■ パラメータや値などの情報を追加入力できる。
.	<p>コード サンプルまたは構文で項目が省略されていることを示す。</p> <p>.</p> <p>.</p> <p>.</p>



1 WebLogic Server の配布

開発したアプリケーションと BEA のコア技術の両方をまとめて、単独の製品として配布することができます。これにより、アプリケーションと J2EE アプリケーション サーバの両方を購入、インストール、および保守する必要がなくなり、顧客側での労力が軽減されます。このしくみは WebLogic Server のプラグアンドプレイ環境と組み合わせることにより、開発した製品との統合を行うための理想的な選択肢となります。

WebLogic Server を配布するには、**ISV ライセンス**と呼ばれる特殊なライセンスを入手し、インストールする必要があります（アプリケーションの開発または WebLogic Server のコンフィグレーションには、ISV ライセンスは必要ありません）。WebLogic Server の ISV ライセンスをインストールすると、サーバが変更され、ファイルが強固にリンクされます。配布キットには、変更されたこれらの WebLogic Server ファイルを収める必要があります。

配布可能な WebLogic Server をセットアップするには、以下の作業を行います。

- WebLogic Server の新しいリリースを配布するための ISV ライセンスのアップグレード
- BEA Star Partner Program への登録
- パートナ開発キットのインストール
- ISV ライセンスのインストール
- ファイルの配布

WebLogic Server の新しいリリースを配布するための ISV ライセンスのアップグレード

WebLogic Server の古いバージョンを配布してある場合は、以下の手順をすべて実行し、既存の ISV ライセンスをアップグレードする必要があります。

1. `licensing@bea.com` に、新しい ISV ライセンスを要求する電子メールを送信します。メールには、古いライセンスが入っている `isv.jar` ファイルを添付します。WebLogic Server リリース 5.1 からアップグレードする場合は、古いライセンス ファイルを添付します。
2. WebLogic Server 6.1 のパートナー開発キット (Partner Development Kit) をインストールします。
3. BEA から新しい `isv.jar` ファイルを受け取ったなら、1-3 ページの「ISV ライセンスのインストール」の説明に従って新しい ISV ライセンスをインストールします。

インストーラの内容を、新しい WebLogic Server ファイルに更新します。

BEA Star Partner Program への登録

BEA Systems, Inc. では Star Partner Program を通じて、独立系ソフトウェアベンダ (ISV) およびアプリケーション ソフトウェア プロバイダ (ASP) との協力関係を結んでいます。BEA Star Partner Program にまだ登録していない場合は、以下の方法で登録してください。

1. BEA WebLogic Server の動作確認状況のページ (<http://edocs.beasys.co.jp/e-docs/wls/certifications/certifications/index.html>) を参照して、対象プラットフォームで WebLogic Server が動作することが保証されていることを確認します。

2. BEA Star Partner Program の Web サイトにアクセスし、プログラムと登録について理解してください。Web サイトには、次の URL を使ってアクセスできます。
<http://www.bea.com/partners>

パートナー開発キットのインストール

プログラムに登録すると、BEA の主要製品がすべて収録された CD コレクションが BEA より発送されます。パートナー開発キット (Partner Development Kit) が到着したら、CD からソフトウェアをインストールします。WebLogic Server のインストールの詳細については、BEA ドキュメント Web サイト (<http://edocs.beasys.co.jp/e-docs/index.html>) 上の『インストールガイド』を参照してください。

警告： 配布可能な WebLogic Server のホストとして使用するコンピュータ上に BEA 製品が既にインストールされている場合、パートナー開発キットをインストールする前に、その時点の `BEA_HOME\license.bea` ファイルをバックアップします。BEA ホームディレクトリと `license.bea` ファイルの詳細については、『インストールガイド』の「BEA ホームディレクトリ」を参照してください。

CD が到着する前に、BEA システムズ Evaluation Center (<http://www.beasys.co.jp/evaluation/index.html>) から BEA のソフトウェアをダウンロードすることもできます。アクティブな WebSUPPORT アカウントをお持ちの場合、WebSUPPORT ログインパスワードを使用してソフトウェアをダウンロードできます。

ISV ライセンスのインストール

Star Partner Program への登録が完了すると、カスタマイズ済みの ISV ライセンスが、電子メールの添付ファイル (`isv.jar`) の形態で BEA より送付されます。ここで説明する ISV ライセンス ファイルのインストール方法は、**WebLogic**

Server のバージョン 6.1 に対してだけ使用できます。WebLogic Server の他のバージョンに ISV ライセンスをインストールする場合は、お使いのソフトウェアバージョンに対するインストール マニュアルを参照してください。

ISV ライセンスのインストールは、3 つの主要な手順を経て行います。

- 手順 1: ISV ライセンスをインストールする準備
- 手順 2: ライセンス データの抽出と WebLogic Server ファイルへのリンク
- 手順 3: WebLogic Server ライセンスの更新

手順 1: ISV ライセンスをインストールする準備

ISV ライセンス ファイルをインストールする前に、次の作業を行います。

1. まだインストールしていない場合、1-3 ページの「パートナー開発キットのインストール」の説明に従って、WebLogic Server をインストールします。

BEA インストーラが使用する BEA ホーム ディレクトリの場所に注意してください。BEA ホーム ディレクトリには、このプロセスの以後の手順で更新される `license.bea` ファイルが置かれます。BEA ホーム ディレクトリと `license.bea` ファイルの詳細については、『インストール ガイド』の「[BEA ホーム ディレクトリ](#)」を参照してください。

2. 電子メールで送付された `isv.jar` ファイルを、インストーラで使用した BEA ホーム ディレクトリにコピーします。
3. コマンド シェルを起動し、`BEA_HOME` ディレクトリに移動します。`BEA_HOME` は BEA ホーム ディレクトリの名前です。
4. 次のコマンドを入力して、コンピュータの `CLASSPATH` 環境変数に `isv.jar` へのパスを追加します。

```
set CLASSPATH=.\isv.jar;%CLASSPATH% (Windows システムの場合)
```

```
export CLASSPATH=./isv.jar:$CLASSPATH (UNIX システムの場合)
```

5. 次のいずれかのコマンドを入力して、コンピュータの `PATH` 環境変数に WebLogic Server JDK へのパスを追加します。

- `set PATH=.\jdk131\bin;%PATH%` (Windows システムの場合)

- `export PATH=./jdk131/bin:$PATH` (UNIX システムの場合)

これで、ISV ライセンス データを抽出して WebLogic Server ファイルにリンクする準備ができました。

手順 2: ライセンス データの抽出と WebLogic Server ファイルへのリンク

ISV ライセンス データを抽出して WebLogic Server ファイルにリンクするには、`BEA_HOME` ディレクトリに移動して次のコマンドを入力します。

```
java -Xmx128m -Dbea.home=BEA_HOME  
-Dbea.jar=WL_HOME\lib\weblogic.jar install
```

`BEA_HOME` は BEA ホーム ディレクトリの絶対パス名であり、`WL_HOME` は WebLogic Server をインストールしたディレクトリの絶対パス名です。

警告： いったん開始した後でこのプロセスを中断しないでください。

コマンドを実行すると、ISV ライセンス データを格納した、`BEA_HOME\license_isv.bea` という名前のファイルが生成されます。また、`WL_HOME` ディレクトリ内部のファイルが ISV ライセンスにリンクされます。これにより、`WL_HOME` ディレクトリ内の指定されたファイルだけが、`license_isv.bea` に抽出された ISV ライセンス データを使用できるようになります。

注意： 一部のプラットフォームと JDK では、「メモリ不足エラー」が発生する場合があります。このエラーに対処するには、(最大ヒープサイズを MB 単位で設定する) `-Xmx` 引数の値を大きくして、コマンドをもう一度実行します。たとえば、デフォルトのヒープサイズを 150MB に増やすには、`-Xmx150m` と指定します。

ISV ライセンスのインストール プロセスを完了するには、`license_isv.bea` 内のデータを使用して WebLogic Server ライセンスを更新する必要があります。

手順 3: WebLogic Server ライセンスの更新

新しく生成された `license_isv.bea` ファイルを使用して `license.bea` ファイルを更新するには、`BEA_HOME` ディレクトリに移動して次のいずれかのコマンドを入力します。

- `UpdateLicense license_isv.bea` (Windows システムの場合)
- `sh UpdateLicense.sh license_isv.bea` (UNIX システムの場合)

`UpdateLicense` コマンドを実行すると、`license_isv.bea` ファイルが `license.bea` ファイルとマージされます。`UpdateLicense` を実行した後は、`license_isv.bea` ファイルを削除してもかまいません。

次の手順 : アプリケーションと WebLogic Server のコンフィグレーション

ISV ライセンスのインストールが終了したら、ISV ライセンスとリンクされた WebLogic Server を起動し、アプリケーションをデプロイし、サーバ コンポーネントをコンフィグレーションします。詳細については、<http://edocs.beasys.co.jp/e-docs/index.html> にある以下のドキュメントを参照してください。

- 『管理者ガイド』の「[WebLogic Server の起動と停止](#)」
- 『[Administration Console オンライン ヘルプ](#)』
- 『[WebLogic JMX Service プログラマーズ ガイド](#)』。WebLogic Server 管理システムを使用した作業の詳細とコード サンプルを示しています。
- このマニュアルの残りの節。ISV に固有の開発上のヒントを示しています。

ファイルの配布

アプリケーションとともに WebLogic Server を配布する準備ができたなら、1-6 ページの「手順 3: WebLogic Server ライセンスの更新」で指定した BEA ライセンス ファイル (BEA_HOME\license.bea) と WL_HOME\lib\weblogic.jar ファイルがインストーラに収められていることを確認する必要があります。

指定したファイルを両方ともインストールしないと、アプリケーションに組み込まれた WebLogic Server は起動しません。

この license.bea と weblogic.jar の同じ組み合わせは、ライセンスにリンクされるすべてのインストールで使用できます。WebLogic Server のサイレントインストール プロセスの使用方法については、「[サイレントインストールによる WebLogic Server のインストール](#)」を参照してください。

2 カスタム WebLogic Server インストールの作成

この章では、WebLogic Server のカスタマイズバージョンをインストールして独自のアプリケーションで使用するパートナーおよび ISV を対象に、WebLogic Server コンフィグレーションに関するトピックについて説明します。

- [概要](#)
- [config.xml ファイルのカスタマイズ](#)
- [fileRealm.properties ファイルのカスタマイズ](#)
- [コンフィグレーションの例](#)

概要

WebLogic Server の `config.xml` ファイルには、特定の WebLogic Server 管理ドメインのコンフィグレーション情報が格納されています。このファイルは、特定のドメインのすべての詳細情報を制御します。このような情報としては、サーバとクラスタの名前、数、およびコンフィグレーション、デプロイ可能なリソースおよびアプリケーションのリスト、デプロイ可能なリソースおよびアプリケーションのサーバおよびクラスタへのマッピングなどがあります。

以降の節では、カスタマイズした WebLogic Server 管理ドメインをインストールする際に、パートナーが `config.xml` に対して行う一般的な変更事項について説明します。`config.xml` ファイルまたは管理ドメインの役割に精通していない場合は、『[管理者ガイド](#)』の「[WebLogic Server 管理の概要](#)」および「[WebLogic Server とクラスタのコンフィグレーション](#)」を参照してください。

通常的环境では、`config.xml` を手動で編集しないことをお勧めします。ただし、パートナーがインストールをカスタマイズするために、このファイルを直接編集することが必要な場合もあります。`config.xml` を直接編集することに慣れて

いない場合は、『[BEA WebLogic Server コンフィグレーション リファレンス](#)』を参照してください。このリファレンス マニュアルには、`config.xml` を編集する際の規則や、このファイルの DTD の詳細な説明が記載されています。

config.xml ファイルのカスタマイズ

サイレント インストールのプロパティ ファイルは WebLogic Server の初期の場所、パスワード、およびファイルセットを定義し、`config.xml` ファイルは管理ドメインにあるすべての WebLogic Server のコンフィグレーション設定の大部分を定義します。コンソールや `weblogic.Admin` ユーティリティなどの WebLogic Server ツールは、`config.xml` ファイルの要素を編集することによって、ドメインのコンフィグレーションを管理します。

パートナーおよび ISV は、カスタム `config.xml` ファイルを使用して、それぞれの Web アプリケーションをサポートするデフォルトの WebLogic Server ドメイン コンフィグレーションを作成できます。通常は、次を行うために `config.xml` ファイルを作成、修正、または生成します。

- アプリケーションが必要とする WebLogic Server リソースをプリコンフィグレーションする。
- パートナ アプリケーションのコンポーネントを WebLogic Server に自動的にデプロイする。

次の節では、パートナーがインストール用に修正する可能性のある `config.xml` ファイルの要素について説明します。

アプリケーション リソースのプリコンフィグレーション

パートナー アプリケーションは通常いくつかの WebLogic Server リソースに依存し、それらの各リソースは config.xml ファイルで定義されます。

リソース	config.xml 要素	メモ
ドメイン	Domain	凝集された単位として動作するために、アプリケーションのコンポーネントをホストするすべての WebLogic Server は、単一の WebLogic Server 管理ドメイン内に置かれる必要がある。ドメイン名の選択も、アプリケーション コンポーネントのインストール場所および config.xml ファイル自身に影響を与える。config.xml、fileRealm.properties、スタートアップクラス、および可能性としてアプリケーション コンポーネントが、ドメインと同じ名前のサブディレクトリ内に含まれる。
サーバ名および接続情報	Server	最も基本的なレベルでは、1 つまたは複数の WebLogic Server 名、IP アドレス、およびポート番号にアクセスするためにパートナー アプリケーションをコンフィグレーションできる。アプリケーションで必要であれば、WebLogic Server ドメインをハードコード化して、特定のサーバ名および接続ポートを使用できる。 コンフィグレーションをインストールする前に、IP アドレスをアプリケーション インストーラによって動的にコンフィグレーションし、config.xml に埋め込むことができる。

リソース	config.xml 要素	メモ
JDBC データソース	JDBCConnectionPool JDBCDataSource JDBCMultiPool JDBCTxDataSource	WebLogic Server をインストールするパートナー アプリケーションは、アプリケーション データを管理するために RDBMS や他のデータストアをインストールする場合があります。使用中の製品のインストーラがアプリケーションと一緒にデータストアをインストールする場合、インストールされた WebLogic Server をプリコンフィグレーションして、そのデータストアのデフォルトのデータソースおよび接続プールを設定できる。

アプリケーション コンポーネントのデプロイメント

パートナー アプリケーションは、`config.xml` に必要な要素を追加することによってインストールすることもできます。ただし、プリコンフィグレーションした WebLogic Server にアプリケーションをインストールする場合は、`config.xml` 設定とアプリケーション コンポーネント ファイル (`.war`、`.jar`、`.html` など) のインストール場所とを調整する必要があります。

次の表は、WebLogic Server 内にアプリケーション コンポーネントを事前にデプロイするために使用する要素についてまとめたものです。これらの要素がどのように実際のアプリケーション コンポーネント ファイルのインストール場所に対応するかを示す例については、後述の [コンフィグレーションの例](#) を参照してください。

コンポーネント	config.xml 要素	メモ
スタートアップ クラス	StartupClass	WebLogic Server スタートアップ クラスは、パートナー アプリケーションの他のコンポーネントに必要なリソースを初期化するために使用できる。
Web サーバ	WebServer	Web アプリケーションには、通常、ビジネス ロジックに加え、静的な <code>.html</code> コンテンツなどの標準 Web リソースが必要である。アプリケーションの静的ファイルのデフォルトの場所をコンフィグレーションするには、 <code>config.xml</code> ファイルを使用する。

コンポーネント	config.xml 要素	メモ
Web アプリケーション	Application	EAR および WAR ファイルは、アプリケーションディレクトリまたは WebLogic Server ディレクトリの任意の場所に格納できる。起動時にアプリケーションをデプロイするには、config.xml 内から最終のインストール場所を参照する。

fileRealm.properties ファイルのカスタマイズ

fileRealm.properties は、デフォルト WebLogic Server ファイル レルムの ACL、グループ、およびセキュリティ プリンシパルを定義します。アプリケーションのセキュリティにサードパーティのセキュリティ レルムが必要でない場合は、fileRealm.properties を使用して必要なすべてのセキュリティ構成をプリコンフィグレーションできます。

アプリケーションをサードパーティのセキュリティ レルム（たとえば Windows NT セキュリティ レルムを使用したシングル サインオン）と統合する必要がある場合、サードパーティのセキュリティ実装で使用するために、fileRealm.properties ファイルとキャッシング レルムの両方をコンフィグレーションする必要があります。この場合、fileRealm.properties コンフィグレーションはバックアップ レルムとして使用されます。このタイプのセキュリティ実装には、fileRealm.properties と config.xml の両方にプリコンフィグレーションが必要です。

次のコンフィグレーションの例では、カスタマイズされた fileRealm.properties ファイルの簡単な例を示します。詳細については、「[WebLogic Server セキュリティのプログラミング](#)」を参照してください。

コンフィグレーションの例

WebLogic Server と一緒にインストールされた Sun の Java Pet Store アプリケーションは、Web アプリケーションをサポートするためにインストールをプリコンフィグレーションする方法の簡単な例を提供しています。WebLogic Server をインストールした後、関連付けられた `petstore` ドメインでサーバを起動することによって、簡単に Pet Store アプリケーションを使用できます。

この節では、Pet Store アプリケーションの主要な側面について紹介し、それらの側面が WebLogic Server コンフィグレーション ファイルを使用してプリコンフィグレーションされている個所を説明します。

ドメインのコンフィグレーション

`config.xml` ファイルは、`petstore` ドメインの新しいドメイン コンフィグレーションを使用して開かれます。アプリケーションのすべてのサーバ、リソース、およびコンポーネントは、このドメイン内にデプロイされます。

```
<Domain  
  Name="petstore"  
>
```

ドメインの選択は、プリコンフィグレーションされた `config.xml` および `fileRealm.properties` ファイルのインストール場所にも影響を与えます。つまり、WebLogic Server は、サーバインストール ディレクトリの `\config\domain_name` サブディレクトリでドメインのコンフィグレーション ファイルを探します。アプリケーション コンポーネントは、WebLogic Server ディレクトリ以外にインストールして、`config.xml` 内から参照することができます。

基本的なサーバの設定

Pet Store アプリケーションは、petstoreServer という 1 つのサーバを利用します。このサーバの接続プロパティは、次のように Server 要素属性でプリコンフィグレーションされています。

```
<Server
  JavaCompiler="C:\bea\jdk131/bin/javac"
  ListenPort="7001"
  Name="petstoreServer"
  RootDirectory="C:\bea\wlserver6.1"
  ThreadPoolSize="15"
  TransactionLogFilePrefix="config/petstore/logs/"
  IIOPEnabled="false"
>
```

サーバ設定の残りの部分では、Pet Store アプリケーションのデフォルト Web サーバおよびセキュリティ コンフィグレーションをコンフィグレーションします。たとえば、Pet Store はサーバで使用する「tour」というデフォルト Web アプリケーション名を定義します。

```
<WebServer
  DefaultWebApp="tour"
  LogFileName="./config/petstore/logs/access.log"
  LoggingEnabled="true"
  Name="petstoreServer"
/>
```

Pet Store アプリケーションは SSL を利用できるため、インストールされた config.xml ファイルはサーバの SSL を可能にし、証明書とログ ファイルのデフォルトの場所を指定します。

```
<SSL
  CertificateCacheSize="3"
```

2 カスタム WebLogic Server インストールの作成

```
Enabled="true"

ListenPort="7002"

ServerCertificateChainFileName="./config/petstore/ca.pem"

ServerCertificateFileName="./config/petstore/democert.pem"

ServerKeyFileName="./config/petstore/demokey.pem"

TrustedCAFileName="./config/petstore/ca.pem"

Ciphersuites="SSL_RSA_EXPORT_WITH_RC4_40_MD5,SSL_RSA_WITH_DES_CBC_SHA,SSL_RSA_EXPORT_WITH_DES_40_CBC_SHA,SSL_NULL_WITH_NULL_NULL"

/>

<Log

  FileName="./config/petstore/logs/weblogic.log"

/>

</Server>
```

アプリケーション JDBC 要件

Pet Store アプリケーションは、1 つの JDBC プールを使用して 3 つの異なるデータソースにアクセスします。インストールされた `config.xml` ファイルはそのプールを定義し、すべてのデータソースの対象として `petstoreServer` というデフォルト Pet Store サーバを使用します。Pet Store アプリケーションは接続プールのデフォルト ユーザ名とパスワードを指定しませんが、アプリケーション インストーラはデフォルト ユーザ名とパスワードを要求し、必要に応じて `config.xml` 要素を設定できます。

```
<JDBCDataSource

  JNDIName="jdbc.EstoreDB"

  Name="EstoreDB"

  PoolName="petstorePool"

  Targets="petstoreServer"

/>
```

```

<JDBCDataSource
    JNDIName="jdbc.InventoryDB"
    Name="InventoryDB"
    PoolName="petstorePool"
    Targets="petstoreServer"
/>

<JDBCDataSource
    JNDIName="jdbc.SignOnDB"
    Name="SignOnDB"
    PoolName="petstorePool"
    Targets="petstoreServer"
/>

...

<JDBCConnectionPool
    CapacityIncrement="1"
    DriverName="COM.cloudscape.core.JDBCdriver"
    InitialCapacity="1"
    MaxCapacity="1"
    Name="petstorePool"
    Properties="user=none;password=none;server=none"
    Targets="petstoreServer"
    URL="jdbc:cloudscape:petStore"
/>

```

セキュリティ レalmの定義

Pet Store アプリケーションはデフォルト WebLogic Server ファイル レalmを使用するため、インストールされた `config.xml` ファイルは単純にデフォルト レalm名を参照します。実際の ACL、グループなどは、`fileRealm.properties` に定義されています。

```
<FileRealm
  Name="myFileRealm"
/>

<Security
  Realm="myRealm"
/>

<Realm
  FileRealm="myFileRealm"
  Name="myRealm"
/>
```

アプリケーション コンポーネント

Pet Store アプリケーションは、`tour.war`、`petstore.ear`、および `petstoreAdmin.ear` という3つのアプリケーション コンポーネント ファイルをインストールします。これらのコンポーネントはそれぞれ WebLogic Server サブディレクトリにインストールされ、`petstoreServer` にデプロイされます。たとえば、`tour.war` は次の要素を使用してデプロイされます。

```
<Application
  Name="tour"
  Path="C:\bea\wlserver6.1/config/petstore/applications/tour.war">
  <WebAppComponent
```

```

Name="tour"

Targets="petstoreServer"

URI="tour.war"

/>
</Application>

```

アプリケーション コンポーネント パスの `c:\bea\wlserver6.1` の部分は WebLogic Server のインストール中に決定され、パスの残りの部分はハードコード化されていることに注意してください。必要であれば、アプリケーション インストーラによって、同じような方法で WebLogic Server に関連しないサブディレクトリにアプリケーション コンポーネントをインストールできます。

スタートアップ クラス

Windows システムでは、`petstoreServer` の起動後、Pet Store アプリケーションはスタートアップ クラスを使用してシステムの Web ブラウザを起動します。このスタートアップ クラスの定義は、ほとんどの場合、インストールされた `config.xml` ファイルにハードコードされ、ポート番号だけが WebLogic Server インストール プログラムによって動的に設定されます。

```

<StartupClass
  Arguments="port=7001"
  ClassName="com.bea.estore.startup.StartBrowser"
  FailureIsFatal="false"
  Name="StartBrowser"
  Targets="petstoreServer"
  Notes="On Windows, this class automatically starts a browser after the server
has finished booting."
/>

```

fileRealm.properties の例

最後に、デフォルト ACL、グループ、およびプリンシパルが、インストールされた `fileRealm.properties` 内に定義されます。Pet Store アプリケーションの場合、`fileRealm.properties` のすべての情報がハードコード化されます。使用中のアプリケーションは、カスタムグループとプリンシパルの指定を要求し、インストールすることができ、同様に必要に応じて必要な情報をハードコード化できます。

```
acl.modify.weblogic.jndi.weblogic.fileSystem=everyone
user.j2ee=0xe22513c99d9279bdf9318894033ef310b9aa830f
acl.lookup.weblogic.jndi.weblogic.fileSystem=everyone
acl.lookup.weblogic.jndi.weblogic.ejb=system,everyone
acl.lookup.weblogic.jndi=system,everyone
acl.list.weblogic.jndi.weblogic.rmi=system,everyone
acl.lockServer.weblogic.admin=system,guest
acl.shutdown.weblogic.admin=system,guest
group.gold=
acl.boot.weblogic.server=system,everyone
user.jps_admin=0xcc0b7594b451623dd59a3db8148de6984c60ac74
...
```

ユーザパスワードは、`fileRealm.properties` 内に暗号化されることに注意してください。ユーザパスワードをハードコード化するには、Administration Console を使用してファイルに必要なユーザを追加したのち、生成されたファイルを、インストールされた `fileRealm.properties` のテンプレートとして使用します。

3 WebLogic Server MBean の使用

この章では、特定の WebLogic Server MBean の使用方法について説明します。

- [概要](#)
- [JDBC プロファイリング MBean の使用](#)

概要

注意：『[WebLogic JMX Service プログラマーズ ガイド](#)』には、WebLogic Server MBean を使用するための詳細な情報、およびコード サンプルが記載されています。JMX API に精通していない場合は、まずこのマニュアルを参照してください。

WebLogic Server で MBean を使用する基本手順は、以下のとおりです。

1. 管理サーバまたは個々の WebLogic Server インスタンスの MBeanHome インタフェースを取得します。詳細については、『[WebLogic JMX Service プログラマーズ ガイド](#)』の「[WebLogic Server MBean へのアクセス](#)」を参照してください。
2. アプリケーションで純粋な JMX メソッド呼び出しを使用する必要がある場合は、MBeanHome から MBeanServer インタフェースを取得します。
3. 使用する MBean を個々に取得します。

アプリケーションで JMX 通知を使用する場合は、「[MBean 通知の使い方](#)」で、リスナを設定する方法およびリスナを WebLogic MBean に登録する方法の詳細について参照してください。

独自のカスタム MBean をデプロイする場合は、「[WebLogic Server MBean へのアクセス](#)」を参照してください。

以降の節では、さまざまな WebLogic Server API に固有の MBean を使用方法について説明します。

JDBC プロファイリング MBean の使用

BEA では、SQL 文、prepared statements、および JDBC 接続リークのメトリックを格納および解析するために使用できる複数の JDBC MBean を提供しています。次の節では、JDBC プロファイリングを有効にする方法および使用する方法について説明します。詳細については、次の WebLogic Server MBean および関連するクラスの「[API ドキュメント](#)」を参照してください。

- [JDBCConnectionPoolMBean](#)
- [JDBCConnectionPoolRuntimeMBean](#)
- [JDBCStatementProfile](#)
- [JDBCConnectionLeakProfile](#)

JDBC プロファイリングの有効化

SQL 文または接続リーク プロファイルを解析する前に、調査する接続プールのプロファイリングを有効にする必要があります。プロファイリングが有効になると、接続プールによってメトリックは、後で解析するために外部リポジトリに格納されます。

アプリケーションは、[JDBCConnectionPoolMBean](#) を使用して JDBC プロファイリング オプションを有効および無効にします。標準接続プール プロパティの `get/set` メソッドを提供するのに加えて、[JDBCConnectionPoolMBean](#) はプロファイリングを有効および無効にする次のメソッドを提供します。

- `setConnLeakProfilingEnabled()` は、JDBC 接続リークのプロファイリングを有効または無効にします。接続リークは、接続プールからチェックアウトされても、`close()` メソッドによって返されない接続を示します。リークされた接続は、のちの接続要求の実行に使用できないため、接続リーク プロファイルを解析することは重要です。
- `setSqlStmtProfilingEnabled()` は、SQL 文のプロファイリングを有効または無効にします。このタイプのプロファイリングが有効になると、接続プールによって SQL 文および文の実行時間と他のメトリックが格納されます。SQL 文のプロファイルを解析することによって、アプリケーションで最も時間のかかるクエリを調べることができます。

- `setSqlStmtParamLoggingEnabled()` は、prepared および callable statements のバインド パラメータのプロファイリングを有効または無効にします。文パラメータは非常に大きくなる可能性があるため、プロファイルに格納されるパラメータのサイズを制限するために、`setSqlStmtMaxParamLength()` を任意で使用できます。

次の例は、`JDBCConnectionPoolMBean` を取得し、すべてのプロファイリング オプションを有効にするアプリケーションを示します。この例では、各文パラメータについて最大 20 文字を格納します。

```
// 管理サーバの MBeanHome を取得する
...
JDBCConnectionPoolMBean mbean =
    (JDBCConnectionPoolMBean)home.getConfigurationMBean(poolName,
        "JDBCConnectionPoolConfig");
mbean.setConnLeakProfilingEnabled(true);
mbean.setSqlStmtParamLoggingEnabled(true);
mbean.setSqlStmtMaxParamLength(maxLen);
...
```

JDBC プロファイルのアクセス

必要なプロファイリング オプションを有効にしたら、`JDBCStatementProfile` および `JDBCConnectionLeakProfile` クラスを使用して、格納されたメトリックを解析できます。これらのプロファイル クラスは共に、`JDBCConnectionPoolRuntimeMBean` を使用して簡単に取得できます。

`JDBCStatementProfile` は、接続プールの SQL 文と関連メトリック（およびオプションとして、バインド パラメータ）を格納します。

`JDBCConnectionLeakProfile` は、リークされた接続のスタック トレースを格納します。

すべてのプロファイルを一度に取得する場合、非常に多くのリソースが使用される可能性があります。この理由により、通常、アプリケーションは指定した時間でプロファイルのサブセットのみを検索します。これを行うには、最初にストレージ内のプロファイルの合計数を決定してから、より小さいサブセットでプロファイルを検索します。

3 WebLogic Server MBean の使用

次の例は、プロファイルの数をより小さい単位に分ける簡単な方法を示します。

```
// 接続プールをホストするサーバの MBeanHome を取得する
. . .

// 「testPool」接続プールの JDBCRuntimeMbean を取得する
String poolName = "testPool";

JDBCConnectionPoolRuntimeMBean mbean =
    (JDBCConnectionPoolRuntimeMBean)home.getRuntimeMBean
        (poolName, "JDBCConnectionPoolRuntime");

JDBCConnectionLeakProfile[] profiles = null;

// 使用可能な prepared statement のキャッシュ プロファイルの合計数を取得する
int profileCount = mbean.getConnectionLeakProfileCount();

// プロファイルの profilesPerStep の数を要求する
int profilesPerStep = 10;

// プロファイル数 profileIndex で開始する
int profileIndex = 0;
boolean done = (profileCount > 0);
while (!done) {

    // プロファイルを取得する
    profiles = mbean.getConnectionLeakProfiles(profileIndex,
        profilesPerStep);

    // 検索されたプロファイルを調査する
    for (int index = 0; index < profiles.length; index++) {

        // プール名を取得する
        String poolName = profiles[index].getPoolName();

        // スタック トレースを取得する
        String stackTrace = profiles[index].getStackTrace();
    }

    profileIndex = profileIndex + profilesPerStep - 1;

    // 取得したプロファイルの数が
    // 要求した数よりも少ない場合は終了
    done = (profiles.length < profilesPerStep);
}
}
```