



BEA WebLogic Integration™

WebLogic Integration の起動、停止および カスタマイズ

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 Systems, Inc. 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、Jolt、Tuxedo、および WebLogic は BEA Systems, Inc. の登録商標です。BEA Builder、BEA Campaign Manager for WebLogic、BEA eLink、BEA Manager、BEA WebLogic Commerce Server、BEA WebLogic Enterprise、BEA WebLogic Enterprise Platform、BEA WebLogic Express、BEA WebLogic Integration、BEA WebLogic Personalization Server、BEA WebLogic Platform、BEA WebLogic Portal、BEA WebLogic Server、BEA WebLogic Workshop、および How Business Becomes E-Business は、BEA Systems, Inc の商標です。

その他の商標はすべて、関係各社が著作権を有します。

WebLogic Integration の起動、停止およびカスタマイズ

パート番号	日付	ソフトウェアのバージョン
なし	2002年6月	7.0

目次

1. はじめに

WebLogic Server ドメイン	1-2
WebLogic Integration サンプルドメイン	1-3
WebLogic Integration コンフィグレーション テンプレート	1-5
管理および設計ツール	1-7
WebLogic Integration リポジトリについて	1-9
WebLogic Integration ユーザおよびパスワード	1-9
ドメイン コンフィグレーションの要件	1-11
ドメインに対するデータベースのコンフィグレーション	1-14
サンプルドメインのコンフィグレーションと起動	1-15
WebLogic Integration の起動	1-18
Windows での WebLogic Integration の起動	1-18
メニューから WebLogic Integration を起動する場合	1-18
コマンド ラインで WebLogic Integration を起動する場合	1-19
UNIX での WebLogic Integration の起動	1-20
起動完了メッセージ	1-21
WebLogic Integration の終了	1-21
Windows での WebLogic Integration の終了	1-22
UNIX での WebLogic Integration の終了	1-23
次のステップ	1-24

2. WebLogic Integration 管理ツールと設計ツール

Web ブラウザ コンフィグレーションの要件	2-1
クッキーの受け付け	2-2
プロキシ サーバの回避	2-2
Netscape でのプロキシ サーバの回避	2-3
Microsoft Internet Explorer でのプロキシ サーバの回避	2-3
サンプル起動ページの起動	2-4
サンプルの使用法の学習	2-6
WebLogic Server Administration Console の起動	2-8
WebLogic Server Administration Console の使用法の学習	2-10

Application View Console の起動.....	2-10
アプリケーション ビューの使用方法的学習	2-13
Studio の起動	2-14
Studio の起動手順	2-14
BPM プラグインに関する注意	2-17
Studio の使用方法的学習	2-18
Worklist の起動.....	2-19
Worklist の使用方法的学習	2-21
Format Builder の起動	2-22
Format Builder の使用方法的学習	2-24
B2B Console の起動	2-25
B2B Console の使用方法的学習	2-27
B2B Console での B2B エンジンの起動と終了.....	2-27
B2B Console での B2B エンジンの終了.....	2-28
B2B Console での B2B エンジンの再起動.....	2-30

3. WebLogic Integration のカスタマイズ

WebLogic Integration コマンドとサポート ファイル	3-2
ドメインに対する新しいデータベースのコンフィグレーション	3-2
ドメインに対するデータベースの初期化	3-5
データベース ウィザードの使用法.....	3-6
データベース接続情報.....	3-6
データベース ウィザード オプション	3-8
データベース ウィザードの仕組み.....	3-9
データベース コンフィグレーション モード	3-9
グラフィック モードによるデータベース ウィザードの使用法	3-10
コンソール モードによるデータベース ウィザードの使用法.....	3-16
WebLogic Server Administration Console からのデータベース コンフィグレーションの更新.....	3-18
JDBC 接続プールの更新.....	3-19
RDBMS レルム プロパティの更新	3-20
データベース アクセス パラメータ	3-21
データベース アクセス URL の形式.....	3-22
新規ドメインの作成とカスタマイズ	3-23
ドメイン作成の準備.....	3-23
単一サーバ用ドメインの作成およびカスタマイズ手順	3-24

サポートされているディレクトリおよびファイル	3-26
WebLogic Integration 環境の更新	3-32
カスタム Java Message Service キューのコンフィグレーション	3-34
ビジネス オペレーションに対する EJB と Java クラスのデプロイ	3-38
WebLogic Integration application 要素に EJB を追加する	3-39
Java クラスの CLASSPATH への追加	3-42
JAR ファイルを setenv コマンド ファイルに追加する方法	3-43
JAR ファイルを startWebLogic コマンド ファイルに追加する方法 ...	3-43
NULL 変数をサポートする BPM のコンフィグレーション	3-45
wli.bpm.server.evaluator.supportsNull=true	3-45
wli.bpm.server.evaluator.supportsNull=false	3-47
BPM セキュリティ モデルについて	3-48
BPM ユーザグループおよびパーミッション	3-50
セキュリティ レルムのガイドライン	3-51
RDBMS セキュリティ レルムの有効化	3-52
カスタム セキュリティ レルムのコンフィグレーション	3-54
カスタム レルムの作成	3-54
キャッシング レルムのコンフィグレーション	3-55
WLPI キャッシング レルムの有効化	3-55
キャッシング レルムの削除	3-56
パスワードの更新	3-56
システム パスワードの更新	3-57
BPM パスワードの更新	3-57
wlisystem パスワードの更新	3-58
メール セッション プロパティのカスタマイズ	3-59
ドメインのルート ディレクトリ	3-61
代替文字セットの使用	3-61
WebLogic Integration ポート番号の変更	3-62
手順 1. - WebLogic Server のリスン ポート番号の変更	3-62
WebLogic Server Administration Console でリスン ポートを変更する	3-62
コンフィグレーション ファイルを編集してリスン ポートを変更す	る
る	3-64
手順 2. - 他の WebLogic Integration ポート番号の変更	3-65

B2B URI エンドポイントの変更	3-66
Application View Console のポート番号の変更	3-67
B2B Console のポート番号の変更	3-67
WebLogic Integration シャットダウン スクリプトの変更	3-68
非永続モードの B2B をデプロイする	3-68

A. WebLogic Integration サンプル コンフィグレーション ファイル

config.xml.....	A-2
setEnv	A-15
startWeblogic.....	A-16
fileRealm.properties	A-18

B. WebLogic Integration コマンド

コマンド リスト	B-2
環境変数.....	B-5
コマンドの説明	B-13

索引

このマニュアルの内容

このマニュアルでは、**WebLogic Integration™** を初めて使用する際に必要な基本情報を提供します。具体的には、以下のトピックを取り上げます。

- 第 1 章「はじめに」では、起動および停止方法など、**WebLogic Integration** に関する基本的な情報について説明します。
- 第 2 章「**WebLogic Integration** 管理ツールと設計ツール」では、個々の設計ツールおよび管理ツールの起動方法を説明し、詳細な情報がある参照先を示します。
- 第 3 章「**WebLogic Integration** のカスタマイズ」では、標準的なカスタマイズを行う方法について説明します。
- 付録 A 「**WebLogic Integration** サンプル コンフィグレーション ファイル」では、開発者およびシステム管理者向けに、**WebLogic Integration** を構成するリソースやアプリケーションのコンフィグレーションおよびデプロイメントに使用するファイルのサンプルを示します。
- 付録 B 「**WebLogic Integration** コマンド」では、開発者およびシステム管理者を対象に、主要な **WebLogic Integration** コマンドについて説明します。

対象読者

このマニュアルは主に、**BEA WebLogic Integration** ソリューションの評価、デプロイメント、および管理を担当する開発者およびシステム管理者を対象としています。

WebLogic Server の管理、および **J2EE** アプリケーションの開発原理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。このマニュアルに進む前に、『*WebLogic Integration 入門*』を一読されることをお勧めします。

e-docs Web サイト

BEA 製品のドキュメントは、BEA Systems, Inc. の Web サイトで入手できます。BEA のホーム ページで [製品のドキュメント] をクリックするか、または「e-docs」という製品ドキュメント 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を一度に 1 ファイルずつ印刷できます。

このマニュアルの PDF 版は、Web サイトで入手できます。WebLogic Integration 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 Integration ドキュメントのホーム ページを開き、[PDF 版] ボタンをクリックして、印刷するマニュアルを選択します。

Adobe Acrobat Reader がない場合は、Adobe の Web サイト (<http://www.adobe.co.jp/>) で無料で入手できます。

サポート情報

BEA WebLogic Integration 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、WebLogic Integration の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 WebLogic Integration のリリースをお書き添えください。

本バージョンの **BEA WebLogic Integration** について不明な点がある場合、または **BEA WebLogic Integration** のインストールおよび動作に問題がある場合は、**BEA WebSupport (websupport.bea.com/custsupp)** 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
太字	用語集で定義されている用語を示す。
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
斜体	強調または書籍のタイトルを示す。
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、データ構造体とそのメンバー、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。 <i>例</i> <pre>#include <iostream.h> void main () the pointer psz chmod u+w * \tux\data\ap .doc tux.doc BITMAP float</pre>
太字の等幅 テキスト	コード内の重要な箇所を示す。 <i>例</i> <pre>void commit ()</pre>
斜体の等幅 テキスト	コード内の変数を示す。 <i>例</i> <pre>String <i>expr</i></pre>

表記法	適用
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。 <i>例</i> LPT1 SIGNON OR
{ }	構文の中で複数の選択肢を示す。実際には、この括弧は入力しない。
[]	構文の中で任意指定の項目を示す。実際には、この括弧は入力しない。 <i>例</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
	構文の中で相互に排他的な選択肢を区切る。実際には、この記号は入力しない。
...	コマンドラインで以下のいずれかを示す。 <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 実際には、この省略記号は入力しない。 <i>例</i> buildobjclient [-v] [-o name] [-f file-list]... [-l file-list]...
.	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。実際には、この省略記号は入力しない。



1 はじめに

企業内部および企業全体にまたがって、**BEA WebLogic Integration** は、**WebLogic Server** のコア インフラストラクチャと機能を拡張し、これらのアプリケーションやビジネスプロセスの開発、デプロイ、および統合を実現する完全なソリューションを提供します。

この章では、**WebLogic Integration** を初めて使用する際に必要な情報を提供します。ここでは、**WebLogic Server** ドメインのコンフィグレーションを行い、**WebLogic Integration** リソースをデプロイする方法について説明します。また、コンポーネントとパスワードの概要を説明し、サンプルドメインやカスタムドメインで **WebLogic Integration** を起動する方法を示します。

各節の内容は以下のとおりです。

- **WebLogic Server** ドメイン
- **WebLogic Integration** サンプルドメイン
- **WebLogic Integration** コンフィグレーションテンプレート
- 管理および設計ツール
- **WebLogic Integration** リポジトリについて
- **WebLogic Integration** ユーザおよびパスワード
- ドメイン コンフィグレーションの要件
- サンプルドメインのコンフィグレーションと起動
- **WebLogic Integration** の起動
- **WebLogic Integration** の終了
- 次のステップ

WebLogic Server ドメイン

WebLogic Server ドメインは、1つの単位として管理される WebLogic Server リソースの集まりです。ドメインには、1つまたは複数の WebLogic Server インスタンスと、場合によっては WebLogic Server クラスタが含まれます。

WebLogic Integration は、EJB、Web アプリケーション、JDBC 接続プールなどのアプリケーションとリソースの集まりで、包括的な E ビジネス ソリューションを開発するための統一プラットフォームを提供するためにドメインにデプロイされます。

各 WebLogic Server ドメインには、管理サーバが 1 つあります。ドメイン内の他の WebLogic Server インスタンスは、管理されるサーバ（管理対象サーバ）です。管理対象サーバをクラスタ構成にすると、一連のサーバにリソースをデプロイするタスクを簡素化できます。ドメイン全体のコンフィグレーションは、管理サーバ上に保持されます。管理対象サーバは、起動時に管理サーバからコンフィグレーションを取得します。

ドメインのコンフィグレーションは、XML (eXtensible Markup Language) で定義され、`config.xml` というファイルに格納されます。このファイルは、次のディレクトリに格納されています。

```
domain/config.xml
```

ディレクトリ名（すなわち、`domain` の値）が、ドメイン名になります。このディレクトリは、システムの任意の場所に配置できます。利便性のために、管理サーバの起動および終了に必要なスクリプトは多くの場合、`config.xml` ファイルと同じ場所にあります。

管理サーバは、ドメイン全体の中央管理ポイントです。WebLogic Server Administration Console は、管理タスクの実行を簡便化するためのグラフィカルインタフェースです。リソース パラメータ、サーバ インスタンス、デプロイメントは、管理サーバで保持されている `config.xml` 上に WebLogic Server Administration Console を介して定義されます。

WebLogic Server ドメイン管理の一般情報については、『*BEA WebLogic Server 管理者ガイド*』の「WebLogic Server システムの概要」を参照してください。

以下の節では、WebLogic Integration サンプルドメインおよび WebLogic Integration コンフィグレーション テンプレートについて説明します。カスタムドメインを作成する場合は、このコンフィグレーション テンプレートを、BEA

コンフィグレーション ウィザードで使用できます。また、サンプルドメインを使用して、サンプルアプリケーションを通じて示される機能と特長を体験することができます。

独自の **WebLogic Integration** ソリューションの開発とテストを行っていく場合、最初の段階では、**WebLogic Integration** で提供されるコンフィグレーションテンプレートのいずれかを基にしたカスタム ドメインを使用することをお勧めします。

WebLogic Integration サンプル ドメイン

サンプルドメインでは、**WebLogic Integration** サンプルアプリケーションを実行できます。サンプルドメインは、インストール時に次に示す一連のタスクのいずれかを実行した場合に作成されます。

- 標準インストールを選択。
- カスタム インストールを選択し、インストール項目の 1 つとして **WebLogic Integration** サンプルを選んだ。

このサンプルドメインでは、**WebLogic Integration** の全機能がサポートされます。さらに、**Web** アプリケーション、カスタム **JMS** キュー、およびサンプル実行に必要なその他のリソースがコンフィグレーションされます。

WebLogic Integration サンプルドメインは次のようにコンフィグレーションされます。

- ドメインには、単一の **WebLogic Server** インスタンスを含む。これは、そのドメインの管理サーバになる。
注意： 管理対象サーバの追加、およびクラスタ化環境での **WebLogic Integration** のデプロイについては、『*WebLogic Integration ソリューションのデプロイメント*』を参照してください。
- デフォルト リスポートは 7001。
- デフォルトでは、**WebLogic Platform** に **PointBase** データベースがインストールされる。
- **PointBase** データベースは、サンプル アプリケーション用にコンフィグレーションされる。

- **PointBase** データベースへの接続に必要な **JDBC** 接続プールがコンフィグレーションされる。
- 互換モードのファイルベースのセキュリティレルム (**FileRealm**) が、**PointBase** データベースを使用するようにコンフィグレーションされる。

サンプルドメインディレクトリには、対象となるドメインの **WebLogic Integration** リソースおよびアプリケーションのデプロイとコンフィグレーションの管理に使用される、複数のファイルも配置されます。それらのファイルは、次のとおりです。

- **config.xml** ファイル – **WebLogic Server** ドメイン全体のコンフィグレーションを記述した **XML** ドキュメント。ドメインでデプロイされるリソースは、**config.xml** ファイルで決定されます。
- **application.xml** ファイル – **WebLogic Integration** の **J2EE** コンポーネントのデプロイ順を決定する **XML** ドキュメント。
- **Start** コマンド – ドメインのサーバを起動するコマンド。このコマンドは、プラットフォームによって異なります。各ドメインディレクトリには、**startWeblogic.cmd** ファイル (**Windows**) か **startWeblogic** ファイル (**UNIX**) のいずれかがあります。
- **Stop** コマンド – ドメインのサーバを正常に終了させるコマンド。このコマンドは、プラットフォームによって異なります。各ドメインディレクトリには、**stopWeblogic.cmd** ファイル (**Windows**) か **stopWeblogic** ファイル (**UNIX**) のいずれかがあります。
- **fileRealm.properties** – **WebLogic Server** の起動時に作成される **User**、**Group**、および **ACL** オブジェクトを格納するファイル。互換モードのファイルベースセキュリティレルム (**FileRealm**) は、**WebLogic Integration** 環境のデフォルトセキュリティレルムです。

以上の各ファイルの例は、付録 A 「**WebLogic Integration** サンプル コンフィグレーション ファイル」にあります。

WebLogic Integration 環境の設定では、ここで説明したコマンド以外のコマンドも使用します。これらのコマンドについては、付録 B 「**WebLogic Integration** コマンド」で説明します。

WebLogic Integration コンフィグレーション テンプレート

BEA コンフィグレーション ウィザード（「[コンフィグレーション ウィザードの使い方](#)」を参照）を使用して、新しい WebLogic Server ドメインを作成できます。コンフィグレーション ウィザードで作成されるドメインには、サーバや各サーバにデプロイされるリソース（EJB、Web アプリケーション、JDBC 接続プールなど）が含まれます。どのようなサーバやリソースが含まれるかは、選択したコンフィグレーション ウィザード テンプレートやコンフィグレーション ウィザードの質問に対する回答によって異なります。

WebLogic Integration を BEA WebLogic Platform の一部としてインストールする場合、次の表に示すコンフィグレーション ウィザード テンプレートをコンフィグレーション ウィザードで使用できます。

テンプレートの対象ドメイン	コンフィグレーションされたドメインで開発するソリューションで使用する WebLogic Integration の機能
WebLogic Integration (WLI) ドメイン	全内容 – Application Integration、Data Integration、BPM (Business Process Management)、および B2B Integration
EAI (Enterprise Application Integration) ドメイン	BPM、Application Integration、および Data Integration
BPM ドメイン	BPM および Data Integration

これらのテンプレートにより、最も一般的に使用される 3 つの WebLogic Integration 開発環境を構築できます。これらのテンプレートのいずれかで作成したドメインは、ほとんどの場合、評価、開発、およびテストの要件を満たすことができます。

各テンプレートの詳細については、『[Configuration Wizard Template Reference](#)』を参照してください。

「*コンフィグレーション ウィザードの使い方*」で説明するように、コンフィグレーション ウィザードを使用してサーバに保存するドメイン コンフィグレーションを設定する際に、ドメイン名 (*domain*) の割り当てと *domain* ディレクトリのインストール先の指定を求められます。サーバが必要とするファイルは、*domain* ディレクトリの下に指定した位置にインストールされます。

たとえば、デフォルト位置 (*BEA_HOME/user_projects*) を使用する場合、管理サーバが必要とする各ファイルは、次のディレクトリにインストールされません。

BEA_HOME/user_projects/domain

このマニュアルでは、ドメインのルートへの絶対パスを示すために、*DOMAIN_HOME* を使用します。

WebLogic Integration ドメイン テンプレートのいずれかを使用して作成したドメインの管理サーバは、**WebLogic Integration** サンプルドメインの場合と同様に、次のファイルを必要とします。

- **config.xml** ファイル – **WebLogic Server** ドメイン全体のコンフィグレーションを記述した **XML** ドキュメント。管理サーバ、管理対象サーバ、サーバ クラスタ、および各サーバにデプロイされるリソースのコンフィグレーションは、管理サーバの *DOMAIN_HOME* ディレクトリに配置される **config.xml** ファイルで管理されます。
- **Start** コマンド – ドメインの管理サーバを起動するコマンド。このコマンドは、プラットフォームによって異なります。*DOMAIN_HOME* ディレクトリには、**startWeblogic.cmd** ファイル (**Windows**) か **startWeblogic** ファイル (**UNIX**) のいずれかがあります。
- **Stop** コマンド – ドメインの管理サーバを正常に終了させるコマンド。このコマンドは、プラットフォームによって異なります。*DOMAIN_HOME* ディレクトリには、**stopWeblogic.cmd** ファイル (**Windows**) か **stopWeblogic** ファイル (**UNIX**) のいずれかがあります。
- **fileRealm.properties** – **WebLogic Server** の起動時に作成される **User**、**Group**、および **ACL** オブジェクトを格納するファイル。ファイル レalm は、**WebLogic Server** 環境でのデフォルト セキュリティ レalm になります。

以上の各ファイルの例は、付録 A 「**WebLogic Integration** サンプル コンフィグレーション ファイル」にあります。

ドメインを作成するために選択したテンプレートによっては、別のコマンドが含まれることがあります。各テンプレートによりインストールされるファイルおよびコマンドの説明については、『*Configuration Wizard Template Reference*』を参照してください。使用できる **WebLogic Integration** コマンドについては、付録 B 「WebLogic Integration コマンド」を参照してください。

WebLogic Integration を十分に理解した段階で、3-23 ページの「新規ドメインの作成とカスタマイズ」の説明に従って、コンフィグレーション ウィザードで作成したドメインをカスタマイズできます。

管理および設計ツール

『*WebLogic Integration 入門*』で説明したとおり、**WebLogic Integration** には、以下の設計および管理ツールがあります。

- **WebLogic Server Administration Console**
WebLogic Server Administration Console は、WebLogic Server でデプロイされる **WebLogic Integration** リソース（アプリケーション、EJB、サービスなど）の管理に使用します。
- **Application View Console**
アプリケーション ビューは、**WebLogic Integration ADK (Adapter Development Kit)** で構築されたアダプタの技術的な実装と、対象となるアプリケーションのビジネス データおよび機能の間の抽象化の重要なレイヤを提供します。**Application View Console** では、アダプタ リソースへのアクセスおよび制御が可能になります。
- **WebLogic Integration Studio**
Studio は、プロセスのビジネス ルールを実装するのに必要なアクション（タスク、イベント、決定）の定義を容易にすることで、プロセス モデリングを簡略化するグラフィカル インタフェースです。アクションには、アプリケーションの呼び出し、ユーザとのやり取り、データ形式の変換、XML メッセージの送受信、電子メールの送信などがあります。**Studio** では、フローチャートでドラッグ アンド ドロップを使用し、アクションをワークフローに組み込むことができます。
- **WebLogic Integration Worklist**
Worklist は、プロセス エンジンと共に機能するグラフィカル アプリケーションです。それぞれのユーザに対して、アクションが必要なワークフロー

を通知します。**Worklist** を使用すると、ワークフロー開発者は、ユーザ トランザクションを簡単にワークフローに組み込むことができるようになります。

注意： **Worklist** クライアント アプリケーションは、このリリースの **WebLogic Integration** から非推奨になります。**Worklist** に代わる機能の詳細については、『*WebLogic Integration リリース ノート*』を参照してください。

■ *WebLogic Integration Format Builder*

データ変換は、エンタープライズ アプリケーション統合ソリューションで非常に重要な役割を果たします。**XML** は、アプリケーション間での情報のやり取りにおいて標準になりつつあります。**Format Builder** アプリケーションは、ユーザの組織で使用されているバイナリ データの内容と構造を記述するメッセージフォーマット言語 (**MFL**) ファイル (メッセージフォーマット) 作成ツールです。これらのメッセージフォーマットは、**Data Integration** プラグインと組み合わせて、**WebLogic Integration** プロセス エンジンでバイナリ データと **XML** のやり取りをするのに使用されます。

■ *WebLogic Integration B2B Console*

B2B Console は、トレーディング エクスチェンジ、サプライチェーン管理、および協調的商取引アプリケーションに必要な要素を定義、エクスポート、またはインポートできるようにするグラフィカル ツールです。また、**B2B** エンジン を起動および停止する場合や、現在アクティブなやり取りで **B2B** エンジン、トレーディング パートナ セッション、配信チャネルなどをモニタする場合にも **B2B Console** を使用します。

第 2 章「**WebLogic Integration** 管理ツールと設計ツール」では、以下について説明します。

- 各ツールの起動手順
- 各ツールを使用するのに必要な情報のリファレンス

WebLogic Integration リポジトリについて

WebLogic Integration リポジトリは、WebLogic Integration によって要求される情報を保持するデータベースです。以下に、リポジトリを構成するテーブルを示します。

表 1-1 WebLogic Integration リポジトリ テーブル

テーブルのセット	内容
コモン テーブル (別名 XML リポジトリ)	XML、DTD、XSLT スタイル シート、および WebLogic Integration で使用するその他のエンティティ
ワークフロー テーブル	静的ワークフローおよび動的実行時情報
JMS キュー テーブル	メッセージの状態 (動的)
B2B コンフィグレーション テーブル	B2B 統合をサポートするのに必要な静的コンフィグレーション情報。これらのテーブルに格納されるデータの詳細については、『 <i>B2B Integration 管理ガイド</i> 』の「リポジトリの操作」を参照。
永続性テーブル	永続性の状態 (動的)

ドメインのリポジトリの初期化に必要な情報については、1-11 ページの「ドメイン コンフィグレーションの要件」を参照してください。

WebLogic Integration ユーザおよびパスワード

デフォルトのユーザとパスワードは、次の 2 つのドメインでコンフィグレーションされます。

- サンプルドメイン (1-3 ページの「WebLogic Integration サンプルドメイン」を参照)

1 はじめに

- いずれかのテンプレートで作成されたドメイン（1-5 ページの「WebLogic Integration コンフィグレーション テンプレート」を参照）。

表 1-2 に、これらのドメインのユーザとパスワードを示します。

表 1-2 デフォルトのユーザ名とパスワード

ユーザ名	パスワード	説明	更新に関する考慮事項
サンプルドメインの場合 - system コンフィグレーション ウィザードを使用してドメインを作成するとき、サーバのユーザ名を指定するよう求められる。	サンプルドメインの場合 - security コンフィグレーション ウィザードを使用してドメインを作成するとき、サーバのパスワードを指定するよう求められる。	以下の操作に使用するログイン。 <ul style="list-style-type: none">■ WebLogic Server の起動■ WebLogic Server Administration Console へのアクセス■ B2B Console へのアクセス■ Application View Console へのアクセス	WebLogic Server Administration Console でこのパスワードを更新。
wlssystem	wlssystem	B2B エンジンおよびプロセス エンジンで使用する内部ログイン。	警告： このパスワードは、B2B Console で更新する必要があります。WebLogic Server Administration Console を使用しないでください。
admin	security	WebLogic Integration Studio、Worklist、および Format Builder クライアントからプロセス エンジンまたは XML リポジトリにログオンおよび接続する際に使用するデフォルト BPM ログイン。	Studio からユーザ名を更新（追加または削除）。
mary	password		WebLogic Server
joe	password		Administration Console で既存のパスワードを更新。

システム セキュリティを最大限に発揮するために、3-56 ページの「パスワードの更新」の説明に従ってデフォルト パスワードを変更できます。

コンフィグレーション テンプレートのいずれかを基にしてドメインを作成する場合は、デフォルトの **FileRealm** セキュリティ レalm がコンフィグレーション されます。 `fileRealm.properties` ファイルは、このセキュリティ レalm の初期コンフィグレーションを反映します。既存の **FileRealm** に加えて、**BPM** セキュリティをサポートするためにデフォルトの **RDBMSRealm** も使用可能です。**BPM** セキュリティ モデルの詳細、**RDBMSRealm** の有効化または代替 **BPM** セキュリティ レalm の設定の手順については、3-48 ページの「**BPM** セキュリティ モデルについて」を参照してください。

WebLogic Integration B2B 機能に関連したセキュリティの問題については、『*B2B Integration セキュリティの実装*』で説明します。

ドメイン コンフィグレーションの要件

次のデータベース コンフィグレーション タスクを、任意の **WebLogic Integration** ドメインで実行する必要があります。

- ドメインのデータベース アクセス情報を定義する。

`config.xml` ファイルには、**JDBC** 接続プールへの接続に使用されるドメイン コンフィグレーション情報が記載されています。コンフィグレーション ウィザードを使用してドメインを作成した後、**WebLogic Integration** データベース ウィザードが起動します。このウィザードで、データベース コンフィグレーション情報を設定します。詳細については、次の「ドメインに対するデータベースのコンフィグレーション」を参照してください。

WebLogic Integration サンプルドメイン、およびコンフィグレーション ウィザードで作成された新規ドメインでは、**JDBC** 接続プールは、デフォルトの **PointBase** データベースを使用するようにコンフィグレーション されます。

警告： ドメインの `config.xml` ファイルのデータベース コンフィグレーション情報を直接更新しないでください。 `config.xml` ファイルは、 `setDBVars` ファイルおよび `setDBVarsExt` ファイルのデータベース コンフィグレーション情報を使用して更新されます。これらのファイルは、 `DOMAIN_HOME\dbinfo\database_type` ディレクトリに配置 されます。ここで `DOMAIN_HOME` は、ドメインのフルパス名（ドメインの名前を含む）、 `database_type` は、データベースの名前（たとえば、 `oracle`）を意味します。 `switchdb` スクリプトやデータベース

ウィザードの [データベースの切り替え] オプションが実行されている場合、変更が上書きされる可能性があるため、`config.xml` ファイルを直接変更しないことをお勧めします。データベース コンフィグレーション情報は、データベース ウィザードを使用して更新してください。手動で変更を行う必要がある場合は、`setDBVars` ファイルおよび `setDBVarsExt` ファイルを更新してから、`switchdb` スクリプトまたはデータベース ウィザードの [データベースの切り替え] オプションを実行して、`config.xml` ファイルを再生成してください。`switchdb` コマンドの詳細については、B-50 ページの「`switchdb`」を参照してください。

- ドメインのデータベース リポジトリを初期化する。

WebLogic Integration ドメインでサーバを起動するには、最初に、`config.xml` ファイルで指定されているデータベースを初期化する必要があります。次の表では、初期化の要件を示します。

表 1-3 データベース初期化の要件

タスク ID	データベース 初期化タスク	対象
1	<p>以下のリポジトリ テーブルを作成する。</p> <ul style="list-style-type: none"> ■ コモン テーブル (XML リポジトリと総称される場合もある) – XML、DTD、XLST スタイル シート、および WebLogic Integration で使用するその他のエンティティを格納するデータベース テーブルのセット ■ ワークフロー テーブル – 静的なワークフローおよび動的な実行時情報を格納するデータベース テーブル セット (BPM データベース スキーマのエンティティの詳細については、『<i>BPM クライアント アプリケーション プログラミング ガイド</i>』の「データベース スキーマ」を参照)。 	WebLogic Integration をサポートするあらゆるドメイン
2	コモン テーブルに必要なデータ (XML スキーマなど) を入れる。	WebLogic Integration をサポートするあらゆるドメイン

表 1-3 データベース初期化の要件 (続き)

タスク ID	データベース 初期化タスク	対象
3	<p>以下のリポジトリ テーブルを作成する。</p> <ul style="list-style-type: none"> ■ B2B コンフィグレーション テーブルー B2B 統合の環境をコンフィグレーションするのに必要な静的情報を格納するデータベース テーブルのセット。これらのテーブルに格納されるデータについては、『<i>B2B Integration 管理ガイド</i>』の「リポジトリの操作」を参照。 ■ 永続性テーブルー永続性の状態を格納するデータベース テーブルのセット (動的)。 	WebLogic Integration により提供される B2B 統合機能をサポートするドメイン。
4	<p>WebLogic Integration B2B 機能をサポートするのに必要なシステム データをロードする。サポートされている各ビジネス プロトコルでは、以下が必要になる。</p> <ul style="list-style-type: none"> ■ ビジネス プロトコル定義 ■ システム ロジック プラグイン定義のセット 	WebLogic Integration B2B により提供される B2B 統合機能をサポートするドメイン。
5	<p>サンプルで必要なコンフィグレーション データとワークフローをロードする。</p>	WebLogic Integration サンプルドメイン
6	<p>End-to-End サンプル アプリケーションのアプリケーション ビューをコンフィグレーションする。</p>	WebLogic Integration サンプルドメイン

データベースの初期化プロセスでは、次のツールが役立ちます。

- 1-15 ページの「サンプルドメインのコンフィグレーションと起動」で説明するように、RunSamples コマンドは、WebLogic Integration サンプル データベースの初期化に使用します。このコマンドは、サンプル データベースの初期化に必要なすべてのタスク (上記の表の 1 ~ 6) を実行します。
- WebLogic Integration データベース ウィザードは、JDBC 接続プールの更新またはデータベース リポジトリの初期化に使用できます。詳細については、次の「ドメインに対するデータベースのコンフィグレーション」を参照してください。

ドメインに対するデータベースのコンフィグレーション

WebLogic Integration データベース ウィザードは、データベース コンフィグレーション タスクの自動化ツールです。コンフィグレーション ウィザードを使用して WebLogic Integration ドメインを作成するとき、新しいドメインにそのドメイン専用のデータベース ウィザードがインストールされます。サンプルドメインをインストールした場合は、サンプルドメイン専用のデータベース ウィザードも、そのドメインに含まれています。

データベース ウィザードには、以下のオプションがあります。

■ [データベースの切り替え]

このオプションでは、データベースの種類など、データベースへのアクセスに必要な情報を指定する必要があります。ウィザードは、次の 2 つを更新します。

- ドメインの config.xml ファイルの JDBC 接続プール。
- ウィザードがデータベースの構築に使用するドメイン固有の環境設定。

■ [データベースの作成]

このオプションでは、ドメインで必要なテーブルを作成し、ドメインに対応したデータをロードします。

たとえば、コンフィグレーション ウィザードを使用して B2B 機能サポートするドメインを作成する場合、そのドメインにインストールされる固有のデータベース ウィザードは、このオプションを選択したとき、タスク 1～4 (表 1-3 を参照) を実行します。たとえば、コンフィグレーション ウィザードを使用して EAI 機能サポートするドメインを作成する場合、そのドメインにインストールされる固有のデータベース ウィザードは、このオプションを選択したとき、タスク 1 および 2 を実行します。

また、このオプションは、必要なテーブルを作成する前に、データベース アクセス情報を入力する必要があります。現在の設定が表示されたら、この設定をそのまま受け入れるか、別のデータベースに対する接続および初期化に必要な情報を指定できます。つまり、このオプションは、ドメインのデータベースの切り替えと初期化の両方に使用できます。

注意： ドメイン専用のデータベース ウィザードは、サンプルドメインでも使用できますが、[データベースの作成] オプションは、サンプルドメインではタスク 5 と 6 を実行しません。サンプルドメインを使用す

るには、`RunSamples` コマンドを実行してサンプルドメインでサーバを設定および実行する必要があります。次の 1-15 ページの「サンプルドメインのコンフィグレーションと起動」を参照してください。

新規ドメインを作成するためにコンフィグレーション ウィザードを使用する際は、データベース ウィザードを使用してリポジトリを初期化する必要があります。手順については、3-6 ページの「データベース ウィザードの使用法」を参照してください。リポジトリを初期化したら、1-18 ページの「**WebLogic Integration** の起動」の説明に従って、ドメインの管理サーバを起動できます。

次の節では、サンプルドメインの設定および起動に必要な情報について説明します。

サンプルドメインのコンフィグレーションと起動

WebLogic Integration をインストールしてから、またはデータベース ウィザードを使用してサンプルドメインを新しいデータベースに切り替えてから初めて `RunSamples.cmd` コマンド (**Windows**) または `RunSamples` コマンド (**UNIX**) を実行すると、コマンドは以下のタスクを実行します。

- データベース内に必要なリポジトリ テーブルを作成する。表 1-3 を参照。
- B2B 統合機能に必要なシステム データおよびサンプル コンフィグレーション データをロードする。
- サンプルドメインで **WebLogic Integration** を起動する。
- **WebLogic Integration** サンプル アプリケーションのアプリケーション ビューをコンフィグレーションする。
- サンプル ワークフローを含むワークフロー パッケージをインポートする。
- デフォルト Web ブラウザを起動し、サンプル起動ページを表示する。

注意： サンプル起動ページを適切に表示するには、Web ブラウザは、2-1 ページの「Web ブラウザ コンフィグレーションの要件」で説明している要件を満たしている必要があります。

以下の各プラットフォームに応じた手順を実行して `RunSamples` コマンドを実行します。

■ Windows の場合

[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Integration Examples | Start Server and Launch Examples (with dataloader)]

`RunSamples` コマンドを Windows システムで実行する場合、コマンド ウィンドウが開きます。

■ UNIX の場合

a. `SAMPLES_HOME/integration/config/samples` ディレクトリに移動します。ここで、`SAMPLES_HOME` は、**WebLogic Platform** サンプルディレクトリを示します。

b. `RunSamples` コマンドを実行します。

たとえば、次のコマンドを入力します。

```
cd SAMPLES_HOME/integration/samples/bin
./RunSamples
```

Windows プラットフォームでも UNIX プラットフォームでも、コマンドの進行状況およびメッセージが表示されます。すべてのタスクを完了するには数分かかる場合もあります。コマンドが完了すると、以下のアクションが実行されます。

■ コマンド ウィンドウに次のメッセージが表示される。

```
Trailer:
REM
REM
=====
REM The program will stop when the server is shut down
REM WARNING: Killing this command will kill the server too!!!
REM
=====+=====
REM

RunSamples execution successful
```

■ デフォルト Web ブラウザが起動し、サンプル起動ページが表示される。

`RunSamples` スクリプトは、必要に応じてサンプル データベースをコンフィグレーションするだけでなく、サンプルドメインで **WebLogic Integration** を起動します (サーバはバックグラウンドで起動されます)。初めて `RunSamples` コマン

ドを実行すると、データベースが正しくコンフィグレーションされている場合は、サンプルドメインでの **WebLogic Integration** の起動が以下のいずれかの方法で可能になります。

- 1-18 ページの「**WebLogic Integration** の起動」の説明に従って、通常どおりに **WebLogic Integration** を起動する。サーバが起動すると、2-4 ページの「サンプル起動ページの起動」で説明されているサンプル起動ページが表示されます。

- `RunSamples` コマンドを再び実行する。

サンプル データベースを適切にコンフィグレーションした後に `RunSamples` コマンドを実行すると、次のメッセージが表示されます。

```
The WebLogic Integration repository has already been created
and populated, possibly from a previous run of this RunSamples
script. Do you want to destroy all the current data in the
repository and create and populate the WebLogic Integration
repository, again?
```

Y for Yes, N for No

「N」を入力すると、`RunSamples` コマンドは、リポジトリの作成およびデータ格納手順をスキップして、サンプルドメインで **WebLogic Server** を起動する手順と、**Web** ブラウザでサンプル起動ページを表示する手順のみを実行します。

「Y」を入力すると、`RunSamples` コマンドは、リポジトリを再作成し、データを再び格納します。次に、**WebLogic Server** を起動し、**Web** ブラウザでサンプル起動ページを表示します。「Y」を入力した場合、`RunSamples` コマンドは、現在リポジトリにある全データを破棄し、リポジトリにサンプルデータを再ロードします。現在のサンプルデータが変更または削除され、新規または未変更のサンプルデータをリポジトリに格納する場合にのみ、「Y」を入力してください。

注意： サーバの実行時にブラウザを終了しても、2-4 ページの「サンプル起動ページの起動」で説明されている手順でサンプル起動ページを表示できます。

WebLogic Integration の起動

1-3 ページの「WebLogic Integration サンプルドメイン」で説明されているように、ドメインに対して `startWeblogic.cmd` コマンド (Windows) または `startWeblogic` コマンド (UNIX) を実行すると、WebLogic Server が起動し、ドメインの `config.xml` ファイルで指定した WebLogic Integration アプリケーションおよびリソースがサーバにデプロイされます。

次の節では、Windows または UNIX システムで起動コマンドを実行する手順を示します。

Windows での WebLogic Integration の起動

Windows システムでは、メニューまたはコマンド ラインから WebLogic Integration を起動できます。

注意： 起動するサーバで使用するものと同じリスポートを使用している WebLogic Server のインスタンスをすでに実行している場合は、起動コマンドを実行する前に最初のサーバを終了する必要があります。

メニューから WebLogic Integration を起動する場合

メニューで WebLogic Integration を起動するには、以下のいずれか 1 つを実行します。

- WebLogic Integration サンプルドメインでサーバを起動するには、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Integration Examples | Start Server (only)] を選択します。

注意： サンプルドメインで初めて WebLogic Integration を起動する場合は、1-15 ページの「サンプルドメインのコンフィグレーションと起動」の説明に従って `RunSamples` コマンドを使用します。

- コンフィグレーション ウィザードで作成したドメインでサーバを起動するには、[スタート | プログラム | BEA WebLogic Platform 7.0 | User Projects | *domain* | start Server] を選択します。

コマンド ウィンドウが開き、起動メッセージが表示されます。1-21 ページの「起動完了メッセージ」で説明されているメッセージが表示されたら、起動は正常に終了したことになります。

コマンド ラインで WebLogic Integration を起動する場合

コマンド ラインで WebLogic Integration を起動する手順は次のとおりです。

1. [スタート | ファイル名を指定して実行] を選択します。
2. [名前] テキスト ボックスで `cmd` を入力して [OK] をクリックします。
3. 以下のいずれか 1 つを実行します。

- コンフィグレーション ウィザードで作成したドメインでサーバを起動するには、`DOMAIN_HOME` ディレクトリに移動します。

たとえば、`c:\bea\user_projects` ディレクトリに `mydomain` を作成した場合は、次のように入力します。

```
cd \bea\user_projects\mydomain
```

- WebLogic Integration サンプルドメインでサーバを起動するには、`SAMPLES_HOME\integration\config\samples` ディレクトリに移動します。ここで、`SAMPLES_HOME` は、WebLogic Platform サンプル ディレクトリを意味します。

たとえば、`c:\bea\weblogic700` ディレクトリに WebLogic Platform をインストールした場合は、次のように入力します。

```
cd \bea\weblogic700\samples\integration\config\samples
```

4. 次のように入力してサーバを起動します。

```
startWeblogic
```

1-21 ページの「起動完了メッセージ」で説明されているメッセージが表示されたら、起動は正常に終了したことになります。

UNIX での WebLogic Integration の起動

注意： 起動するサーバで使用するものと同じリスンポートを使用している WebLogic Server のインスタンスをすでに実行している場合は、起動コマンドを実行する前に最初のサーバを終了する必要があります。

UNIX システムで WebLogic Integration を起動する手順は次のとおりです。

1. 以下のいずれか 1 つを実行します。

- コンフィグレーション ウィザードで作成したドメインでサーバを起動するには、`DOMAIN_HOME` ディレクトリに移動します。

たとえば、`/home/bean/user_projects` ディレクトリに、`mydomain` を作成する場合、次のように入力します。

```
cd /home/bean/user_projects/mydomain
```

- WebLogic Integration サンプルドメインでサーバを起動するには、`SAMPLES_HOME/integration/config/samples` ディレクトリに移動します。ここで、`SAMPLES_HOME` は、WebLogic Platform サンプルディレクトリを意味します。

たとえば、`/home/bean/weblogic700` ディレクトリに WebLogic Platform をインストールした場合は、次のように入力します。

```
cd /home/bean/weblogic700/samples/integration/config/samples
```

2. 次のように入力してサーバを起動します。

```
./startWeblogic
```

注意： サーバをバックグラウンドのタスクとして実行したい場合は、`nohup` コマンドを使用して、`startWeblogic` コマンドが終了あるいはハングアップすることなく実行されるようにします。たとえば、次のように入力します。

```
nohup startWebLogic &
```

以下の節で説明されているメッセージが表示されたら、起動は正常に終了したことになります。

起動完了メッセージ

以下のメッセージが表示された場合は、WebLogic Integration は正常に起動しています。

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>  
<BPM> <000000> <Started WebLogic Integration - BPM Server version  
7.0>
```

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>  
<Management> <141052> <Application Poller started for development  
server.>
```

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>  
<WebLogicServer> <000354> <Thread "ListenThread.Default" listening  
on port 7001>
```

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>  
<WebLogicServer> <000331> <Started WebLogic Admin Server  
"myserver" for domain "samples" running in Development Mode>
```

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>  
<WebLogicServer> <000365> <Server state changed to RUNNING>
```

```
<Month DD, YYYY hh:mm:ss meridianAbbreviation Timezone> <Notice>  
<WebLogicServer> <000360> <Server started in RUNNING mode>
```

ドメインのコンフィグレーションによっては、さらにメッセージが表示されることもあります。

WebLogic Integration を正常に終了するには、1-22 ページの「Windows での WebLogic Integration の終了」の説明に従って、stopWebLogic コマンドを実行します。コマンド ウィンドウを閉じたり、[Ctrl] + [C] を押ししたりしてサーバを終了しないでください。

WebLogic Integration の終了

ドメイン ディレクトリにある stopWeblogic.cmd コマンド (Windows) または stopWeblogic コマンド (UNIX) を実行すると、WebLogic Integration システム全体 (WebLogic Server、およびコンフィグレーション済みドメインでデプロイ

されたすべての WebLogic Integration アプリケーションおよびリソース) を終了できます。次の節では、Windows または UNIX システムで終了コマンドを実行する手順を示します。

Windows での WebLogic Integration の終了

WebLogic Integration を終了する手順は次のとおりです。

1. [スタート | ファイル名を指定して実行] を選択します。
2. [名前] テキストボックスで `cmd` を入力して [OK] をクリックします。
3. 以下のいずれか 1 つを実行します。
 - コンフィグレーション ウィザードで作成したドメインでサーバを終了するには、`DOMAIN_HOME` ディレクトリに移動します。
たとえば、`c:\bea\user_projects` ディレクトリに `mydomain` を作成した場合は、次のように入力します。

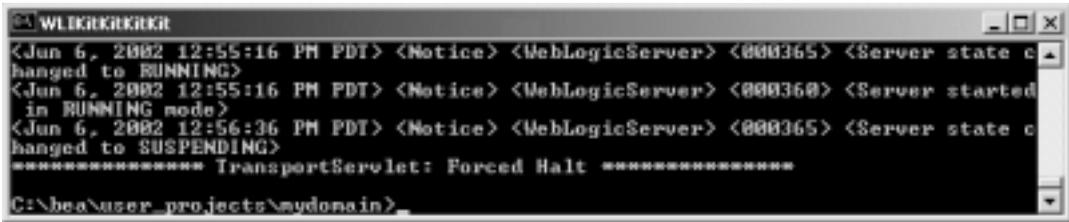
```
cd \bea\user_projects\mydomain
```
 - WebLogic Integration サンプルドメインでサーバを終了するには、`SAMPLES_HOME\integration\config\samples` ディレクトリに移動します。ここで、`SAMPLES_HOME` は、WebLogic Platform サンプルディレクトリを意味します。
たとえば、`c:\bea\weblogic700` ディレクトリに WebLogic Platform をインストールした場合は、次のように入力します。

```
cd \bea\weblogic700\samples\integration\config\samples
```
4. 次のように入力して WebLogic Integration を終了します。

```
stopWeblogic
```

シャットダウン メッセージが、図 1-1 のように表示されます。

図 1-1 シャットダウン進行メッセージ



```
WELDKKIKKKI
<Jun 6, 2002 12:55:16 PM PDT> <Notice> <WebLogicServer> <000365> <Server state c
hanged to RUNNING>
<Jun 6, 2002 12:55:16 PM PDT> <Notice> <WebLogicServer> <000360> <Server started
in RUNNING mode>
<Jun 6, 2002 12:56:36 PM PDT> <Notice> <WebLogicServer> <000365> <Server state c
hanged to SUSPENDING>
***** TransportServlet: Forced Halt *****
C:\bea\user_projects\mydomain>
```

UNIX での WebLogic Integration の終了

UNIX システムで WebLogic Integration を終了する手順は次のとおりです。

1. 以下のいずれか 1 つを実行します。

- コンフィグレーション ウィザードで作成したドメインでサーバを起動するには、`DOMAIN_HOME` ディレクトリに移動します。

たとえば、`/home/bea/user_projects` ディレクトリに、`mydomain` を作成する場合、次のように入力します。

```
cd /home/bea/user_projects/mydomain
```

- WebLogic Integration サンプルドメインでサーバを終了するには、`SAMPLES_HOME/integration/config/samples` ディレクトリに移動します。

たとえば、`/home/bea/weblogic700` ディレクトリに WebLogic Platform をインストールした場合は、次のように入力します。

```
cd /home/bea/weblogic700/samples/integration/config/samples
```

2. 次のように入力して WebLogic Integration を終了します。

```
./stopWeblogic
```

シャットダウン メッセージが、図 1-1 のように表示されます。

次のステップ

WebLogic Integration 管理ツールおよび設計ツールを起動する方法について学習するには、第2章「WebLogic Integration 管理ツールと設計ツール」を参照してください。

サンプルアプリケーションを利用することをお勧めします。これは、WebLogic Integration で Web アプリケーション、データベース、エンタープライズ情報システム、プロセス、およびビジネス パートナに接続する方法を学ぶには最良の方法です。

既に実行中の場合は、以下の操作をお勧めします。

- 「サンプルドメインのコンフィグレーションと起動」の説明に従って、サンプルドメインをコンフィグレーションする。
- WebLogic Integration のサンプル アプリケーションを実行する。

WebLogic Integration サンプル（サンプル起動ページで「WLI sample」と表示されています）では、既存のシステムの統合、複雑なビジネスプロセスの合理化、ビジネス パートナへの接続を行うための WebLogic Integration のさまざまな機能の使用例を示しています。『WebLogic Integration チュートリアル』では、このサンプルについて手順を追って説明しています。

2 WebLogic Integration 管理ツール と設計ツール

この章では、WebLogic Integration 設計ツールと管理ツールを初めて使用する際に必要な情報を提供します。この章の内容は以下のとおりです。

- Web ブラウザ コンフィグレーションの要件
- サンプル起動ページの起動
- WebLogic Server Administration Console の起動
- Application View Console の起動
- Studio の起動
- Worklist の起動
- Format Builder の起動
- B2B Console の起動
- B2B Console での B2B エンジンの起動と終了

Web ブラウザ コンフィグレーションの要件

WebLogic Integration を使用するには、Netscape Navigator 4.7 以降または Microsoft Internet Explorer 5.x 以降を使用する必要があります。さらに、クッキーを受け付けて、プロキシサーバを回避するようにブラウザをコンフィグレーションする必要があります。

注意： UNIX システムでは、PATH 変数に実行可能なブラウザ `netscape` を指定する必要があります。

クッキーの受け付け

各ブラウザに対応した手順を実行して、クッキーを受け付けるようにブラウザをコンフィグレーションします。

- **Netscape Navigator** でクッキーを受け付けるようにコンフィグレーションする手順は次のとおりです。
 - a. [編集 | プリファレンス] を選択します。
[プリファレンス] ダイアログ ボックスが表示されます。
 - b. [カテゴリ] リストから [詳細] を選択し、[詳細] オプションを表示します。
 - c. [すべてのクッキー] オプションを選択します。
- **Microsoft Internet Explorer** でクッキーを受け付けるようにコンフィグレーションする手順は次のとおりです。
 - a. [ツール | インターネット オプション] を選択します。
[インターネット オプション] ダイアログ ボックスが表示されます。
 - b. 対応する Web コンテンツ ゾーンのセキュリティ レベル設定をカスタマイズして、クッキーを有効にします。

プロキシ サーバの回避

プロキシ サーバを使用してインターネットにアクセスするように Web ブラウザがコンフィグレーションされている場合、以下にアクセスする際にプロキシを回避する必要があります。

- WebLogic Integration サンプル起動ページ
- WebLogic Server Administration Console
- WebLogic Integration Application View Console
- WebLogic Integration B2B Console

Netscape でのプロキシ サーバの回避

Netscape でのプロキシ サーバの回避

1. [編集 | プリファレンス] を選択し、[プリファレンス] ダイアログ ボックスを開きます。
2. [カテゴリ] リストの [詳細] カテゴリの [+] 記号をクリックして拡張します。
3. [プロキシ] をクリックして [プロキシ] オプションを表示します。
4. [プロキシの手動設定] をクリックして [表示] をクリックします。
5. [以下で始まるドメインにはプロキシ サーバを使用しない] の下のフィールドで、localhost、または WebLogic Integration が動作しているシステムの IP アドレスを入力します。

Microsoft Internet Explorer でのプロキシ サーバの回避

Microsoft Internet Explorer でプロキシ サーバを回避する手順は次のとおりです。

1. [ツール | インターネット オプション] を選択し、[インターネット オプション] ダイアログ ボックスを表示します。
2. [接続] タブを選択します。
3. タブの下部にある [LAN の設定] ボタンをクリックし、[ローカル エリア ネットワーク (LAN) の設定] ダイアログ ボックスを表示します。
4. [プロキシ サーバ] セクションの [ローカル アドレスにはプロキシ サーバを使用しない] をオンにします。
5. [OK] をクリックして設定を保存し、[ローカル エリア ネットワーク (LAN) の設定] ダイアログ ボックスを閉じます。
6. [OK] をクリックして、[インターネット オプション] ダイアログ ボックスを閉じます。

サンプル起動ページの起動

サンプル起動ページは、WebLogic Integration で提供されているサンプルアプリケーションの機能を確認するための入口に当たります。サンプルドメインで初めて WebLogic Integration を起動する場合は、1-15 ページの「サンプルドメインのコンフィグレーションと起動」の説明に従って `RunSamples` コマンドを実行する必要があります。このコマンドは、サンプルドメインをコンフィグレーションし、サーバを起動し、サンプル起動ページを開きます。

サンプルドメインをコンフィグレーションしたら、いつでもサンプル起動ページにアクセスできます。サーバが実行されている場合、ブラウザでサンプル起動ページを開くこともできますし、次の手順に従ってショートカットを使用することもできます。

サンプル起動ページを開く手順は次のとおりです。

1. まだ実行されていない場合は、1-18 ページの「WebLogic Integration の起動」の説明に従って、サンプルドメインで WebLogic Integration を起動します。
2. 以下のいずれか 1 つを実行します。

- Windows または UNIX システムでは、Web ブラウザで次の URL を開きます。

```
http://host:7001/index.html
```

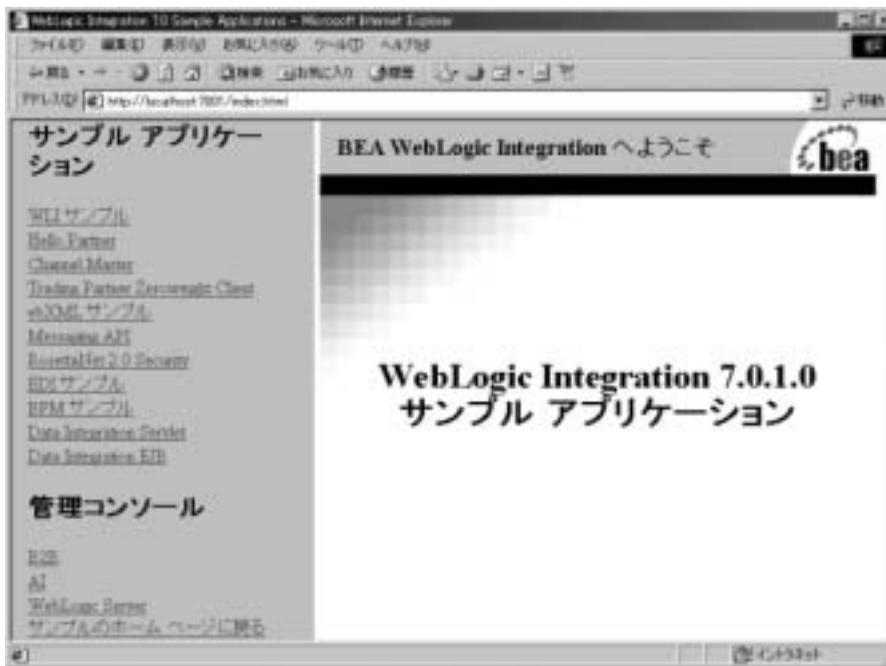
この URL では、`host` はサンプルドメインのサーバを実行しているシステムのコンピュータ名または IP アドレス、7001 は、そのサーバ用にコンフィグレーションされたリスンポートです。サーバが実行されているシステムは特に限定されていません。ローカルコンピュータで実行されている場合は、`localhost` または `127.0.0.1` を指定します。

- Windows システムでは、サーバがローカルコンピュータで実行されている場合、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Integration Examples | Launcher Examples (start server first)] の順に選択します。

このショートカットは、ローカルシステムで実行されているサーバのサンプル起動ページを開きます。

サンプル起動ページは、次のように Web ブラウザで表示されます。

図 2-1 WebLogic Integration サンプル起動ページ



サンプルの使用法の学習

次の表に、サンプルアプリケーションについて説明しているマニュアルを示します。

サンプル	参照するマニュアル
WebLogic Integration (WLI)	『 <i>WebLogic Integration チュートリアル</i> 』 注意： このサンプルは本リリースでは非推奨となった XOCP プロトコルに基づいています。詳細については『 <i>BEA WebLogic Integration リリース ノート</i> 』を参照してください。
Business Process Management (BPM)	『 <i>WebLogic Integration BPM ユーザーズガイド</i> 』(WebLogic Integration で BPM を操作するためのチュートリアル)。
B2B サンプル	『 <i>B2B Integration サンプルの使い方</i> 』
<ul style="list-style-type: none"> ■ Hello Partner ■ Channel Master ■ Trading Partner Zeroweight Client ■ Messaging API ■ RosettaNet 2.0 Security 	注意： Hello Partner、Channel Master および Messaging API サンプルは、本リリースでは非推奨となった XOCP プロトコルに基づいています。Trading Partner Zeroweight Client サンプルは、本リリースでは非推奨となった JSP tag library for the Messaging API、B2B ブラウザ クライアントおよび B2B ファイル共有 クライアントに基づいています。詳細については『 <i>BEA WebLogic Integration リリース ノート</i> 』を参照してください。
電子データ交換 (EDI)	『 <i>WebLogic Integration EDI ユーザーズガイド</i> 』の「EDI のサンプル」

サンプル	参照するマニュアル
Data Integration サンプル <ul style="list-style-type: none">■ サードレット■ EJB	『 <i>Data Integration</i> プラグイン ユーザーズガイド』の「WebLogic Integration サンプルアプリケーションの実行」

WebLogic Server Administration Console の起動

WebLogic Server Administration Console を使用すると、WebLogic Server でデプロイされた WebLogic Integration リソースのコンフィグレーションを表示または変更できます。

WebLogic Server Administration Console を起動する手順は次のとおりです。

- 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 以下の各プラットフォームに応じた手順を実行して WebLogic Server Administration Console を起動します。

- Windows または UNIX システムでは、Web ブラウザで次の URL を開きます。

```
http://host:port/console
```

この URL では、*host* はドメインの管理サーバを実行しているシステムのコンピュータ名または IP アドレス、*port* は、そのサーバ用にコンフィグレーションされたリスンポートです。サーバが実行されているシステムは特に限定されていません。ローカルコンピュータで実行されている場合は、localhost または 127.0.0.1 を指定します。

たとえば、管理サーバがローカルコンピュータ上で実行され、ポート 7001 でリスンするようにコンフィグレーションされている場合、次のように入力します。

```
http://localhost:7001/console
```

- Windows システムでは、管理サーバがローカルコンピュータで実行されている場合、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Server 7.0 | Server Tour and Examples | Start Admin Console] を選択します。

このショートカットは、ローカルシステムで実行されている管理サーバのコンソールを起動します。

3. 入力を求められたら、サーバのユーザ名とパスワードを入力します。

注意： デフォルトのユーザ名とパスワードについては、1-9 ページの「WebLogic Integration ユーザおよびパスワード」を参照してください。

WebLogic Server Administration Console は、次のように Web ブラウザに表示されます。

図 2-2 WebLogic Server Administration Console



注意： すぐにアクセスできるようにするために、WebLogic Server Administration Console をブックマークします。

WebLogic Server Administration Console の使用方法の学習

WebLogic Server 管理の詳細については、次の URL の WebLogic Server 7.0 マニュアルを参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/index.html>

WebLogic Server Administration Console の詳細は、オンライン ヘルプでも提供されます。ヘルプについては、必要に応じてページの右上隅にある疑問符をクリックしてください。

Application View Console の起動

Application View Console を使用すると、WebLogic Integration Adapter Development Kit (ADK) で開発するアダプタのアプリケーション ビューを定義できます。アプリケーション ビューは、アクティブなドメインでデプロイされるアダプタに対してのみ定義できます。アプリケーション ビューを定義する際に、エンタープライズアプリケーションに対する XML ベースのインタフェースを作成できます。Application Integration プラグインと共にこのビューを使用し、Studio でモデリングしたワークフローでアプリケーションを使用するオペレーションを含めることができます。

Application View Console を起動する手順は次のとおりです。

1. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。ビューの定義対象となるアダプタをデプロイするようにドメインをコンフィグレーションする必要があります。各タイプのアダプタをデプロイする要件の詳細については、『アダプタの開発』を参照してください。

注意： Application View Console は、ドメインが Application Integration 機能をサポートするようにコンフィグレーションされている場合のみ使用できます。

2. 以下の各プラットフォームに対応した手順を実行してコンソールを起動します。

- Windows または UNIX システムでは、Web ブラウザで次の URL を開きます。

```
http://host:port/wlai
```

この URL では、*host* は WebLogic Integration が実行されているシステムのコンピュータ名または IP アドレス、*port* は、そのサーバ用にコンフィグレーションされたリスンポートです。WebLogic Integration が実行されているマシンは特に限定されていません。ローカルコンピュータで実行されている場合は、localhost または 127.0.0.1 を指定します。

たとえば、管理サーバがローカルコンピュータ上で実行され、ポート 7001 でリスンするようにコンフィグレーションされている場合、次のように入力します。

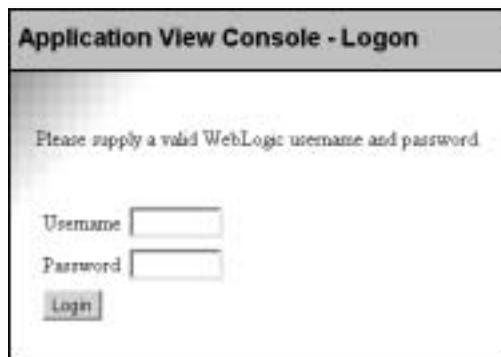
```
http://localhost:7001/wlai
```

- Windows システムでは、WebLogic Integration がローカルコンピュータで実行されている場合、[スタート | プログラム | BEA WebLogic Platform 7. 0 | WebLogic Integration 7. 0 | Application View Console] の順に選択します。

このショートカットは、ローカルシステムで実行されているサーバのコンソールを起動します。

3. サーバのユーザ名とパスワードを入力します。[Login] をクリックします。

図 2-3 アプリケーション ビュー ログオン画面



注意： 1-9 ページの「WebLogic Integration ユーザおよびパスワード」を参照。

Application View Console のホーム ページが表示されます。たとえば、次の図は、サンプルドメインの Application View Console の初期ページです。

図 2-4 サンプルドメインの Application View Console ページ



このページから、既存のビューにアクセスするか、デプロイされたアダプタの新しいビューを定義できます。

アプリケーションビュー名を選択して、指定したビューの概略が表示されるページにアクセスします。たとえば、次の図は、WebLogic Integration サンプルで使用する WLISAppView アプリケーションビューの概略ページです。

図 2-5 WLISAppView アプリケーションビューの概略ページ



アプリケーションビューの使用法の学習

BEA WebLogic Integration ADK で構築されるアダプタを使用する際の詳細については、『*Application Integration ユーザーズガイド*』を参照してください。このマニュアルでは、アプリケーションビュー サービスとイベントの定義および使用方法について説明します。

Studio の起動

WebLogic Integration Studio は、ワークフローを作成およびモニタするためのプロセスモデリングクライアントアプリケーションです。studio.cmd コマンド (Windows) または studio コマンド (UNIX) を実行すると、WebLogic Integration のローカルインスタンスまたはリモートインスタンスで実行されているプロセスエンジンにログインするよう要求されます。

以下の節では、Studio を初めて使用するために必要な情報を提供します。

- Studio の起動手順
- BPM プラグインに関する注意
- Studio の使用方法の学習

Studio の起動手順

Studio を起動する手順は次のとおりです。

- 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 以下の各プラットフォームに対応した手順を実行して Studio を起動します。
 - Windows:
[スタート | プログラム | BEA WebLogic Platform 7. 0 | WebLogic Integration 7. 0 | Studio] を選択します。

studio.cmd コマンドを実行すると、コマンド ウィンドウが短い間だけ表示されます。しばらくしてから、Studio ウィンドウが表示されます。
 - UNIX:
WLI_HOME/bin ディレクトリに移動し、studio コマンドを実行します。
たとえば、/home/bean/weblogic700/integration ディレクトリに WebLogic Integration をインストールした場合は、次のように入力します。

```
cd /home/bea/weblogic700/integration/bin
./studio
```

次の図に示すログオン ダイアログ ボックスが表示されます。

図 2-6 Studio ログオン ダイアログ ボックス



3. ユーザ名とパスワードを入力します。WebLogic Integration プロセス エンジンに対するユーザ名とパスワードが割り当てられていない場合、デフォルトのユーザ名とパスワードを入力します。1-9 ページの「WebLogic Integration ユーザおよびパスワード」を参照。
4. [サーバ URL] フィールドで、WebLogic Integration を実行しているシステムを次のように指定します。

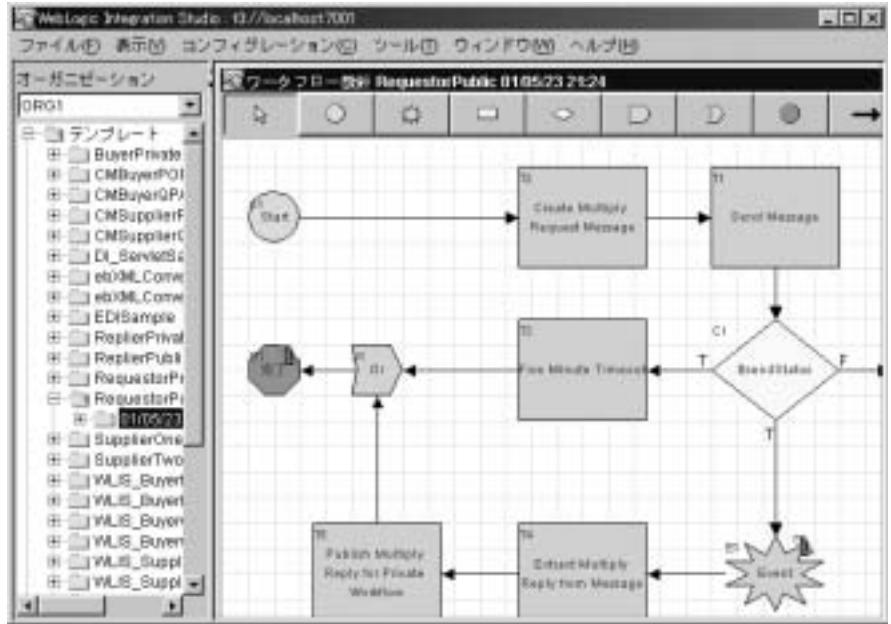
```
t3://host:port
```

ここで *host* は、システムのコンピュータ名または IP アドレスです。また、*port* は、サーバ用にコンフィグレーションされたリスン ポートです。ローカル コンピュータで WebLogic Integration が実行されている場合は、localhost または 127.0.0.1 を指定します。

5. [OK] をクリックします。

接続すると、Studio アプリケーション ウィンドウが表示されます。次の図は、サンプルドメインでプロセス エンジンに接続されたときに表示される Studio アプリケーション ウィンドウです。ここでは、Hello Partner サンプルの要求側パートナーのパブリックワークフローが選択されています。

図 2-7 Studio で表示されるサンプル ワークフロー



BPM プラグインに関する注意

WebLogic Integration プロセス エンジンと Studio クライアント アプリケーションの機能は、組み込み BPM プラグインまたはユーザが開発した BPM プラグインによって拡張できます。デプロイされるプラグインは、アクティブなドメインのコンフィグレーションによって異なります。次の表では、WebLogic Integration ドメインでのプラグインのデプロイメントを示します。

表 2-1 ドメイン固有のプラグインのデプロイメント

プラグイン	Studio 機能を拡張した結果 可能になること	デプロイするドメイン
Data Integration	Format Builder で開発された メッセージフォーマットでデー タ変換を実行する。	任意の WebLogic Integration ドメイン
Application integration	アプリケーションビューサー ビスおよびイベントをワークフ ローで使用する。	Application Integration 機 能をサポートする任意の WebLogic Integration ド メイン
B2B Integration	トレーディング パートナ交換で 使用するためのワークフローを 作成する。	B2B 機能をサポートする 任意の WebLogic Integration ドメイン

現在デプロイされているプラグインのリストを表示するには、[Studio] メニューから、[コンフィグレーション | プラグイン] を選択します。

カスタム プラグインの開発方法については、『WebLogic Integration BPM プラグインプログラミングガイド』を参照してください。

Studio の使用方法の学習

Studio を使用する際の詳細については、以下のマニュアルで説明します。

学習内容	参照するマニュアル
基礎知識	『WebLogic Integration Studio ユーザーズ ガイド』 『WebLogic Integration BPM ユーザーズ ガイド』 (手順に従って説明するチュートリアル)
B2B アプリケーションで使用するワークフローを作成する方法	『B2B Integration ワークフローの作成』
アプリケーション ビューサービ スおよびイベントをワークフ ローで使用方法	『Application Integration ユーザーズ ガイド』の 「Studio におけるアプリケーション ビューの使用」
ワークフロー アクションでデー タ変換を利用する方法	『Data Integration プラグイン ユーザーズ ガイ ド』
ユーザ独自のプラグインを作成 して機能を拡張する方法	『WebLogic Integration BPM プラグインプログ ラミング ガイド』

Studio を使用する際の詳細については、以下のオンライン ヘルプでも説明しま
す。

ヘルプの対象	選択
Studio を使用する際の一般的な情報	[ヘルプ ヘルプ トピック]
Application Integration プラグイン	[ヘルプ プラグイン ヘルプ AI ヘル プ]
Data Integration プラグイン	[ヘルプ プラグイン ヘルプ DI ヘル プ]

注意： B2B Integration プラグイン用のオンライン ヘルプは用意されていません。B2B Integration プラグインの使用法の詳細については、『*B2B Integration ワークフローの作成*』を参照してください。

Worklist の起動

Worklist を使用すると、ユーザの操作が必要なワークフロー タスクに関するシステムからの通知およびユーザからの応答を管理できます。次の手順では、Worklist の起動方法を説明します。

注意： Worklist クライアント アプリケーションは、このリリースの WebLogic Integration からは非推奨になります。Worklist に代わる機能の詳細については、『*WebLogic Integration リリースノート*』を参照してください。

Worklist を起動する手順は次のとおりです。

- 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 以下の各プラットフォームに対応した手順を実行して Studio を起動します。

- Windows:

[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Worklist] を選択します。

worklist.cmd コマンドを実行すると、コマンド ウィンドウが短い間だけ表示されます。しばらくしてから、Worklist アプリケーション ウィンドウが表示されます。

- UNIX:

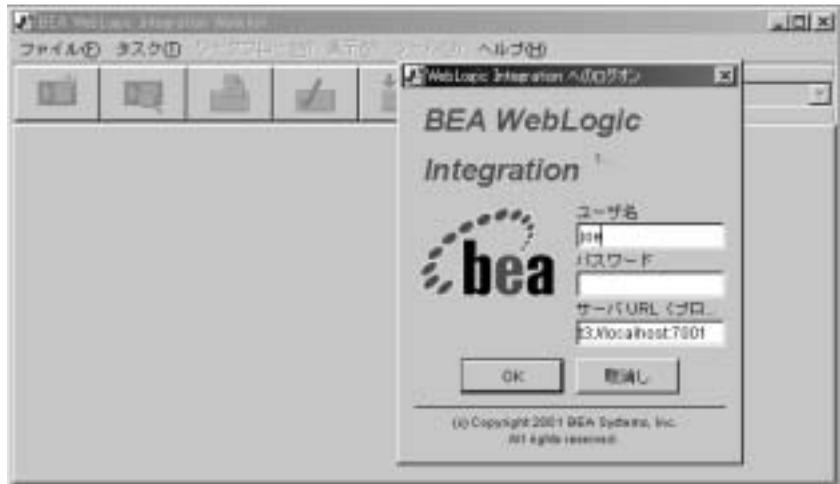
WLI_HOME/bin ディレクトリに移動し、worklist コマンドを実行します。

たとえば、/home/bean/weblogic700/integration ディレクトリに WebLogic Integration をインストールした場合は、次のように入力します。

```
cd /home/bean/weblogic700/integration/bin
. ./worklist
```

次の図に示すログオン ダイアログ ボックスが表示されます。

図 2-8 Worklist ログオン ダイアログ ボックス



3. ユーザ名とパスワードを入力します。WebLogic Integration プロセス エンジンに対するユーザ名とパスワードが割り当てられていない場合、デフォルトのユーザ名とパスワードを入力します。1-9 ページの「WebLogic Integration ユーザおよびパスワード」を参照してください。

4. [サーバ URL] フィールドで、WebLogic Integration を実行しているシステムを次のように指定します。

```
t3://host:port
```

ここで *host* は、システムのコンピュータ名または IP アドレスです。また、*port* は、コンフィグレーションされたリスポートです。ローカルコンピュータで WebLogic Integration が実行されている場合は、localhost または 127.0.0.1 を指定します。

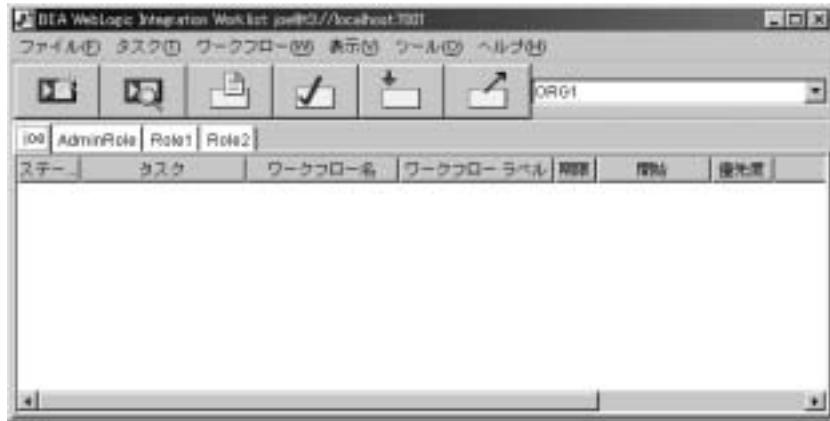
たとえば、サーバがローカルコンピュータ上で実行され、ポート 7001 でリスンするようにコンフィグレーションされている場合、次のように入力します。

```
t3://localhost:7001
```

5. [OK] をクリックします。

接続すると、次の図のように Worklist アプリケーション ウィンドウが表示されます。

図 2-9 Worklist アプリケーション ウィンドウ



Worklist の使用方法の学習

Worklist の使用方法を学習するには、『*WebLogic Integration Worklist ユーザーズ ガイド*』を参照してください。

Format Builder の起動

Format Builder を使用すると、ユーザの組織の他のシステムで使用されているバイナリ データの内容と構造を記述する MFL (Message Format Language) ファイル (メッセージフォーマット) を作成できます。これらのメッセージフォーマットは、Data Integration プラグインと組み合わせて、プロセス エンジンでバイナリ データと XML のやり取りをするのに使用されます。この変換機能は、システムとの間でデータをやり取りするアクションをワークフローに組み込むタスクを単純化します。

Format Builder を起動する手順は次のとおりです。

- 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 以下の各プラットフォームに対応した手順を実行して Format Builder を起動します。

- Windows:

[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Format Builder] を選択します。

fb.cmd コマンドを実行すると、コマンド ウィンドウが短い間だけ表示されます。しばらくしてから、Format Builder アプリケーション ウィンドウが表示されます。

- UNIX:

WLI_HOME/bin ディレクトリに移動し、fb コマンドを実行します。

たとえば、/home/bean/weblogic700/integration ディレクトリに WebLogic Integration をインストールした場合は、次のように入力します。

```
cd /home/bean/weblogic700/integration/bin
. ./fb
```

Format Builder アプリケーション ウィンドウが表示されます。

3. [Repository | LogIn] を選択します。

次の図に示すログオン ダイアログ ボックスが表示されます。

図 2-10 Format Builder ログイン ダイアログ ボックス



4. ユーザ名とパスワードを入力します。XML リポジトリに対するユーザ名とパスワードが割り当てられていない場合、デフォルトのユーザ名とパスワードを入力します。1-9 ページの「WebLogic Integration ユーザおよびパスワード」を参照。
5. [サーバ URL] フィールドで、WebLogic Integration を実行しているシステムを次のように指定します。

```
t3://host:port
```

ここで *host* は、システムのコンピュータ名または IP アドレスです。また、*port* は、サーバ用にコンフィグレーションされたリスンポートです。

WebLogic Integration がローカル コンピュータで実行されている場合は、localhost または 127.0.0.1 を指定します。

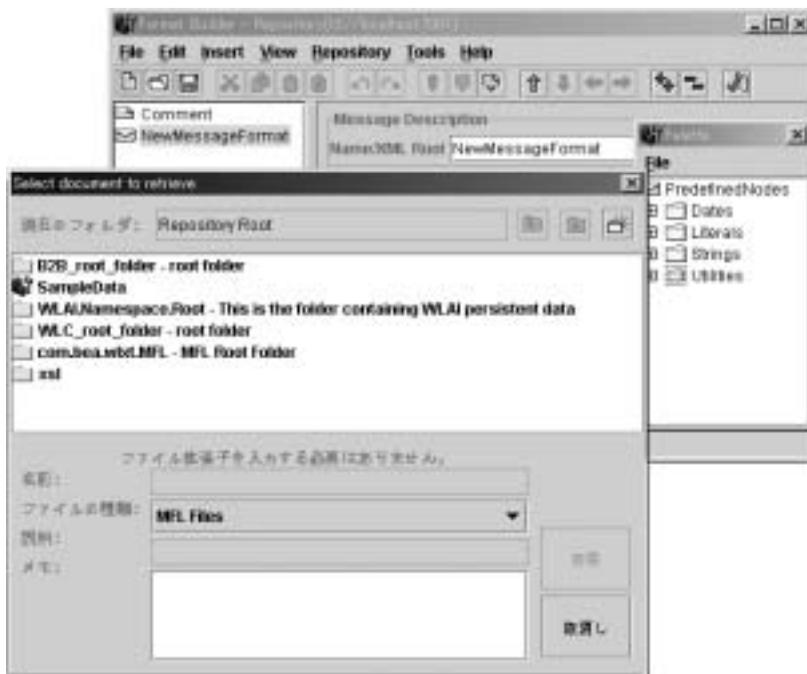
たとえば、サーバがローカル コンピュータ上で実行され、ポート 7001 でリスンするようにコンフィグレーションされている場合、次のように入力します。

```
t3://localhost:7001
```

6. [接続] をクリックします。

接続すると、アプリケーションウィンドウが表示されます。新しいメッセージフォーマットを作成することもできますし、既存のフォーマットを検索または修正することもできます。たとえば、XML リポジトリから項目を検索するには、[Repository | Retrieve] を選択します。次の図に示す検索ダイアログボックスが表示されます。

図 2-11 Format Builder の検索ダイアログボックス



Format Builder の使用方法の学習

Format Builder の使用法を学習するには、『WebLogic Integration データ変換』を参照してください。

Format Builder を使用する際の詳細については、Format Builder 付属のオンラインヘルプでも説明します。ヘルプを表示するには、[ヘルプ | ヘルプ トピック] を選択します。

B2B Console の起動

WebLogic Integration B2B Console では、以下のことを行うことができます。

- B2B エンジンの設定、トレーディング パートナ、会話の定義、コラボレーション アグリーメント、ビジネス プロトコル定義、およびロジック プラグインのコンフィグレーション
- コンフィグレーションされた要素のエクスポートおよびインポート
- B2B エンジン、トレーディング パートナ セッション、会話、コラボレーション アグリーメントのモニタ
- B2B エンジンの起動と終了

WebLogic Integration B2B Console を起動する手順は次のとおりです。

1. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 以下のいずれか 1 つを実行します。

- Windows または UNIX システムでは、Web ブラウザで次の URL を開きます。

```
http://host:port/b2bconsole
```

この URL では、*host* は WebLogic Integration が実行されているシステムのコンピュータ名または IP アドレス、*port* は、そのサーバ用にコンフィグレーションされたリスンポートです。WebLogic Integration が実行されているシステムは特に限定されていません。ローカル コンピュータで実行されている場合は、localhost または 127.0.0.1 を指定します。

たとえば、管理サーバがローカル コンピュータ上で実行され、ポート 7001 でリスンするようにコンフィグレーションされている場合、次のように入力します。

```
http://localhost:7001/b2bconsole
```

- Windows システムでは、WebLogic Integration がローカル コンピュータで実行されている場合、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | B2B Console] の順に選択します。

このショートカットは、ローカルシステムで実行されているサーバのコンソールを起動します。

3. プロンプトが表示されたら、サーバのユーザ名とパスワードを入力します。

注意： デフォルトのユーザ名とパスワードについては、1-9 ページの「WebLogic Integration ユーザおよびパスワード」を参照してください。

B2B Console は、次のように Web ブラウザに表示されます。

図 2-12 B2B Console の初期ビュー



注意： すぐにアクセスできるようにするために、B2B Console をブックマークします。

注意： ナビゲーション ツリー（左ペイン）で表示される項目は、B2B の設定によって決まります。たとえば、[拡張コンフィグレーション コントロールを表示] オプションが無効になっている場合、[ロジックプラグイン] は表示されません。B2B の基本設定については、『*B2B Integration Administration Console* オンライン ヘルプ』の「B2B Integration のコンフィグレーション」の「[プリファレンス] の設定」を参照してください。

B2B Console の使用方法の学習

WebLogic Integration に付属の B2B Integration 機能のコンフィグレーション要件の概要、B2B Console ツアー、B2B リポジトリの詳細については、『*B2B Integration 管理ガイド*』を参照してください。

各ページのフィールドおよびオプションの説明を含む B2B Console の使用の詳細は、オンライン ヘルプで提供されています。ヘルプについては、必要に応じて B2B Console のページの右上隅にある疑問符をクリックしてください。

注意： オンライン ヘルプの情報は、『*B2B Integration Administration Console* オンライン ヘルプ』というマニュアルでも参照できます。

B2B Console での B2B エンジンの起動と終了

ここでは、B2B Console で B2B エンジンをシャットダウンおよび再起動する手順について説明します。

B2B Console で B2B エンジンをシャットダウンまたは再起動しても、影響を受けるのは B2B エンジンだけです。B2B Console Web アプリケーション自体を内蔵する Web サーバも含めて、他の WebLogic Integration アプリケーションはすべて動作を続けます。

B2B Console での B2B エンジンの終了

WebLogic Integration アプリケーションおよびサービスを実行したままで、B2B エンジンをシャットダウンする手順は次のとおりです。

1. ナビゲーション ツリーから [B2B] を選択して B2B ページを表示します。
2. 次の図のように、最初に上位レベルの [モニタ] タブを選択し、その下の [一般] タブを選択します。

図 2-13 B2B の [モニタ] タブの [一般] タブ



3. [このサーバをシャットダウン] を選択します。
シャットダウン確認ページが表示されます。

図 2-14 シャットダウン確認ページ



4. 表 2-2 の情報を参照して、適切なシャットダウン モードを選択し、**B2B** エンジンをシャットダウンしてもよいかという確認メッセージが表示されたら、[はい] をクリックします。

表 2-2 シャットダウン モード

選択肢	説明
[即時]	すべてのアクティビティを直ちに停止する。
[停止]	B2B エンジンをシャットダウンし、トレーディング パートナ セッションのシャットダウンをトリガする。これにより、すべての会話が終了し、すべてのメッセージ キューが削除される。 注意： B2B エンジンをシャットダウンした後、BulkLoader ユーティリティまたは B2B Console を使用して、リポジトリ データをインポートしたい場合は、[停止] を選択してシャットダウンを行ってください。

5. B2B エンジンがシャットダウンされます。WebLogic Server および他のすべての WebLogic Integration アプリケーションおよびリソースは、アクティブなままです。B2B Console で B2B エンジンをシャットダウンしたら、次の手順に従って再起動できます。

B2B Console での B2B エンジンの再起動

B2B Console で B2B エンジンをシャットダウンした後で再起動する手順は次のとおりです。

1. 次の図のように、最初に上位レベルの [モニタ] タブを選択し、その下の [一般] タブを選択します。

図 2-15 B2B の [モニタ] タブの [一般] タブ



2. [このサーバを起動] を選択します。
起動確認ページが表示されます。

図 2-16 起動確認ページ



3. [はい]をクリックします。

再起動プロセスには、数分かかることがあります。プロセスが完了したら、[モニタ]の[一般]タブに戻ります。

3 WebLogic Integration のカスタマイズ

以下の節では、WebLogic Integration のデフォルト インストールを変更する方法について説明します。

- WebLogic Integration コマンドとサポート ファイル
- ドメインに対する新しいデータベースのコンフィグレーション
- ドメインに対するデータベースの初期化
- データベース ウィザードの使用法
- WebLogic Server Administration Console からのデータベース コンフィグレーションの更新
- 新規ドメインの作成とカスタマイズ
- WebLogic Integration 環境の更新
- カスタム Java Message Service キューのコンフィグレーション
- ビジネス オペレーションに対する EJB と Java クラスのデプロイ
- NULL 変数をサポートする BPM のコンフィグレーション
- BPM セキュリティ モデルについて
- パスワードの更新
- メールセッションプロパティのカスタマイズ
- ドメインのルート ディレクトリ
- 代替文字セットの使用
- WebLogic Integration ポート番号の変更
- 非永続モードの B2B をデプロイする

クラスタ化など、高度なコンフィグレーション オプションについては、『WebLogic Integration ソリューションのデプロイメント』を参照してください。

WebLogic Integration コマンドとサポート ファイル

WebLogic Integration には、いくつかのコマンドとサポート ファイルが用意されています。コマンドの多くは、1-11 ページの「ドメイン コンフィグレーションの要件」で説明している、データベース ウィザードと、RunSamples スクリプトで使用されます。通常、このようなコマンドは単独では呼び出されませんが、コマンドを変更したり、コマンドを実行する際のトラブルを解決したりしなければならない場合があります。これらのコマンドの使用に習熟するために、付録 B 「WebLogic Integration コマンド」にコマンド リファレンスが掲載されています。

また、標準的な WebLogic Integration ドメインのコンフィグレーションと起動を制御するファイルのサンプルを、付録 A 「WebLogic Integration サンプル コンフィグレーション ファイル」で紹介します。

ドメインに対する新しいデータベースの コンフィグレーション

ドメインに対する新規データベースの指定にどの方法を使用するかについては、以下の条件を検討してください。

- 切り替えるデータベースが、サンプルドメイン用のものか、それともコンフィグレーション ウィザードを使用して作成したカスタムドメイン用のものか。

注意： WebLogic Integration ドメインの作成とカスタマイズについては、3-23 ページの「新規ドメインの作成とカスタマイズ」を参照してください。

- ドメインの現在のデータベースに格納されている WebLogic Integration リポジトリ データを保持する必要があるかどうか。

次の表では、それぞれの状態で使用する方法について説明します。

表 3-1 データベース更新方法

更新するデータベース	ユーザの目的	結果
カスタムドメイン	既存のデータベースのリポジトリ データを保持しない。	<ol style="list-style-type: none">1. 対象ドメインのデータベース ウィザードを起動する。2. [データベースの作成] オプションを選択する。このオプションでは、新しいデータベース アクセス情報を入力するよう要求し、データベースを初期化する。
	既存のデータベースのリポジトリ データを保持する。	<ol style="list-style-type: none">1. 必要な情報をエクスポートする。2. 対象ドメインのデータベース ウィザードを起動する。3. [データベースの作成] オプションを選択する。このオプションでは、新しいデータベース アクセス情報を入力するよう要求し、データベースを初期化する。4. 必要な情報をインポートする。 「注意」を参照。

表 3-1 データベース更新方法（続き）

更新するデータベース	ユーザの目的	結果
サンプルドメイン	既存のデータベースのリポジトリデータを保持しない。	<ol style="list-style-type: none"> 1. 対象ドメインのデータベース ウィザードを起動する。 2. [データベースの切り替え] オプションを選択してコンフィグレーションを更新する。 3. RunSamples コマンドを実行して、新しいサンプルデータベースを初期化する。
	既存のデータベースのリポジトリデータを保持する。	<ol style="list-style-type: none"> 1. 必要な情報をエクスポートする。 2. 対象ドメインのデータベース ウィザードを起動する。 3. [データベースの切り替え] オプションを選択してコンフィグレーションを更新する。 4. RunSamples コマンドを実行して、新しいサンプルデータベースを初期化する。 5. 必要な情報をインポートする。

注意

既存のドメインを保持する方法もあります。既存のドメインを保持するには、3-23 ページの「新規ドメインの作成とカスタマイズ」の説明に従って、新しいドメインのコンフィグレーションを行い、そのドメインで新規データベースを使用してください。次に、必要に応じてワークフロー パッケージまたは B2B コンフィグレーション要素をエクスポートし、新しいドメインにインポートします。不要になったら、古くなったドメインを削除できます。

次の表では、以上のタスクを実行するのに必要な情報があるトピックを示します。

表 3-2 データベース更新手順に関するトピック

実行するタスク	参照
データベース ウィザードの起動	3-6 ページの「データベース ウィザードの使用法」
RunSamples コマンドの実行	1-15 ページの「サンプルドメインのコンフィグレーションと起動」

表 3-2 データベース更新手順に関するトピック

実行するタスク	参照
B2B コンフィグレーション要素のエクスポートおよびインポート	『 <i>B2B Integration 管理ガイド</i> 』の「B2BIntegration コンポーネントのインポートとエクスポート」
ワークフローパッケージのエクスポートおよびインポート	『 <i>WebLogic Integration Studio ユーザーズガイド</i> 』の「ワークフローパッケージのインポートとエクスポート」

ドメインに対するデータベースの初期化

ドメインのデータベースの初期化に使用する方法は、データベースがサンプルドメイン用か、コンフィグレーションウィザードで作成されたカスタムドメイン用かによって異なります。

- サンプルドメイン用にデータベースを初期化する場合は、1-15 ページの「サンプルドメインのコンフィグレーションと起動」の説明に従って `RunSamples` コマンドでデータベースを初期化します。
- コンフィグレーションウィザードで作成したドメイン用のデータベースを初期化する場合は、データベースウィザードを使用します。[データベースの作成] オプションを選択し、データベースを作成し初期化してから、システムデータをデータベースにロードします。次の「データベースウィザードの使用法」を参照してください。

注意： ドメインの作成とカスタマイズについては、3-23 ページの「新規ドメインの作成とカスタマイズ」を参照してください。

`RunSamples` コマンドとデータベースウィザードで実行されるタスクの詳細については、1-11 ページの「ドメインコンフィグレーションの要件」を参照してください。

データベース ウィザードの使用法

WebLogic Integration データベース ウィザードは、1-14 ページの「ドメインに対するデータベースのコンフィグレーション」で説明されているように、データベース コンフィグレーション タスクを自動化します。各 WebLogic Integration ドメインには、ドメイン固有バージョンのコマンドがインストールされます。

以下の節では、データベース ウィザードによるドメインのデータベースの初期化、ドメインに対する新しいデータベースの指定、または WebLogic Integration 2.1 データベースから WebLogic Integration 7.0 への移行に必要な情報について説明します。

- データベース接続情報
- データベース ウィザード オプション
- データベース ウィザードの仕組み
- データベース コンフィグレーション モード
- グラフィック モードによるデータベース ウィザードの使用法
- コンソール モードによるデータベース ウィザードの使用法

データベース接続情報

次の表では、サポートされている各データベースへの接続に必要な情報を示します。データベース ウィザードでは、コンフィグレーション対象データベースへの接続に必要な値の入力が必要です。

表 3-3 データベース アクセス情報

データベースの種類	接続パラメータ	説明
PointBase	なし	PointBase データベースは、 <code>DOMAIN_HOME\dbInfo\cloudscape\db</code> ディレクトリ内にあるドメイン用に作成される。デフォルト アクセス パラメータが使用される。

表 3-3 データベース アクセス情報 (続き)

データベースの種類	接続パラメータ	説明
Oracle	[Server Hostname]	Oracle サーバのホストとなるシステム名
	[Server Port Number]	Oracle ポート番号 (デフォルトは 1521)
	[User]	ユーザ ID (スキーマ)
	[Password]	ユーザ パスワード
	[Oracle SID]	Oracle システム ID
	[Net Service Name]	tnsnames.ora または Names サーバで表示されるデータベース名
Microsoft SQL Server	[Hostname]	Microsoft SQL Server のホストとなるシステム名
	[Port]	Microsoft ポート番号 (デフォルトは 1433)
	[User]	アカウント ログイン名
	[Password]	アカウント パスワード
	[Database Name]	Microsoft SQL Server で定義されているデータベース名
	[Server Name]	クライアント ネットワーク ユーティリティの [一般] タブに表示されるサーバエリアス。エリアスがコンフィグレーションされていない場合は、Hostname が使用される。

表 3-3 データベース アクセス情報 (続き)

データベースの種類	接続パラメータ	説明
Sybase	[Hostname]	Sybase サーバのホストとなるシステム名
	[Port]	Sybase ポート番号 (デフォルトは 5000)
	[User]	アカウント ログイン名
	[Password]	アカウント パスワード
	[Database Name]	Sybase サーバで定義されているデータベース名
	[Server Name]	Sybase クライアント コンフィグレーションで割り当てられたサーバ名。この名前は、Directory Services Editor (DSEDIT1) で設定する。

データベース ウィザード オプション

データベース ウィザードには、以下のオプションがあります。

- [データベースの切り替え]
このオプションを選択すると、ドメインに対するデータベースを切り替えることができます。データベースを初期化するために呼び出されるコマンド (CreateDB および RunSamples など) で使用される環境変数は更新され、config.xml ファイルは、新しい設定を反映するように変更されます。このオプションは、データベースを初期化するわけではありません。データベースの初期化に備えて環境をコンフィグレーションするだけです。

- [データベースの作成]

このオプションを選択すると、現在指定されているデータベースを初期化するか、新しいデータベースに切り替えてから初期化することができます。

注意： このオプションは、サンプルドメインでも使用できますが、サンプルデータベースの初期化に必要な全タスクを完了することができません。サンプルドメインでデータベースを初期化する場合は、1-15 ページの「サンプルドメインのコンフィグレーションと起動」の説明に従って `RunSamples` コマンドを実行する必要があります。

データベース ウィザードの仕組み

WebLogic Integration データベース ウィザードは、ドメインの `setDBVars` コマンドと `setDBVarsExt` コマンドで設定された環境変数を更新し、提供したデータベース アクセス情報を反映するように `config.xml` ファイルを変更し、選択したタスクの実行に必要なコマンドを呼び出します。

環境変数とコマンド ファイルの詳細については、付録 B 「WebLogic Integration コマンド」の `wlconfig` コマンドを参照してください。

データベース コンフィグレーション モード

WebLogic Integration データベース ウィザードは、WebLogic Integration インストールプログラムと同様に、以下のモードをサポートしています。

- グラフィック モード

Windows バージョンおよび UNIX バージョンのウィザードは、対話型のグラフィカル モードを提供します。このモードでは、グラフィック（ウィンドウ）ターミナルまたはワークステーションが必要です。

- コンソール モード

UNIX バージョンのウィザードでは、対話型のテキストのみのモードを使用して、グラフィック ターミナルまたはワークステーションのないシステムに対して、ドメイン用のデータベースをコンフィグレーションできます。

以下の節の説明に従って、システムに対応した手順を実行します。

- グラフィック モードによるデータベース ウィザードの使用法

- コンソール モードによるデータベース ウィザードの使用法

グラフィック モードによるデータベース ウィザードの使用法

次の手順では、WebLogic Integration データベース ウィザードを、グラフィックモードで使用してデータベース コンフィグレーションを更新する方法について説明します。

ドメインに対してデータベースをコンフィグレーションする手順は次のとおりです。

1. UNIX プラットフォームを使用する場合は、手順 2 に進んでください。
Windows プラットフォームを使用する場合は、次の表の該当する手順を完了してください。

データベース コンフィグレーションの対象	選択
WebLogic Integration サンプル ドメイン	[スタート プログラム BEA WebLogic Platform 7.0 WebLogic Integration 7.0 Integration Examples Integration Database Wizard]
コンフィグレーション ウィザード で作成された ドメイン	[スタート プログラム BEA WebLogic Platform 7.0 User Projects <i>domain</i> Integration Database Wizard]

[コンフィグレーション オプションの選択] ダイアログ ボックスが表示されます。手順 3 に進みます。

2. UNIX プラットフォームでは、次の手順を実行します。
 - a. `DOMAIN_HOME` ディレクトリに移動します。

たとえば、/home/bea/user_projects ディレクトリに、mydomain を作成した場合、次のように入力します。

```
cd /home/bea/user_projects/mydomain
```

- b. 次のコマンドを実行します。

```
wliconfig
```

次の図に示す [コンフィグレーション オプションの選択] ダイアログ ボックスが表示されます。

図 3-1 [コンフィグレーション オプションの選択] ダイアログ ボックス

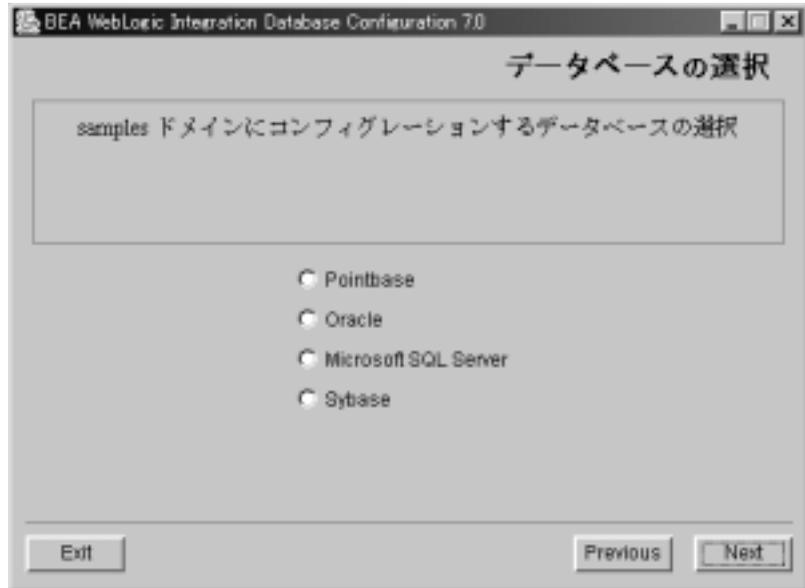


注意： UNIX 用の Microsoft SQL クライアントがないので、UNIX システムで Microsoft SQL Server データベースに接続している場合、ウィザードでデータベースを初期化することはできません。Windows システムからデータベースを初期化する必要があります。

3. オプションを選択します。3-8 ページの「データベース ウィザード オプション」を参照。
4. [Next] をクリックします。

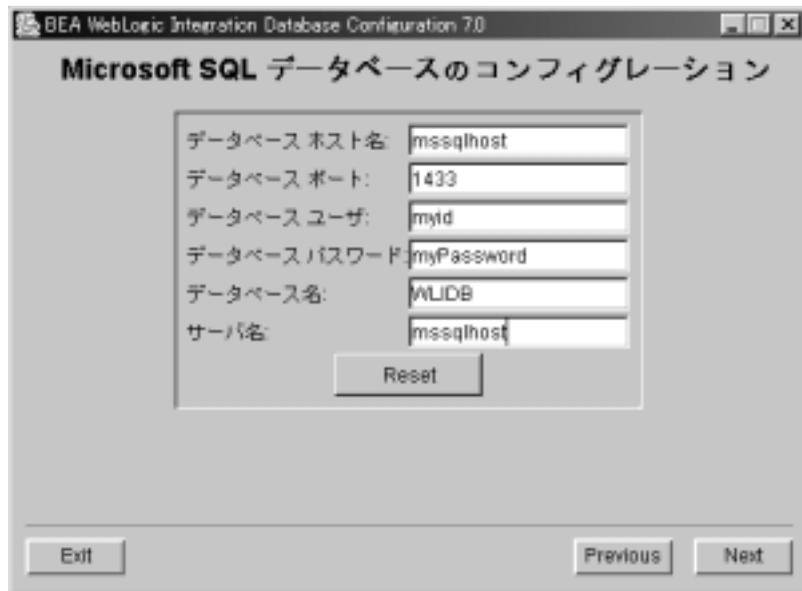
[データベースの選択] ダイアログ ボックスが表示されます。このダイアログ ボックスは、ドメイン用に現在コンフィグレーションされているデータベースを反映しています。次の図に示すように、ドメインを作成した際に、Pointbase データベースの接続パラメータを指定した場合は、Pointbase が選択されます。

図 3-2 データベース選択ダイアログ ボックス



5. 以下のいずれか 1 つを実行します。
 - 現在の設定でよければ、[Next] をクリックします。
 - 新しいデータベースに変更するには、目的のデータベースを選択してから [Next] をクリックします。
 - 次の図に示すように、データベース コンフィグレーション ダイアログ ボックスが表示されます。ダイアログ ボックスは、ドメインを作成したとき、またはウィザードで最後に新しいデータベースを指定したときに提供した情報を反映します。

図 3-3 データベース コンフィグレーション ダイアログ ボックス



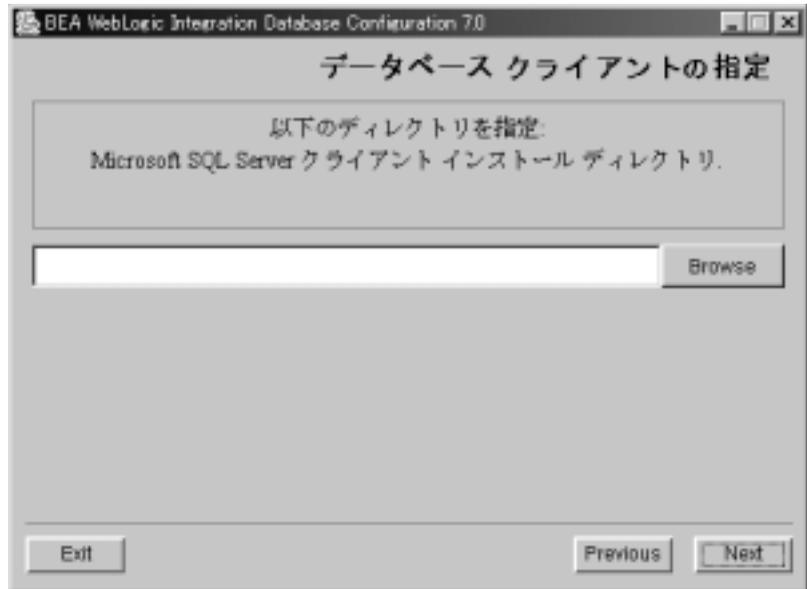
6. 以下のいずれか 1 つを実行します。

- 現在の値でよければ、[Next] をクリックします。
- 新しいデータベースに変更するには、データベースへの接続に必要な情報を入力してから [Next] をクリックします。各データベース タイプで必要な情報の概要については、3-6 ページの「データベース接続情報」を参照してください。

結果は次のようになります。

- [データベースの切り替え] を選択した場合、選択した操作が続行されます。変更が完了すると、図 3-7 に示す [変更の成功] ダイアログ ボックスが表示されます。手順 12 に進みます。
- [データベースの作成] を選択した場合は、次の図に示すようにデータベースのクライアント インストール ディレクトリを指定するよう要求されます。

図 3-4 [データベース クライアントの指定] ダイアログ ボックス



7. [Browse] をクリックして、[開く] ダイアログ ボックスを表示します。
8. データベースのクライアント インストール ディレクトリを検索します。たとえば、Microsoft SQL Server 7.0 クライアントの場合、次の図のように MSQL7 ディレクトリを検索します。(この例では、データベース クライアント インストールディレクトリは、c:\Program Files\MSSQL7 です)。

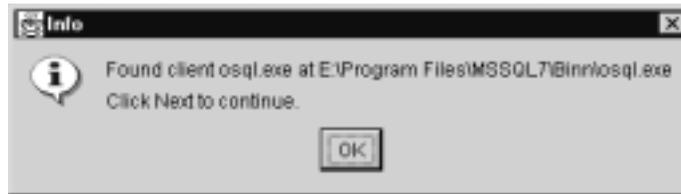
図 3-5 [開く] ダイアログ ボックス



9. [開く] をクリックします。

次の図のように、適切なクライアント アプリケーションが見つかったという確認メッセージが表示されます。

図 3-6 確認メッセージ



注意： クライアント アプリケーションが見つからなかったというエラーメッセージが表示されたら、[OK] をクリックしてメッセージを閉じ、[参照] をクリックして正しいディレクトリを検索します。

10. [OK] をクリックしてメッセージを閉じ、[データベース クライアントの指定] ダイアログ ボックスに戻ります。
11. [Next] を押して続きます。

変更が完了すると、次の図に示す [変更の成功] ダイアログ ボックスが表示されます。

図 3-7 [変更の成功] ダイアログ ボックス



注意： 選択した操作でエラーが発生した場合は、[エラー] ダイアログ ボックスが表示されます。このダイアログ ボックスは、ログ ファイルの場所を示します。ログ ファイルを参照し、問題に対処する方法を検討してください。

12. [Finish] をクリックしてインストールプログラムを終了します。

コンソール モードによるデータベース ウィザードの使用法

この節では、グラフィカルなインストール手順をエミュレートする一連の手順である、コンソール モードによるデータベースのコンフィグレーション手順を説明します。

コンソール モードでデータベース ウィザードを起動するには、プロンプトで次のコマンドを入力します。

```
cd DOMAIN_HOME
wliconfig console
```

次のリストでは、3-8 ページの「データベース ウィザード オプション」で説明されている [データベースの作成] オプションのコンソールモード プロンプトおよび応答を示します (応答は太字で示されています)。

表示されるプロンプトは、「Please Select Configuration Option:」および「Please specify the database type to configure:」プロンプトに対する応答によって、例とは異なります。

コード リスト 3-1 コンソール モードによるデータベース ウィザードの手順

```
=====
BEA WebLogic Integration Database Wizard 7.0
-----

=====
Configuration Selection
-----

    1- Switch Database
    2- Create Database

Please Select Configuration Option: 2

=====
Database Selection
-----

Please specify the database type to configure.

    1- Oracle
   ->2- Microsoft SQL Server
    3- Sybase

Select a number or <Enter> for default(2): 1
Oracle Server Hostname: oraclehost
Oracle Server Port Number: 1521
Oracle User: myuser
Oracle Password: mypassword
Oracle SID: nj908
Oracle Net Service Name: oraclehost.world
```

```
Please enter the location of your Oracle client installation: c:\oracle
Searching for Oracle client(Sqlplus.exe)...
```

```
Found client Sqlplus.exe at c:\oracle\sqlplus.exe
```

- >1- Use Located Client
- 2- Choose new Location

```
Please choose a number(1): 1
```

```
=====
Switching Database please wait...
-----
```

```
=====
Creating Database please wait...
-----
```

```
=====
Finalizing changes, please wait...
-----
```

```
Your configuration changes were successful. Press Enter to exit the Configuration
Utility.
```

WebLogic Server Administration Console からのデータベース コンフィグレーション の更新

WebLogic Integration ドメインに対するデータベース コンフィグレーションの更新にはデータベース ウィザードを使用しますが、WebLogic Server Administration Console からコンフィグレーションを更新しなければならない場合もあります。ここでは、その手順について説明します。

この方法で新しいデータベースに変更する場合は、データベースに対してテーブルを作成したり、データを入れたりするのに、ウィザード、`createDB` コマンド、または `RunSamples` コマンドを使用することはできなくなります。これらのコマンドは、`DOMAIN_HOME/dbInfo` ディレクトリに格納されている一部の環境変数およびデータベース アクセス情報に依存します。この情報を手動で更新しない限り、ウィザード、`createDB` コマンド、および `RunSamples` コマンドは、新しいデータベースで動作しません。

WebLogic Server Administration Console でデータベース コンフィグレーションを更新する場合、`WLI_HOME/dbscripts` ディレクトリにあるデータベース固有の SQL スクリプトを実行すると、テーブルを作成できます。テーブルを作成したら、Bulk Loader または B2B Console を使用して、`WLI_HOME/dbscripts/SystemRepData.xml` ファイルのシステム データをインポートできます。

JDBC 接続プールの更新

WebLogic Integration JDBC 接続プールを更新する手順は次のとおりです。

1. 2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って、WebLogic Server Administration Console を起動します。
2. ナビゲーション ツリーで、[サービス | JDBC | 接続プール | wlipool] を選択します。
3. 上位レベルの [コンフィグレーション] タブを選択します。
4. 表示が無い場合は、その下の [一般] タブを選択します。
5. [URL]、[ドライバクラス名]、[プロパティ] を必要に応じて編集し、JDBC 接続プールの設定をカスタマイズします。詳細については、3-21 ページの「データベース アクセス パラメータ」を参照してください。
6. [適用] をクリックして変更を保存します。
7. WebLogic Server Administration Console を閉じます。
8. WebLogic Integration をシャットダウンして再起動し、新しい設定を有効にします。

RDBMS レルム プロパティの更新

デフォルトでは、RDBMS レルムは、サンプルドメインやコンフィグレーションウィザードで作成したカスタムドメイン用にはコンフィグレーションされません。ドメイン用に RDBMS レルムのコンフィグレーションを行うには、config.xml ファイルに要素を手動で追加する必要があります。手順については、『*BEA WebLogic Integration 移行ガイド*』の「WebLogic Integration 2.1 から WebLogic Integration 7.0 への移行」の「RDBMS レルムからの移行」を参照してください。事前に、RDBMS レルムのコンフィグレーションを行わないと、RDBMS レルムのプロパティを更新できません。

RDBMS レルム プロパティを更新する手順は次のとおりです。

1. 2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って、WebLogic Server Administration Console を起動します。
2. ナビゲーションツリーで、[セキュリティ | レルム | wlpIRDBMSRealm] を選択します。
3. 上位レベルの [コンフィグレーション] タブを選択します。
4. 下部の [データベース] タブを選択します。
5. [ドライバ]、[URL]、[ユーザ名]、[パスワード] を必要に応じて編集し、データベースの設定をカスタマイズします。詳細については、次の「データベース アクセス パラメータ」を参照してください。
6. [適用] をクリックして変更を保存します。
7. WebLogic Server Administration Console を閉じます。
8. WebLogic Integration をシャットダウンして再起動し、新しい設定を有効にします。

データベース アクセス パラメータ

次の表では、WebLogic Server Administration Console による JDBC 接続プールのコンフィグレーションに必要なデータベース アクセス パラメータ情報の概要を示します。

表 3-4 JDBC 接続プール パラメータ

パラメータ	説明
[JDBC Driver]	データベースの接続に使用する JDBC ドライバ <ul style="list-style-type: none">■ Oracle Thin ドライバを使用している場合に入力するコマンド： <code>oracle.jdbc.driver.OracleDriver</code>■ Sybase jConnect ドライバを使用している場合に入力するコマンド： <code>com.sybase.jdbc.SybDriver</code>■ PointBase ドライバを使用している場合に入力するコマンド： <code>com.pointbase.jdbc.jdbcUniversalDriver</code>■ WebLogic jDriver for SQL Server ドライバを使用している場合に入力するコマンド： <code>weblogic.jdbc.mssqlserver4.Driver</code>
[Database User]	データベース サーバへの接続に必要なアカウント ログイン名
[Database Password]	データベース サーバへの接続に必要なパスワード
[URL]	JDBC ドライバのマニュアルで指定されているデータベースの URL。JDBC 接続プールの URL の形式については、次節を参照。

データベース アクセス URL の形式

各 JDBC 接続プール URL には、以下の情報が含まれます。

- *host* は、データベース サーバ ホスト名。
- *port* は、データベース サーバへの接続に使用するポート。
- *database* は、データベース サーバでリポジトリ テーブルを格納しているデータベース。

次の表では、サポートされている各データベースに対するサンプル URL を示しています。

対象ドライバ	URL
Oracle Thin ドライバ	<code>jdbc:oracle:thin:@host:port:database</code> たとえば、 <code>jdbc:oracle:thin:@rdbmshost:1521:wlidb</code>
WebLogic jDriver for Microsoft SQL Server	<code>jdbc:weblogic:mssqlserver4:database@host:port?sql7=true</code> たとえば、 <code>jdbc:weblogic:mssqlserver4:wlidb@rdbmshost:1433?sql7=true</code>
Sybase jConnect ドライバ	<code>jdbc:sybase:Tds:host:5000/database</code> たとえば、 <code>jdbc:sybase:Tds:rdbmshost:5000/wlidb</code>
PointBase ドライバ	<code>jdbc:pointbase://host:port/database</code> たとえば、 <code>jdbc:pointbase://localhost:9092/WLIDB</code>

新規ドメインの作成とカスタマイズ

この節では、WebLogic Integration ドメインの作成とカスタマイズに関する基本手順について説明します。詳細については、次のマニュアル参照してください。

- 『WebLogic Integration ソリューションのデプロイメント』
- 『コンフィグレーション ウィザードの使い方』
- 『コンフィグレーション ウィザード テンプレート リファレンス』

ドメイン作成の準備

新しいドメインを作成する前に、必ず完了しておかなくてはならないタスクを次に示します。

- ご使用のアプリケーションの要件に最適な、コンフィグレーション テンプレートの確認

一連のドメイン コンフィグレーション テンプレートは、WebLogic Platform インストールプログラムに含まれています。コンフィグレーション ウィザードでは、これらのテンプレートを使用して、新規 WebLogic Server ドメインを作成します。ご使用のアプリケーションの要件に最適なコンフィグレーション テンプレートを決定するには、1-5 ページの「WebLogic Integration コンフィグレーション テンプレート」および『コンフィグレーション ウィザード テンプレート リファレンス』を参照してください。

注意： 必ず、WebLogic Integration の機能をサポートするテンプレートを選択してください。このタイプのテンプレートを使用すれば、確実に、互換モードの WebLogic Server 6.x ファイル ベースのレルム (FileRealm) に基づくドメインを作成できます。WebLogic Integration では、新しい WebLogic Server 7.0 LDAP ベースのレルムはサポートされません。WebLogic Server または WebLogic Portal テンプレートを選択して新規ドメインを作成すると、新規ドメインでは、WebLogic Server 7.0 LDAP ベースのレルムが使用されます。WebLogic Integration 用の RDBMS レルムのコンフィグレーション手順については、『BEA WebLogic Integration 移行ガイド』の

「WebLogic Integration 2.1 から WebLogic Integration 7.0 への移行」の「RDBMS レルムからの移行」を参照してください。

- 「[コンフィグレーション ウィザードの使い方](#)」を参照して、入力する必要がある情報を確認する。
- 対象のドメインが、単一サーバ（スタンドアロン）構成で使用されるのか複数サーバ コンフィグレーションで使用されるのかを確認します。単一サーバ（スタンドアロン）コンフィグレーションで新規ドメインを作成する手順は次のとおりです。複数サーバ コンフィグレーションでの作成手順については、『[WebLogic Integration ソリューションのデプロイメント](#)』の「[クラスタデプロイメントのコンフィグレーション](#)」の「[手順 2. WebLogic Integration ドメインの作成](#)」を参照してください。

単一サーバ用ドメインの作成およびカスタマイズ手順

単一サーバ用に WebLogic Integration ドメインを作成してカスタマイズする手順は次のとおりです。

1. ご使用のプラットフォームに合わせて適切な手順を実行し、コンフィグレーション ウィザードを起動します。
- Windows システムでは、次の 3 つの方法があります。

- メニューからコンフィグレーション ウィザードを起動する手順は次のとおりです。

[スタート | プログラム | BEA WebLogic Platform 7. 0 | Domain Configuration Wizard] を選択します。

- コマンド ラインからコンフィグレーション ウィザードをグラフィックモードで起動するには、次のように入力します。

```
cd c:\bea\weblogic700\integration
setenv.cmd
cd c:\bea\weblogic700\common\bin
dmwiz
```

- コンフィグレーション ウィザードをコンソールまたはテキスト ベースモードで起動するには、次のように入力します。

```
cd c:\bea\weblogic700\integration
setenv.cmd
cd c:\bea\weblogic700\common\bin
dmwiz console
```

- UNIX システムでは、次の 2 つの方法があります。
 - コンフィグレーション ウィザードをグラフィック モードで起動するには、次のように入力します。

```
cd /home/joe/bea/weblogic700/integration
. setenv.sh
cd /home/joe/bea/weblogic700/common/bin
dmwiz.sh
```

- コンフィグレーション ウィザードをコンソールまたはテキスト ベース モードで起動するには、次のように入力します。

```
cd /home/joe/bea/weblogic700/integration
. setenv.sh
cd /home/joe/bea/weblogic700/common/bin
dmwiz.sh console
```

[ドメインのタイプと名前を選択] ウィンドウが表示されます。

詳細については、次の URL から **BEA WebLogic Platform** ドキュメント内の『**コンフィグレーション ウィザードの使い方**』を参照してください。

<http://edocs.beasys.co.jp/e-docs/platform/docs70/configwiz/index.html>

2. [ドメインのタイプと名前を選択] ダイアログ ボックスで、ご使用のアプリケーションに必要な **WebLogic Integration** コンポーネントを含むドメイン テンプレートを選択します。
3. [サーバタイプを選択] ダイアログ ボックスで、[**Single Server (Standalone Server)**] を選択します。
4. プロンプトが表示されたら、引き続き新規ドメインの作成に必要な情報を入力します。プロセスが完了したら、[コンフィグレーション ウィザードが完了しました] ダイアログ ボックスで [コンフィグレーション ウィザードを終了します] を選択し、次に、[**Done**] ボタンをクリックしてコンフィグレーション ウィザードを終了します。

5. (省略可能) 生成される config.xml ファイルのコピーを保存します。生成される config.xml ファイルの各コメントには、コンフィグレーション情報が記載されます。この情報は、WebLogic Server のインスタンスが起動したとき、削除されます。config.xml ファイルのコピーを作成するには、ご使用のオペレーティング システムに合わせて、次のコマンドを入力します。

- Windows システムの場合

```
copy config.xml config.xml.backup
```

- UNIX システムの場合

```
cp config.xml config.xml.backup
```

6. 新規ドメインのデータベース接続情報のコンフィグレーションを行います。手順 1 ~ 4 で作成した新規ドメインで、WebLogic Integration データベースウィザード (wliconfig) を起動します。

たとえば、デフォルト位置に mydomain と名付けたドメインを作成した場合、ご使用のオペレーティング システムに合わせて次のコマンドを入力します。

- Windows システムの場合

```
cd %BEA_HOME%\user_projects\mydomain  
wliconfig
```

- UNIX システムの場合

```
cd $BEA_HOME/user_projects/mydomain  
wliconfig
```

[コンフィグレーション オプションの選択] ウィンドウが表示されます。

7. 引き続き、データベースのコンフィグレーションを行うのに必要な情報を入力します。詳細については、3-10 ページの「グラフィック モードによるデータベース ウィザードの使用法」を参照してください。

次の節では、WebLogic Integration ドメインの作成の際にインストールされるディレクトリおよびファイルについて詳しく説明します。

サポートされているディレクトリおよびファイル

次の表に、WebLogic Integration をサポートするドメインに含まれるディレクトリおよびファイルを示します。表 3-5 に示されるスクリプトおよびディレクトリは、すべてドメイン ディレクトリに配置されます。たとえば、

c:\bea\user_projects\mydomain ディレクトリのコンフィグレーション ウィザードを使用してドメインを作成した場合、アプリケーションのディレクトリパスは、c:\bea\user_projects\mydomain\applications になります。

注意： 特に指定しない限り、表の各項目は、WebLogic Integration をサポートするすべてのドメインに含まれます。

表 3-5 ドメインディレクトリの内容

ディレクトリ/ファイル	内容説明
applications ディレクトリ	<p>次のようなアプリケーション固有の情報。このディレクトリの内容は、ご使用のアプリケーションの要件により決まる。</p> <p>applications ディレクトリのセットアップの詳細については、次の URL から BEA WebLogic Server ドキュメンテーション内の『<i>Web アプリケーションのアセンブルとコンフィグレーション</i>』の「Web アプリケーションの基本事項」の「ディレクトリ構造」を参照。</p> <p>http://edocs.beasys.co.jp/e-docs/wls/docs70/webapp/basics.html</p>

表 3-5 ドメイン ディレクトリの内容 (続き)

ディレクトリ/ファイル	内容説明
dbInfo ディレクトリ	<p data-bbox="512 293 1251 347">データベースを作成して初期化する際に使用されるデータベース固有の情報</p> <p data-bbox="512 370 1251 518">注意： 各サブディレクトリのデータベース変数ファイルのコンフィグレーション情報を更新する必要がある場合は、データベースウィザードを使用して更新を行ってください。詳細については、3-6 ページの「データベース ウィザードの使用法」を参照してください。</p> <p data-bbox="512 532 1251 586">このディレクトリには、次のサブディレクトリが配置される (サポートされている各データベースに対応)。</p> <ul data-bbox="512 609 946 789" style="list-style-type: none"> ■ oracle: Oracle データベース ■ pointbase: PointBase データベース ■ mssql: Microsoft SQL データベース ■ sybase: Sybase データベース <p data-bbox="512 803 1251 857">それぞれのデータベース ディレクトリには、次のファイルが定義される。</p> <ul data-bbox="512 880 1200 963" style="list-style-type: none"> ■ setDBVars コマンドードメインの基本データベース変数 ■ setDBVarsExt コマンドードメインの拡張データベース変数 <p data-bbox="512 977 1251 1031">setDBVars および setDBVarsExt の詳細については、B-37 ページの「setDBVars setDBVarsExt」を参照。</p>
scripts ディレクトリ	<p data-bbox="512 1063 1251 1117">現在のデータベース設定を変更したり、ドメイン固有のファイルを処理したりするスクリプト。</p> <p data-bbox="512 1131 1251 1222">このディレクトリは、トップレベルのスクリプト (<code>WLI_HOME/lib/scripts/SwitchDB.xml</code>) を参照する単一の Ant コールを含む。</p> <p data-bbox="512 1237 1251 1352">このスクリプトは、データベース ウィザードが、switchdb コマンドを実行して、ドメイン用にコンフィグレーションされているデータベースを更新する際に呼び出される。switchdb コマンドの詳細については、B-50 ページの「switchdb」を参照。</p>

表 3-5 ドメインディレクトリの内容（続き）

ディレクトリ/ファイル	内容説明
config.xml ファイル	<p>WebLogic Server ドメインについて記述し、デプロイされているリソースを管理するコンフィグレーション情報。</p> <p>config.xml ファイルの内容の詳細については、A-2 ページの「config.xml」を参照。</p>
fileRealm.properties ファイル	<p>WebLogic Server が起動されるとき作成される User、Group、および ACL オブジェクトが保存されるデフォルトのセキュリティレム。</p> <p>このファイルの内容の詳細については、A-18 ページの「fileRealm.properties」を参照。BPM セキュリティモデルの詳細については、3-48 ページの「BPM セキュリティモデルについて」を参照。</p>
SerializedSystemIni.dat ファイル	<p>システムパスワードのハッシュ値を生成するために使用されるデータファイル。このファイルは、対応する fileRealm.properties ファイルに関連付けられる。</p> <p>このファイルの詳細については、『WebLogic Security の管理』の「WebLogic Security のコンフィグレーション」の「ユーザアカウントの保護」を参照。このガイドは、次の URL からアクセスできる BEA WebLogic Server ドキュメンテーション内にある。</p> <p>http://edocs.beasys.co.jp/e-docs/wls/docs70/secmanage/security7.html</p>

表 3-5 ドメイン ディレクトリの内容 (続き)

ディレクトリ/ファイル	内容説明
起動スクリプト たとえば、 startWeblogic.cmd (Windows) または startWeblogic.exe (UNIX)	<p>現在のドメインで WebLogic Server インスタンスの起動に使用するスクリプト。</p> <p>このスクリプトの機能は必要に応じて変更できるが、通常は、次のようなタスクを実行する。</p> <ul style="list-style-type: none"> ■ setenv コマンドを呼び出して環境設定する。 ■ 該当するコマンドを呼び出して、現在のドメインを設定する。 ■ setdbdata コマンドを使用して、データベース固有の変数を現在のプロセスにロードする。 ■ 古いログ ファイルを削除する。 ■ ドメイン内の WebLogic Server を起動し、config.xml ファイル (この表の前の項で説明済み) の指定に従って、WebLogic Integration アプリケーションとリソースをデプロイする Java コマンドを実行する。 <p>注意： 起動スクリプトおよびこのスクリプトで定義できるコマンドの詳細については、B-46 ページの「startWeblogic」を参照してください。</p> <p>注意： ご使用のアプリケーションで必要な場合は、コマンドライン オプションを更新したり他のコマンドライン オプションを追加したりすることができます。WebLogic Server コマンドライン オプションの詳細については、「管理サーバの起動」の「コマンドラインからの管理対象サーバの起動」を参照してください (『WebLogic Server 管理者ガイド』の「WebLogic Server の起動と停止」にあります)。このガイドは、次の URL からアクセスできる BEA WebLogic Server ドキュメンテーション内にあります。</p> <p>http://edocs.beasys.co.jp/e-docs/wls/docs70/adminguide/startstop.html</p>

表 3-5 ドメインディレクトリの内容（続き）

ディレクトリ/ファイル	内容説明
停止スクリプト たとえば、 stopWeblogic.cmd (Windows) または stopWeblogic.exe (UNIX)	<p>現在のドメインの WebLogic Server インスタンスを正常にシャットダウンするためのスクリプト。</p> <p>このスクリプトの機能は必要に応じて変更できるが、通常は、次のようなタスクを実行する。</p> <ul style="list-style-type: none"> ■ <code>setenv</code> コマンドを呼び出して環境設定する。 ■ ドメインで実行されている WebLogic Server インスタンスに対するシャットダウン要求を発行する。 <p>停止スクリプトおよびこのスクリプトで定義できるコマンドの詳細については、B-48 ページの「stopWeblogic」を参照。</p> <p>ご使用のアプリケーションでの必要な場合に、コマンドラインオプションを更新または追加できる。WebLogic Server コマンドラインオプションの詳細については、「管理サーバの起動」の「Starting the WebLogic Administration Server from the Command Line」を参照（『<i>WebLogic Server 管理者ガイド</i>』の「WebLogic Server の起動と停止」に掲載）。このガイドは、次の URL からアクセスできる BEA WebLogic Server ドキュメンテーション内にある。</p> <p>http://edocs.beasys.co.jp/e-docs/wls/docs70/adminguide/startstop.html</p>

表 3-5 ドメイン ディレクトリの内容 (続き)

ディレクトリ/ファイル	内容説明
他の WebLogic Integration コマンドのカスタムバージョン	<p>ご使用のアプリケーション環境内でカスタマイズできる WebLogic Integration コマンド。</p> <p>たとえば、ドメインの作成に使用したテンプレートによっては、次の WebLogic Integration コマンドのカスタムバージョンがドメインに含まれることがある。</p> <ul style="list-style-type: none">■ <code>checkdomain</code> - カスタムバージョンは、<code>setdomain</code> (echo 行) への呼び出しで現在のドメインを反映する。このコマンドは、<code>setdbdata</code> コマンドにより呼び出される。■ <code>setdbdata</code> - このコマンドのカスタマイズされたバージョンにより、<code>checkdomain</code> コマンドのカスタムバージョンが呼び出される。このコマンドは、<code>setDomainTypeData</code> コマンドにより呼び出される。■ <code>setDomainTypeData</code> - カスタムバージョンは、ドメイン固有の環境変数を設定し、<code>setdbdata</code> コマンドのカスタムバージョンを呼び出す。Application Integration 固有の変数を設定する場合に、このファイルを使用することをお勧めする。このコマンドは、この表の前の項で説明した、対応する起動スクリプトにより呼び出される。 <p>詳細については、B-1 ページの「WebLogic Integration コマンド」を参照。</p>

WebLogic Integration 環境の更新

WebLogic Integration で使用する環境変数は、`setEnv.cmd` ファイル (Windows) または `setenv.sh` ファイル (UNIX) で設定します。このファイルは、WebLogic Integration のインストールディレクトリ (`WLI_HOME`) にあります。以上の各ファイルの例は、付録 A 「WebLogic Integration サンプル コンフィグレーション ファイル」にあります。

`setEnv` ファイルの変数は、WebLogic Integration のインストール時に設定され、通常更新する必要はありません。ただし、その必要がある場合は、次の手順を実行すると環境を更新できます。

WebLogic Integration 環境を更新する手順は次のとおりです。

1. 以下のいずれか 1 つを実行します。

- Windows の場合

WebLogic Integration インストールディレクトリに移動して [setEnv.cmd] を右クリックし、ショートカット メニューから [編集] を選択してメモ帳でファイルを開きます。

- UNIX の場合

目的のドメイン ディレクトリに移動し、任意のテキスト エディタで setenv.sh を開きます。

2. 以下の変数を、使用している環境に対応した値に設定します。

- JAVA_HOME

JDK 1.3.1 をインストールした場所。たとえば、C:\bea\jdk131_03.

- BEA_HOME

BEA Home ディレクトリの場所。たとえば、C:\bea.

- WL_HOME

WebLogic プラットフォームをインストールした場所。たとえば、C:\bea\weblogic700\server.

- WLI_HOME

WebLogic Integration をインストールした場所。たとえば、C:\bea\weblogic700\integration.

- SAMPLES_HOME

WebLogic Integration サンプルの場所。たとえば、C:\bea\weblogic700\samples. 詳細については、付録 A 「WebLogic Integration サンプル コンフィグレーション ファイル」を参照してください。

3. 変更を保存し、ファイルを閉じます。

1-18 ページの「WebLogic Integration の起動」の説明に従って startWeblogic コマンドを実行すると、setEnv コマンドが呼び出され、環境変数が有効になります。

カスタム Java Message Service キューの コンフィグレーション

カスタム Java Message Service (JMS) キューを作成し、Message Driven Bean Generator ユーティリティを実行して、カスタム キューでリスンするデプロイ可能な Java Archive (JAR) ファイルを生成できます。カスタム キューをコンフィグレーションする手順は次のとおりです。

- WebLogic Server Administration Console で、選択したドメインに対するカスタム JMS キューを作成する。
- `mdbgenerator` ユーティリティを実行して、キューでリスンする MDB (Message-Driven Bean : メッセージ駆動型 Bean) を生成する。
- WebLogic Integration コンフィグレーションを更新して新しい MDB を認識する。

JMS キュー作成の詳細については、WebLogic Server 7.0 マニュアルを参照してください。

カスタム JMS キューを作成する手順は次のとおりです。

1. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って、WebLogic Server Administration Console を起動します。
3. ナビゲーションツリーで [JMS | サーバ | WLJMServer | 送り先のコンフィグレーション] を選択します。新しい JMSQueue を作成するには、[送り先のコンフィグレーション] を右クリックし [新しい JMSQueue のコンフィグレーション] を選択します。キューの名前と JNDI 名を指定します。ほかのフィールドをデフォルト設定にするか、WebLogic Server 7.0 マニュアルを参照してその他の値を設定します。
注意： 順序付けされたキューに対して優先順位を設定しないでください。
4. 1-21 ページの「WebLogic Integration の終了」の説明に従って、WebLogic Integration をシャットダウンします。

5. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を再起動します。
6. サーバが再び実行されたら、WebLogic Server Administration Console を再起動し、[JMS | サーバ | WLJMServer | 送り先のコンフィグレーション] を選択して、キューが作成されていることを確認します。

mdbgenerator ユーティリティを実行する手順は次のとおりです。

1. コマンド ウィンドウを開き、WebLogic Integration インストールディレクトリの bin サブディレクトリに移動します。
2. コマンド プロンプトで、次のように入力します。

```
mdbgenerator -queue queue_name [-min number] [-max number]
[-order number] [-transact] [-validate] [-timeout seconds] [-help]
```

次の表では、mdbgenerator コマンドのオプションを示します。

表 3-6 mdbgenerator コマンド ライン オプション

オプション	説明
-queue <i>queue_name</i>	MDB を生成するカスタム キュー名。このオプションは必須。
-min <i>number</i>	順序付けされていないリスナの最少数。
-max <i>number</i>	順序付けされていないリスナの最大数。
-order <i>number</i>	順序付けされたリスナの数。31 以下の素数でなければならない。
-transact	トランザクションを必須に設定する。
-validate	XML 検証を有効にする。

表 3-6 mdbgenerator コマンド ライン オプション

オプション	説明
<code>-timeout seconds</code>	トランザクション タイムアウトを設定する (秒単位)。このオプションは、デフォルトで、30 秒に設定されている。 この値は、 <code>transact</code> フラグが設定されていない場合にのみ使用される。 <code>transact</code> フラグが設定されている場合、ユーティリティは、WebLogic Server トランザクション タイムアウト値を使用する。この値は、Administration Console で設定でき、デフォルトでは 30。
<code>-help</code>	コマンドの用法を示す構文を表示する。

`queue_name-mdb-generator.jar` ファイルが `bin` ディレクトリに作成されます。

- 生成したファイルを、WebLogic Integration インストール ディレクトリの `lib` サブディレクトリに移動します。
- ドメインで実行されている場合は、1-21 ページの「WebLogic Integration の終了」の説明に従って、WebLogic Integration をシャットダウンします。
- ドメインの `config.xml` ファイルを開きます。
- 次の XML タグを追加します。

```
<Application Name="WebLogic Integration"
Path="c:/bea/weblogic700/integration/lib">
```

他のすべての `<EJB Component>` 要素の後に、次の行を追加します。

```
<EJBComponent Name="MyComponent" Targets="myserver"
URI="queue_name-mdb-generator.jar"/>
```

注意： `config.xml` ファイルでは、大文字と小文字を区別します。正しい形式でテキストを入力してください。

- `path/META-INF` ディレクトリにある `application.xml` ファイルを更新します。ここで `path` は、手順 6 に示すように (太字で強調)、`application` 要素の `Path` 属性により指定された位置を示します。たとえば、手順 6 では `application.xml` ファイルは、次のディレクトリに配置されます。

```
c:\bea\weblogic700\integration\lib\META-INF
```

J2EE コンポーネントのデプロイ順は、application.xml ファイルでのコンポーネントの順序により決定されます。

警告： application.xml ファイルで示されている WebLogic Integration コンポーネントを削除したりその順序を変更したりすることは避けてください。

application.xml ファイルの application 要素の終わり付近にある bpm-init-ejb.jar module 要素のすぐ前に、JAR ファイル用の <module> 要素および <ejb> 要素を追加します。たとえば、コード リスト 3-4 で太字で強調されているコード行を参照してください。

コード リスト 3-2 サンプルドメインの application.xml ファイル

```
<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application
1.2//EN" "http://java.sun.com/j2ee/dtds/application_1_2.dtd">
<application>
  <display-name>WebLogic Integration</display-name>
  .
  .
  .
  <module>
    <ejb>queue_name-mdb-generator.jar</ejb>
  </module>

  <!--BPM Initialization Bean must be deployed after BPM plug-ins-->
  <module>
    <ejb>bpm-init-ejb.jar</ejb>
  </module>
</application>
```

8. ファイルを保存します。
9. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を再起動します。

ビジネス オペレーションに対する EJB と Java クラスのデプロイ

EJB をデプロイして、関連付けられたビジネス オペレーションを WebLogic Integration で使用できるようにする手順は次のとおりです。

1. EJB JAR ファイルを `WLI_HOME/lib` ディレクトリにコピーまたは移動します。
2. EJB のデプロイメント記述子を WebLogic Integration application 要素に追加します。

EJB デプロイメント記述子ファイルの詳細については、次の URL から BEA WebLogic Server ドキュメンテーション内の『*WebLogic エンタープライズ Java Beans プログラマーズガイド*』の「WebLogic Server への EJB のデプロイ」を参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/ejb/deploy.html>

3. コンフィグレーション ファイル `config.xml` を手動で更新して、EJB を WebLogic Integration の application 要素に追加します。このファイルは、`DOMAIN_HOME` ディレクトリに配置されています。ここで、`DOMAIN_HOME` はドメインのルートフルパス名（たとえば、`c:\bea\user_projects\domain`）です。

この手順については、3-39 ページの「WebLogic Integration application 要素に EJB を追加する」を参照してください。WebLogic Integration application 要素の初期コンフィグレーションについては、図 A-3 で示します。（詳細については、付録 A 「WebLogic Integration サンプル コンフィグレーション ファイル」を参照してください）。

カスタム Java クラスをデプロイして、WebLogic Integration アプリケーションで使用できるようにするには、カスタム Java クラスを格納する JAR (Java Archive) ファイルを、`CLASSPATH` に追加する必要があります。この手順については、3-42 ページの「Java クラスの `CLASSPATH` への追加」を参照してください。

注意： *WebLogic Integration* アプリケーションという用語は、WebLogic Integration を使用して開発されたアプリケーションを意味します。この用語を `config.xml` ファイルの WebLogic Integration application 要素や `WLI application` 要素と混同しないでください。WebLogic

Integration application 要素は、WebLogic Integration を構成する J2EE コンポーネントの集まりを含みます。WebLogic Integration サンプル config.xml ファイルでは、WLI が、WebLogic Integration を構成する J2EE コンポーネントの集まりを含む application 要素の名前です。コンフィグレーションウィザードで生成された config.xml ファイルでは、WebLogic Integration が、WebLogic Integration を構成する J2EE コンポーネントの集まりを含む application 要素の名前です。

WebLogic Integration application 要素に EJB を追加する

EJB を WebLogic Integration application 要素に追加する手順は次のとおりです。

1. config.xml (*DOMAIN_HOME* ディレクトリにあるコンフィグレーションファイル) を更新して、WebLogic Integration application 要素に関連付けられたデプロイメント記述子ファイルを指定します。

EJB 記述子ファイルを指定するには、<EJBComponent> 要素を使用します。

次のコード リストは、サンプルドメインの config.xml ファイルから抜粋したものです。このコード リストには、MyEJB.jar という名前の EJB をデプロイするのに必要な情報が示されています。このファイルは、*SAMPLES_HOME/integration/config/samples* ディレクトリに配置されています。ここで、*SAMPLES_HOME* は、WebLogic Platform サンプルディレクトリを示します。重要なコード行は、太字で表示されています。

コード リスト 3-3 WebLogic Integration application 要素に EJB を追加する

```
<Application Name="WebLogic Integration"
  Path="c:/bea/weblogic700/integration/lib" TwoPhase="true">

  <!--Repository-->
  <EJBComponent Name="repository-ejb.jar" Targets="myserver"
    URI="repository-ejb.jar" />

  <!--B2B-->
  <EJBComponent Name="WLI-B2B Startup" Targets="myserver"
    URI="b2b-startup.jar" />
```

```
<WebAppComponent Name="TransportServlet" Targets="myserver"
  URI="b2b.war" />
<EJBComponent Name="WLI-B2B RN MDB" Targets="myserver"
  URI="b2b-rosettanet.jar" />
<WebAppComponent Name="b2bconsole" ServletReloadCheckSecs="1"
  Targets="myserver" URI="b2bconsole.war"/>

<!--AI-->
<EJBComponent Name="WLI-AI Server" Targets="myserver"
  URI="wlai-server-ejb.jar" />
<WebAppComponent Name="wlai" URI="wlai.war" Targets="myserver"/>
<EJBComponent Name="WLI-AI Async Processor" Targets="myserver"
  URI="wlai-asyncprocessor-ejb.jar" />
<EJBComponent Name="WLI-AI Event Processor" Targets="myserver"
  URI="wlai-eventprocessor-ejb.jar" />

<!--BPM-->
<EJBComponent Name="wlpi-ejb.jar" Targets="myserver" URI="wlpi-ejb.jar" />
<EJBComponent Name="wlpi-master-ejb.jar" Targets="myserver"
  URI="wlpi-master-ejb.jar" />
<EJBComponent Name="wlpi-mdb-ejb.jar" Targets="myserver"
  URI="wlpi-mdb-ejb.jar" />

<!--XT-->
<EJBComponent Name="wlxtpi.jar" Targets="myserver" URI="wlxtpi.jar" />

<!--Plugins-->
<EJBComponent Name="WLI-B2B RN Plugin for BPM" Targets="myserver"
  URI="wlc-wlpi-plugin.jar" />
<EJBComponent Name="WLI-B2B EBXML Plugin for BPM" Targets="myserver"
  URI="ebxml-bpm-plugin.jar" />
<EJBComponent Name="WLI-AI Plug-In for BPM" Targets="myserver"
  URI="wlai-plugin-ejb.jar" />
<WebAppComponent Name="WLAIPugin" URI="wlai-plugin.war" Targets="myserver"/>
<WebAppComponent Name="XTPlugin" Targets="myserver" URI="wlxtpi.war"/>

<!--Samples-->
<WebAppComponent Name="com.bea.wlpi.SamplePlugin" Targets=""
  URI="sampleplugin.war" />
<EJBComponent Name="sampleplugin-ejb.jar" Targets=""
  URI="sampleplugin-ejb.jar" />
<EJBComponent Name="wlxtejb.jar" Targets="" URI="WLXTEJB.jar" />
<EJBComponent Name="pobean.jar" Targets="myserver" URI="pobean.jar" />

<!--BPM Initialization-->
<EJBComponent Name="WLI-BPM Initialization" Targets="myserver"
  URI="bpm-init-ejb.jar" />
<EJBComponent Name="WLI Error Listener" Targets="myserver"
```

```
URI="wli-errorlistener-mdb.jar" />
</Application>
```

注意： WebLogic Integration の最後すなわち WLI の application 要素リスト (</Application> タグの前) に EJB を追加します。

警告： サンプル config.xml ファイルでは、WLI が、WebLogic Integration を構成する J2EE コンポーネントの集まりを含む application 要素の名前です。コンフィグレーション ウィザードで生成された config.xml ファイルでは、WebLogic Integration が、WebLogic Integration を構成する J2EE コンポーネントの集まりを含む application 要素の名前です。

config.xml ファイルの更新の詳細については、次の URL から BEA WebLogic Server ドキュメンテーション内の『*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。

http://edocs.beasys.co.jp/e-docs/wls/docs70/config_xml/index.html

2. application.xml ファイルを更新します。このファイルは、path/META-INF ディレクトリに配置されています。ここで、path は、コード リスト 3-3 で示す (太字で強調) ように、application 要素の Path 属性で指定された位置を意味します。コード リスト 3-3 のサンプルでは、application.xml ファイルは、次のディレクトリに配置されています。

```
c:\bea\weblogic700\integration\lib\META-INF
```

次のコード リストは、サンプルドメインの application.xml ファイルから抜粋したものです。このコード リストには、MyEJB.jar という名前の EJB をデプロイするのに必要な情報が示されています。

EJB JAR ファイルのデプロイ順は、application.xml ファイルでの EJB JAR ファイルの並び順によって決まります。複数の EJB JAR ファイルを追加するケースでは、EJB A が EJB B に依存する場合、必ず、EJB B を先にデプロイしてください。

警告： application.xml ファイルで示されている WebLogic Integration EJB JAR ファイルや WAR ファイルを削除したりその順序を変更したりすることは避けてください。

EJB JAR ファイルを指定するには、application.xml ファイルの application 要素の終わり付近にある bpm-init-ejb.jar module 要素のす

ぐ前に、各 EJB JAR ファイルの <module> 要素および <ejb> 要素を追加します。たとえば、コード リスト 3-4 で強調表示されているコード行を参照してください。

コード リスト 3-4 サンプルドメインの application.xml ファイル

```
<!DOCTYPE application PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application
 1.2//EN" "http://java.sun.com/j2ee/dtds/application_1_2.dtd">
<application>
  <display-name>WebLogic Integration</display-name>
  .
  .
  .
  <module>
    <ejb>MyEJB.jar</ejb>
  </module>

  <!--BPM Initialization Bean must be deployed after BPM plug-ins-->
  <module>
    <ejb>bpm-init-ejb.jar</ejb>
  </module>
</application>
```

3. 1-1 ページの「はじめに」の説明に従って、サーバを再起動します。

Java クラスの CLASSPATH への追加

以下のいずれかの方法で、カスタム Java クラスの JAR ファイルをサーバの CLASSPATH に追加できます。

- **JAR ファイルを setenv コマンド ファイルに追加する方法**
JAR ファイルを setEnv コマンド ファイルの WLISERVERCP 環境変数に追加すると、マシンにインストールされている任意の WebLogic Integration ドメインで使用できるようになります。startWeblogic を実行するたびに、setEnv コマンドが呼び出されてから、サーバを起動する Java コマンドが発行されます。

- **JAR ファイルを startWeblogic コマンド ファイルに追加する方法**
startWeblogic コマンド ファイルで指定した CLASSPATH に JAR ファイルを追加すると、そのコマンド ファイルで起動したサーバ上でのみ Java クラスを使用できるようになります。

JAR ファイルを setenv コマンド ファイルに追加する方法

JAR ファイルを setEnv コマンド ファイルに追加する手順は次のとおりです。

1. `WLI_HOME/setEnv.cmd` (Windows) または `WLI_HOME/setenv.sh` (UNIX) を任意のテキスト エディタで開きます。
2. 次の行を追加します。

```
set WLISERVERCP=%WLISERVERCP%;%JAVA_HOME%\lib\tools.jar
```
3. 絶対パス名をこの文の最後に追加します。セミコロンで、既存のエントリと新しいエントリを区切ります。
4. ファイルを保存して閉じます。

Java クラスをクライアント アプリケーションで使用できるようにするために、WebLogic Integration クライアントに対する CLASSPATH の設定に使用した WLICP 変数に Java クラスを追加することもできます。

注意： WLISERVERCP 環境変数は、startWeblogic コマンド ファイルに表示されないことがあります。ドメインごとに、setEnv コマンド ファイルの後で setDomainTypeData ファイルが呼び出されます。WLISERVERCP 環境変数の値は、このコマンド ファイルの SVRCP 環境変数に転送されます。

JAR ファイルを startWebLogic コマンド ファイルに追加する方法

JAR ファイルを startWeblogic コマンド ファイルに追加する手順は次のとおりです。

1. `startWeblogic.cmd` コマンド ファイル (Windows) または `startWeblogic` コマンド ファイル (UNIX) を任意のテキスト エディタで開きます。
2. サーバを起動する Java コマンドを見つけます。

3 WebLogic Integration のカスタマイズ

```
REM Start weblogic
%JAVA_HOME%\bin\java %DB_JVMARGS% -Xmx256m -classpath %SVRCP%...
```

3. サーバ起動コマンドのすぐ上に次のコマンドを追加します。

```
set SVRCP=%SVRCP%;MyJAR
```

ここでは、*MyJAR* は、Java クラスを格納する JAR ファイルの絶対パス名です。

4. ファイルを保存して閉じます。

NULL 変数をサポートする BPM のコンフィグレーション

startWeblogic コマンドの Java コマンドに次の行を追加することで、サーバ起動時に NULL 変数のサポートを有効にできます。

```
-Dwli.bpm.server.evaluator.supportsNull=true
```

サーバを起動する Java コマンドに追加するときに、このオプションは、BPM に対する NULL 変数サポートを有効にします。このオプションが false に設定されている場合、またはコマンドで指定されていない場合、NULL 変数サポートは無効です。

サーバがドメインで起動し、式が初めて評価されると、NULL 変数サポートのステータスを示すメッセージが表示されます。

- NULL サポートが有効になっている場合、「WebLogic Integration BPM server supports NULL」というメッセージが表示されます。
- NULL サポートが無効になっている場合、「WebLogic Integration BPM server does not support NULL」というメッセージが表示されます。

サーバは、式を初めて評価したときに一度だけこのメッセージを表示します。

以下の節では、NULL サポートが有効になっているとき、および無効になっているときの変数の初期値と式の評価の仕組みについて説明します。

wli.bpm.server.evaluator.supportsNull=true

wli.bpm.server.evaluator.supportsNull=true の場合は、以下の規則が適用されます。

- 値 NULL (大文字でも小文字でも可) は、null を発行する式で有効な定数となります。たとえば、ワークフロー変数 \$a 式 NULL を設定します。

- 演算子 = および <> を、NULL 値の比較に使用できます。その他の比較演算子 (>=, < ...) で他の NULL 値のものはすべて、**false** になります。

たとえば、ワークフロー変数 \$a および \$c に NULL 値があり、\$b に非 NULL 値がある場合は、次のようになります。

```
$a=NULL      true
$a=$c       true
$a<>NULL    false
$b<>NULL    true
$b>$a      false
```

- 算術演算式に NULL 値を使用することはできません。たとえば、\$a * NULL では、例外が発生します。
- 論理演算式に NULL 値を使用することはできません。たとえば、\$a XOR NULL では、例外が発生します。
- 文字列連結 (StrCat) では、NULL 値を使用できます。たとえば、"wli " + NULL では "wli null" という結果になります。
- 変数初期値に関して、変数には初期値が割り当てられません。つまり、すべての変数は、初期値として NULL を持ちます。これは、NULL サポートが無効の場合に適用される機能とは異なります。
- どのタイプの変数にも NULL 値を割り当てることができます。
- NULL 変数値は、Studio でモニタおよび更新できます。ワークフロー インスタンスをモニタする場合、NULL 変数値は null と表示されます。[変数を更新] ダイアログ ボックスで、変数値を null に設定できます。

次の節では、wli.bpm.server.evaluator.supportsNull=false の場合に適用される規則について説明します。このオプションを true に設定する前に、違いを理解しておくことが重要です。

旧バージョンで定義されたワークフローまたは NULL サポートが無効なワークフローは、影響を受ける機能に依存している場合、以前と同じようには機能しません。従来のワークフローを再検証し、適切なワークフロー変数値を初期化する必要があります。そうしない場合、NULL 変数サポートを有効にした場合にワークフローが機能しない可能性があります。

wli.bpm.server.evaluator.supportsNull=false

wli.bpm.server.evaluator.supportsNull=false の場合、WebLogic Integration プロセス エンジン は、旧リリースとまったく同じように動作します。以下の規則が適用されます。

- 値 NULL (大文字でも小文字でも可) は、どの式でも有効ではありません。たとえば、\$a * NULL は無効です。
- All comparison operations (=, <>, >=, < ...) on any null value yield false.

たとえば、ワークフロー変数 \$a および \$c に NULL 値があり、\$b に非 NULL 値がある場合は、次のようになります。

```
$a=$c      false
$a<>$c     false
$b<>$c     false
$b>$a     false
```

- 算術演算式に NULL 値を使用することはできません。たとえば、\$b * NULL では、例外が発生します。
- 論理演算式に NULL 値を使用することはできません。たとえば、\$a XOR NULL では、例外が発生します。
- 文字列連結 (StrCat) では、NULL 値を使用できます。たとえば、"wli "+ EventAttribute("not-existing-attribute") では、"wli null" という結果になります。
- 変数は、以下の初期値に設定されます。

```
String      ""
Boolean     false
Integer     0
Double      0.0
Date        現在の日付
Entity      Null
Session     Null
Object      Null
XML         Null
```

- 文字列変数にのみ、NULL 値を割り当てることができます。NULL 値をほかのタイプの変数に割り当てると、例外が発生します。
- NULL 変数値をモニタまたは更新することはできません。

BPM セキュリティ モデルについて

WebLogic Integration の Business Process Management (BPM) 機能用のセキュリティ モデルには、以下の特性があります。

- ユーザおよびグループは、BEA WebLogic Server セキュリティ レルムで保持されます。
- オーガニゼーションは、セキュリティ レルムの外にある BPM 固有のエンティティです。
- ロールは、WebLogic Server グループにマップされます。

注意： WebLogic Server Security Service に関する重要な背景知識を得るには、次のドキュメントを参照してください。

- 『*BEA WebLogic Platform Security の紹介*』 — 他の BEA WebLogic Platform コンポーネントを使用して BPM を使用する計画を立てる場合、このドキュメントの情報は重要です。
- 『*WebLogic Integration ソリューションのデプロイメント*』 の「WebLogic Integration セキュリティの使い方」 — WebLogic Integration セキュリティの概要を説明し、WebLogic Integration ドメインのセキュリティのコンフィグレーションに関する実用的なアドバイスを掲載しています。

コンフィグレーション ウィザードを使用して、BPM 機能をサポートするドメインを作成すると、ドメインは FileRealm セキュリティ レルムを使用するようにコンフィグレーションされます。このタイプのレルムでは、デフォルト ユーザ、グループ、およびアクセス制御リスト (ACL) は、ドメイン ディレクトリにある fileRealm.properties ファイルに保持されます。

デフォルトの RDBMSRealm のコンフィグレーションも行えます。ドメインの RDBMS レルムのコンフィグレーションを行う手順は次のとおりです。

- config.xml ファイルに、手動で要素を追加します。手順については、『*BEA WebLogic Integration 移行ガイド*』の「WebLogic Integration 2.1 から WebLogic Integration 7.0 への移行」の「RDBMS レルム からの移行」で説明します。

- `rdbmsrealm.jar` ファイルのパスを `classpath` に追加します。デフォルトでは、この JAR ファイルは `WLI_HOME\lib` ディレクトリに配置されています。

このレルムのコンフィグレーションを行うと、ユーザとグループは、**WebLogic Integration** リポジトリ用にコンフィグレーションされたデータベースで保持されます。

BPM 定義の ACL オブジェクトは、ユーザとグループでは違った方法で管理されます。**RDBMSRealm** を有効にしても、ACL オブジェクトは、**FileRealm** (`fileRealm.properties` ファイル) に格納され、**WebLogic Server Administration Console** で管理する必要があります。

既存の **FileRealm** を使用したり、**RDBMSRealm** を有効にしたり、代わりにセキュリティレルムをコンフィグレーションすることができます。サポートされている **WebLogic Server** セキュリティレルムのタイプの概要については、次の URL から「セキュリティの基礎概念」を参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/security/concepts.html>

コンフィグレーションするセキュリティレルムのタイプに関わらず、セキュリティレルムが 3-51 ページの「セキュリティレルムのガイドライン」のガイドラインに従っている限り、**WebLogic Integration Studio** でユーザ、オーガニゼーション、およびロールの作成を継続できます。

注意： **WebLogic Integration** および **WebLogic Server** のユーザ名とパスワード (**WebLogic Integration Studio** を使用して作成されたユーザ) は、JDK がサポートする文字セットに含まれる任意の文字 (国際文字を含む) で構成できます。

RDBMSRealm を有効にする手順については、3-52 ページの「**RDBMS** セキュリティレルムの有効化」で説明します。

カスタム セキュリティレルムを作成する手順については、3-54 ページの「カスタム セキュリティレルムのコンフィグレーション」で説明します。別のタイプ (NT または UNIX など) のセキュリティレルムを作成する場合は、**WebLogic Server** マニュアルで手順を参照してください。

BPM ユーザグループおよびパーミッション

BPM 機能をサポートするために、1-9 ページの「WebLogic Integration ユーザおよびパスワード」で説明されているデフォルト ユーザは、次に示す 3 種類のユーザグループにまとめられます。

- パーミッショングループメンバーは、それぞれが実行可能な操作の種類を基に定義されます。
- システムグループプロセスエンジンが必要とします。
- ロールグループメンバーは、BPM 定義ロールにより定義されます。
Studio でロールを作成すると、対応するロール固有のグループがセキュリティレلمで作成されます。

次の表に、デフォルトグループとそれぞれのグループに割り当てられたデフォルトメンバーを示します。

表 3-7 パーミッショングループとメンバー

パーミッショングループ	メンバー
AdministerUser	admin、joe、mary、guest、wlisystem
ConfigureComponents	admin、joe、mary、guest、wlisystem
ConfigureSystem	admin、joe、mary、guest、wlisystem
CreateTemplate	admin、joe、mary、guest、wlisystem
DeleteTemplate	admin、joe、mary、guest、wlisystem
ExecuteTemplate	admin、joe、mary、guest、wlisystem
MonitorInstance	admin、joe、mary、guest、wlisystem
UpdateTemplate	admin、joe、mary、guest、wlisystem

表 3-8 システム グループとメンバー

システム グループ	メンバー
everyone	admin、joe、system、mary、guest、wlisystem
wlpiAdministrators	admin、joe、system、mary、guest、wlisystem
wlpiUsers	admin、joe、system、mary、guest、wlisystem

表 3-9 ロール グループとメンバー

ロール グループ	メンバー
AccountingCDE	admin、joe
CustomerServiceCDE	admin
Role1Org1	admin、joe、mary
Role2Org1	admin、joe、mary
Role1Org2	admin、joe、mary
Role2Org2	admin、joe、mary
ShippingCDE	admin、mary

セキュリティ レルムのガイドライン

以下のガイドラインに従っている限り、代替セキュリティレルムを作成できません。

- BPM ユーザは WebLogic Server ユーザでなければなりません。
- system、admin、および wlisystem のユーザを作成する必要があります。
- 以下の WebLogic Server グループを作成する必要があります。
 - すべてのパーミッション グループ（表 3-7 を参照）

- wlpUsers グループ
- wlsystem ユーザは、必要なすべてのグループのメンバーでなければなりません。
- WebLogic Integration プロセス エンジンまたは XML リポジトリへのアクセスが必要なユーザは、wlpUsers グループのメンバーでなければなりません。

代替セキュリティレームの作成手順については、WebLogic Server 7.0 マニュアルを参照してください。

以上のガイドラインに従っている管理可能なセキュリティレームが格納され、次の節の説明に従ってコンフィグレーションされたら、WebLogic Integration Studio で追加ユーザ、ロール、およびオーガニゼーションを作成し、パーミッションをユーザとロールに割り当てることができます。情報と手順は、『WebLogic Integration Studio ユーザーズガイド』の「データの管理」を参照してください。

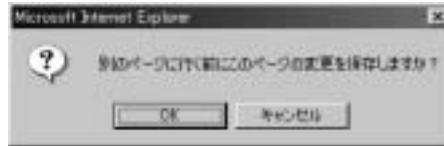
RDBMS セキュリティ レームの有効化

デフォルト FileRealm のセキュリティレームを RDBMSRealm に変更する手順は次のとおりです。

1. RDBMS レームで使用されるデータベースに WebLogic Integration リポジトリが格納されていることを確認してください (Integration データベース ウィザードを使用して、リポジトリをデータベースに格納することができます。このウィザードを起動するには、[スタート | プログラム | BEA WebLogic Platform 7.0 | WebLogic Integration 7.0 | Integration Examples | Integration Database Wizard] を選択してください)。
2. config.xml ファイルに要素を手動で追加して、RDBMSRealm のコンフィグレーションを行います。手順については、『BEA WebLogic Integration 移行ガイド』の「WebLogic Integration 2.1 から WebLogic Integration 7.0 への移行」の「RDBMS レームからの移行」で説明します。
3. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
4. 2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って、WebLogic Server Administration Console を起動します。

5. ナビゲーション ツリーで、[以前のセキュリティ] を選択します。
6. [ファイルレルム] タブを選択します。
注意： 次のメッセージが表示されたら、[キャンセル] をクリックしてこのウィンドウを閉じます。

図 3-8 ページ終了の確認



7. [キャッシングレルム] ドロップダウン リストから、[wlpiCachingRealm] を選択します。
8. [適用] をクリックします。
9. 1-21 ページの「WebLogic Integration の終了」の説明に従って、WebLogic Integration をシャットダウンします。
10. 更新するドメインの `DOMAIN_HOME` ディレクトリに移動します。
11. `fileRealm.properties` ファイルをコピーし、名前を変更してバックアップします。
12. テキスト エディタで元の `fileRealm.properties` ファイルを開きます。
13. `user` および `group` で始まるすべてのエントリを削除し、ファイルを保存します。
注意： `acl` で始まるエントリは削除しないでください。ACL オブジェクトは、`RDBMSRealm` が有効かどうかに関係なく、常に `fileRealm.properties` ファイルに保持されます。
14. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を再起動します。

カスタム セキュリティ レalmのコンフィグレーション

カスタム セキュリティレalmのコンフィグレーションには、次のタスクが必要です。

1. カスタムレalmの作成
2. キャッシングレalm (wlpiCachingRealm) のコンフィグレーション
3. キャッシングレalmの有効化
4. RDBMSRealm の削除 (省略可)

以下の各節では、各タスクの手順について説明します。

カスタムレalmの作成

カスタム セキュリティレalmを作成する手順は次のとおりです。

1. 1-18 ページの「WebLogic Integration の起動」の説明に従って、WebLogic Integration を起動します。
2. 2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って、WebLogic Server Administration Console を起動します。
3. ナビゲーションツリーで、[以前のセキュリティ | レalm] を選択し、[レalm] をクリックします。
4. 右ペインで、[新しい Custom Realm のコンフィグレーション] リンクを選択します。
新しいレalmの [コンフィグレーション] タブが表示されます。
5. カスタムレalm名を入力します。たとえば、「CUSTOMRealmForNetscapeDirSvr」などです。

注意： ほかのタブおよび属性は無視できます。これらの設定は使用されません。

- 『WebLogic Server Administration Console Online Help』の「セキュリティ互換性モード」の「カスタム セキュリティ レルムのインストール」の説明に従って、カスタム レルムを作成します。
- [作成] をクリックします。

キャッシング レルムのコンフィグレーション

キャッシング レルムをコンフィグレーションする手順は次のとおりです。

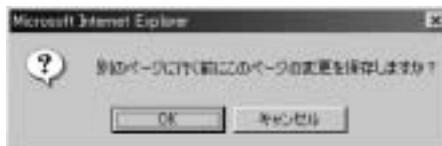
- ナビゲーション ツリーで、[以前のセキュリティ | キャッシング レルム | wlpICachingRealm] を選択します。
- [基本レルム] ドロップダウン リストから、作成したレルムを選択します。
- 右ペインで、[新しい Caching Realm のコンフィグレーション] を選択します。
- [キャッシュで大文字 / 小文字を区別] オプションをオフにします。
- [作成] をクリックします。

WLPI キャッシング レルムの有効化

WLPI キャッシング レルムを有効にする手順は次のとおりです。

- ナビゲーション ツリーで、[セキュリティ] を選択します。
 - [ファイルレルム] タブを選択します。
- 注意：** 次のメッセージが表示されたら、[キャンセル] をクリックしてこのウィンドウを閉じます。

図 3-9 ページ終了の確認



3. [キャッシング レルム] ドロップダウン リストから、[wlpiCachingRealm] を選択します。
4. [適用] をクリックします。

キャッシング レルムの削除

注意： この手順は省略可能です。

RDBMS レルムを削除する手順は次のとおりです。

1. ナビゲーション ツリーで、[セキュリティ | レルム] を選択します。
2. wlpiRDBMSRealm の右にあるゴミ箱アイコンをクリックします。

注意： キャッシング レルムで参照される RDBMS レルムを削除する場合、config.xml ファイルを編集してキャッシング レルムの BasicRealm 属性の値を変更して新しいカスタム レルムを指定するようにしなければなりません。

パスワードの更新

システム セキュリティを確実なものにするために、以下の手順に従って、1-9 ページの「WebLogic Integration ユーザおよびパスワード」で示されているデフォルトパスワードを更新します。

警告： wlsystem ユーザ名およびパスワードは、WebLogic Integration リポジトリに格納され、WebLogic Integration 実行時環境で使用されます。この種類のパスワードは、B2B Console を使用して更新しなければなりません。パスワードの更新に WebLogic Server Administration Console を使用しないでください。B2B Console で wlsystem パスワードを変更すると、WebLogic Integration リポジトリとセキュリティ レルムの両方で、パスワードが更新されます。

注意： WebLogic Integration および WebLogic Server のユーザ名とパスワード (WebLogic Integration Studio を使用して作成されたユーザ) は、JDK がサポートする文字セットにある任意の文字 (国際文字を含む) で構成できます。

システム パスワードの更新

アクティブ ドメインに対する `system` ログイン用パスワードは、次の手順に従って、**WebLogic Server Administration Console** で変更できます。

パスワードを変更する手順は次のとおりです。

1. 2-8 ページの「**WebLogic Server Administration Console** の起動」の説明に従って、**WebLogic Server Administration Console** を起動します。
2. ナビゲーション ツリーから [ユーザ] を選択して [ユーザ] ページを開きます。
3. [ユーザ パスワードの変更] セクションの名前フィールドで、変更対象のパスワードを所有しているユーザの名前を入力します。
たとえば、システム パスワードを変更するには、「`system`」と入力します。
4. [古いパスワード] フィールドに既存のパスワードを入力します。
5. [新しいパスワード] フィールドに新しいパスワードを入力します。
6. [パスワードの確認] フィールドに新しいパスワードを再入力します。
7. [変更] をクリックしてパスワードを更新します。
8. `startWebLogic.cmd` (Windows) または `startWebLogic` (UNIX) でドメイン用に指定したパスワードを変更します。

警告： `startWeblogic.cmd` コマンド (Windows) または `startWeblogic` コマンド (UNIX) が自動ログイン用にコンフィグレーションされます。
`system` 用のパスワードを変更する場合、このファイルで指定したパスワードを変更する必要があります。付録 A 「**WebLogic Integration** サンプル コンフィグレーション ファイル」で説明されているサンプル `startWebLogic` ファイルを参照してください。

BPM パスワードの更新

前の節で説明したとおり、**Studio** アプリケーションで作成した BPM ユーザ `admin`、`joe`、`mary`、および新しいユーザのパスワードを **WebLogic Server Administration Console** で更新できます。

wlssystem パスワードの更新

wlssystem ユーザのパスワードは、次の手順に従って、**B2B Console** でのみ変更できます。**WebLogic Server Administration Console** でこのパスワードを更新しないでください。

wlssystem ユーザのパスワードを更新する手順は次のとおりです。

1. 2-25 ページの「**B2B Console** の起動」の説明に従って、**B2B Console** を起動します。
2. ナビゲーションツリーから **[B2B]** を選択します。
3. まだ選択されていない場合は、上位レベルの **[コンフィグレーション]** タブを選択してから、その下の **[セキュリティ]** タブを選択します。

次の図に示す **[セキュリティ]** タブが表示されます。**[システム パスワード]** フィールドの内容は、wlssystem ユーザのパスワードです。

図 3-10 B2B の **[コンフィグレーション | セキュリティ]** タブ



4. `wlssystem` パスワードを更新するには、[システム パスワード] フィールド内を選択し、間違えないように新しいパスワードを入力します。
5. [適用] をクリックして `wlssystem` パスワードを更新します。
この手順により、**WebLogic Integration** リポジトリおよびセキュリティ レルムの両方で、`wlssystem` パスワードが更新されます。

メール セッション プロパティのカスタマイズ

コンフィグレーション ウィザードを使用してドメインを作成したときに指定したメールセッションプロパティは、ドメインのメールセッションの基本コンフィグレーションの設定に使用されます。既存の設定を更新するか、プロパティを追加することでメールセッションプロパティを変更できます。次の手順では、ドメインに対するメールセッションプロパティの更新方法について説明します。

ドメインに対するメールセッションプロパティを更新する手順は次のとおりです。

1. 1-18 ページの「**WebLogic Integration** の起動」の説明に従って、ドメインの **WebLogic Server Administration Console** を起動します。
2. [サービス | メール | `wlpiMailSession`] を選択します。次の図に示す `wlpiMailSession` の [コンフィグレーション] タブが表示されます。

図 3-11 wpiMailSession の [コンフィグレーション] タブ



3. 必要に応じて [プロパティ] フィールドの内容を編集します。
次の表では、設定可能なプロパティについて説明します。

表 3-10 メールセッションプロパティ

プロパティ	説明
mail.from	ワークフローによって送信される電子メールの送信元となる電子メール アドレス。
mail.host	メールまたは SMTP サーバのドメイン名。
mail.sender	受信時に電子メールに表示される送信者の名前。デフォルトは WebLogic Integration。
mail.user	ログオンに使用するユーザ名 (メールサーバが要求する場合)。
mail.password	ログオン認証に使用するパスワード (メールサーバが要求する場合)。

4. 変更が完了したら、[適用] をクリックします。

ドメインのルート ディレクトリ

WebLogic Integration の一部のコンポーネントは、ルート ディレクトリの特定のファイル (XML DTD など) にアクセスする必要があります。デフォルトでは、WebLogic Integration の `startWeblogic` スクリプトは、ドメインのホーム ディレクトリ (`DOMAIN_HOME`) をドメインのルート ディレクトリとして使用します。

たとえば、コンフィグレーション ウィザードを使用して `c:/bea/user_projects/mydomain` にドメインを作成した場合、そのディレクトリが `DOMAIN_HOME` になります。`DOMAIN_HOME` ディレクトリの `startWeblogic.cmd` スクリプトは、`weblogic.RootDirectory` プロパティを使用して、ルート ディレクトリ (実行時ディレクトリとも呼ぶ) を同じディレクトリ (`c:/bea/user_projects/mydomain`) に設定します。

代替文字セットの使用

別のロケールで WebLogic Integration を実行する場合でも、データベース、オペレーティング システム、WebLogic Integration、および WebLogic Integration クライアント アプリケーション (Studio、Worklist、および Format Builder) がすべて同じロケールで実行されている限り、特別なコンフィグレーションは必要ありません。

インストール後に必要な唯一の変更は、B2B の設定です。B2B Console で代替文字セットによる入力を有効にするには、言語および文字セット設定が正しく定義されていることを確認する必要があります。言語と言語のデフォルト文字セットの設定の詳細については、『*B2B Integration Administration Console* オンラインヘルプ』の「B2B Integration のコンフィグレーション」の「[Preference] の設定」を参照してください。

WebLogic Integration ポート番号の変更

1 台のマシン上で実行されている 2 つの WebLogic Integration インスタンスがデフォルトのポート番号 (7001) を共有することはできません。1 台のマシン上で 2 つのインスタンスを実行する場合、一方のインスタンスが使用するポート番号を変更する必要があります。

警告: また、WebLogic Server のリスン ポート番号以外のパラメータを変更しなければならないこともあります。詳細については、表 3-11 を参照してください。

ご使用のアプリケーションで WebLogic Integration ポート番号を変更する手順は次のとおりです。

- 手順 1. – WebLogic Server のリスン ポート番号の変更
- 手順 2. – 他の WebLogic Integration ポート番号の変更

手順 1. – WebLogic Server のリスン ポート番号の変更

WebLogic Server リスン ポート番号は、サーバが接続のリスンを行う専用 TCP/IP ポートを指定します。サーバでコンフィグレーションされているリスンポートを変更するには、3-62 ページの「WebLogic Server Administration Console でリスン ポートを変更する」または 3-64 ページの「コンフィグレーション ファイルを編集してリスン ポートを変更する」の手順を実行してください。

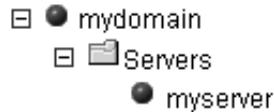
WebLogic Server Administration Console でリスン ポートを変更する

WebLogic Server Administration Console でリスン ポートを変更する手順は次のとおりです。

1. 2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って、WebLogic Server Administration Console を起動します。

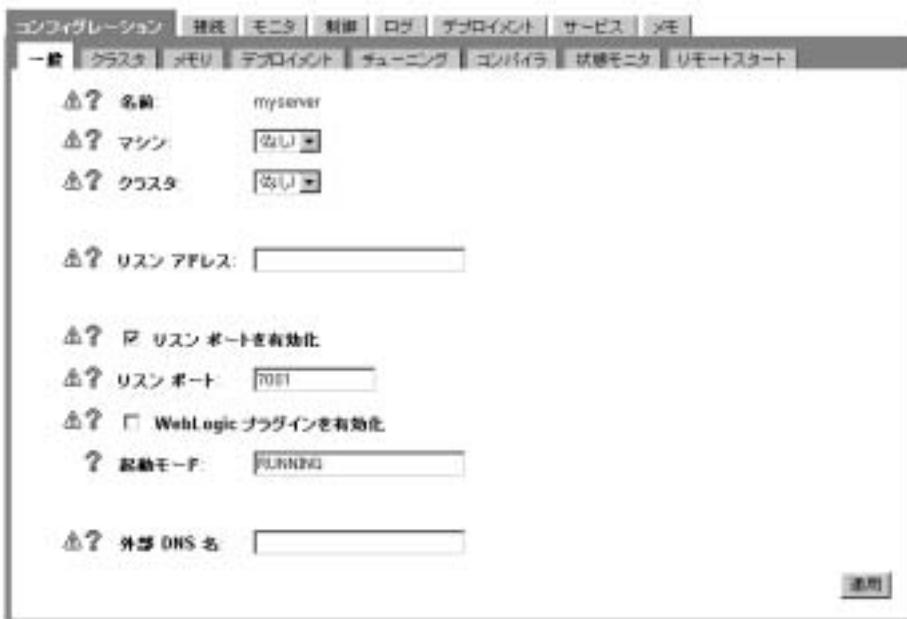
2. WebLogic Server Administration Console の左ペインのナビゲーション ツリーで、[サーバ | myserver] を選択します。ここで、myserver は、次の図に示すように、変更するリスンポートのある WebLogic Server の名前です。

図 3-12 サーバの選択



次の図に示す WebLogic Server の [コンフィグレーション] ページが表示されます。

図 3-13 WebLogic Server Administration Console の [コンフィグレーション] ページ



3. [リスンポート] フィールドに新しいポート番号を入力して、[適用] をクリックします。

4. サーバを再起動して、新しい設定を有効にします。

警告: 必ず、WebLogic Server Administration Console を使用して、リスンポートを変更してください。B2B Console でポート番号を変更しないでください。

コンフィグレーション ファイルを編集してリスンポートを変更する

注意: サーバの実行中に、config.xml ファイルを編集しないでください。サーバが実行されているときにリスンポートを更新するには、上記の手順を実行してください。

config.xml ファイルを編集してリスンポートを変更する手順は次のとおりです。

1. 管理サーバの `DOMAIN_HOME/config.xml` ファイルを任意のテキストエディタで開きます。
ここで、`DOMAIN_HOME` は、ドメインのルートへの絶対パス（たとえば、`c:\bea\user_projects\domain` ディレクトリ）です。
2. config.xml ファイルで、対象サーバの ListenPort 設定を見つけ、リスンポート番号を変更します（コードリスト 3-5 の太字参照）。
3. ListenPort を、同じマシンで実行されている他の WebLogic Server インスタンスで使用していないユニークな番号に変更します。

コード リスト 3-5 config.xml ファイルで、リスンポート要素を変更する

```
<Server
  ListenPort="7001"
  Name="myserver"
  TransactionLogFilePrefix="c:\bea\weblogic700\samples/integration/config/samples/logs/"
  StdoutDebugEnabled="true"
  StdoutEnabled="true"
  StdoutSeverityLevel="64" NativeIOEnabled="true">
```

手順 2. ー他の WebLogic Integration ポート番号の変更

これ以外の WebLogic Integration ポート番号を更新するための手順は、アプリケーションで WebLogic Integration のどの機能が使用されているかによって決まります。ご使用のアプリケーションのポート番号を変更するための手順を確認するには、次の表を参照してください。

表 3-11 WebLogic Integration ポート番号の変更

ご使用のアプリケーションが実装する WebLogic Integration の機能	実行する手順を説明している節
B2B セキュリティ	<i>B2B Integration</i> セキュリティの実装の「セキュリティのコンフィグレーション」の「SSL プロトコルと相互認証のコンフィグレーション」
Application Integration	3-67 ページの「Application View Console のポート番号の変更」
B2B エンジン	3-66 ページの「B2B URI エンドポイントの変更」
B2B Administration Console	3-67 ページの「B2B Console のポート番号の変更」
Application Intergration、B2B、または BPM	3-68 ページの「WebLogic Integration シャットダウン スクリプトの変更」

警告： サンプルに含まれる多くのファイルでは、ポート番号が設定されていません。サンプルでのデフォルト ポート番号の変更は、エラーにつながるので、お勧めできません。

B2B URI エンドポイントの変更

WebLogic Integration URI エンドポイントは、トレーディング パートナが B2B メッセージをリスンする位置を指定します。3-62 ページの「手順 1. – WebLogic Server のリスン ポート番号の変更」の説明に従って、WebLogic Server のリスンポートを変更しており、しかも WebLogic Integration B2B 機能を使用する場合は、B2B トレーディング パートナの URI エンドポイントを変更する必要があります。

URI エンドポイントのポート番号を変更する手順は次のとおりです。

1. 2-25 ページの「B2B Console の起動」の説明に従って、B2B Console を起動します。
2. B2B Console の左ペインのナビゲーション ツリーで、[B2B | トレーディング パートナ] を選択します。
3. 右ペインに表示されるリストから、ご使用の WebLogic Integration アプリケーションに関するトレーディング パートナを選択します。
4. 右ペインで、[URI] を選択して、ポート番号を更新し、[追加/適用] をクリックします。チェーンのすべての URI について、この手順を繰り返します。
5. コンフィグレーションを行うすべてのトレーディング パートナについて、手順 2～4 を繰り返します。

警告： URI エンドポイントは、Bulk Loader リポジトリ データファイルでも指定されています。リポジトリで URI エンドポイントのポート番号を変更し、しかも Bulk Loader リポジトリにその変更を反映する場合は、必ず、B2B Console またはコマンドライン Bulk Loader を使用して現在のリポジトリ情報をエクスポートしてください。詳細については、『*B2B Integration 管理ガイド*』の「B2B Integration コンポーネントのインポートとエクスポート」を参照してください。

Application View Console のポート番号の変更

3-66 ページの「B2B URI エンドポイントの変更」の説明に従って、WebLogic Server のリスンポートを変更し、しかも WebLogic Integration Application View Console を使用する場合は、startAIconsole コマンドで指定されるポート番号を更新する必要があります。

startAIconsole コマンドで指定されるポート番号を更新する手順は次のとおりです。

1. 任意のテキスト エディタで、次のいずれかのコマンド ファイルを開きます。
`WLI_HOME/bin/startAiconsole.cmd` (Windows) または
`WLI_HOME/bin/startAiconsole.sh` (UNIX)
2. ポート番号を、サーバのリスンポートを更新したとき割り当てた値に設定します。
3. ファイルを保存します。

B2B Console のポート番号の変更

3-66 ページの「B2B URI エンドポイントの変更」の説明に従って、WebLogic Server のリスンポートを変更し、しかも WebLogic Integration B2B Console を使用する場合は、startB2Bconsole コマンドで指定されるポート番号を更新する必要があります。

startB2Bconsole コマンドで指定されるポート番号を更新する手順は次のとおりです。

1. 任意のテキスト エディタで、次のいずれかのコマンド ファイルを開きます。
`WLI_HOME/startB2bconsole.cmd` (Windows) または
`WLI_HOME/startB2bconsole.sh` (UNIX)
2. ポート番号を、サーバのリスンポートを更新したとき割り当てた値に設定します。
3. ファイルを保存します。

WebLogic Integration シャットダウン スクリプトの変更

3-66 ページの「B2B URI エンドポイントの変更」の説明に従って、WebLogic Server のリスン ポートを変更した場合は、サーバのシャットダウン スクリプトのポート番号を変更する必要があります。

シャットダウン スクリプトのポート番号を変更する手順は次のとおりです。

1. 任意のテキスト エディタで、次のいずれかのサーバ シャットダウン コマンドのファイルを開きます。`DOMAIN_HOME\stopWeblogic.cmd (Windows)` または `DOMAIN_HOME/stopWeblogic.cmd (UNIX)`

たとえば、`c:\bea\user_projects` ディレクトリの `mydomain` ドメインのサーバ ポート番号を定義する場合、

`c:\bea\user_projects\mydomain\stopWebLogic.cmd` ファイルを開きます。

2. ポート番号を、サーバのリスン ポートを更新したとき設定した値に設定します。
3. ファイルを保存します。

非永続モードの B2B をデプロイする

デフォルトでは、WebLogic Integration は、永続モードでデプロイされます。このモードでは、ハードウェア障害やネットワーク障害が発生しても WebLogic Integration を復旧できます。WebLogic Integration を永続モードでデプロイすることの利点については、『WebLogic Integration ソリューションのデプロイメント』の「WebLogic Integration の高可用性」の「WebLogic Integration の回復から期待できること」を参照してください。

永続モードをオフにする機能は、WebLogic Integration で提供される B2B 機能についてのみサポートされます。永続モードは、WebLogic Server Administration console を使用してオフにできます。詳細については、『B2B Integration 管理ガイド』の「永続と回復のコンフィグレーション」を参照してください。

A WebLogic Integration サンプル コンフィギュレーション ファイル

この付録では、開発者とシステム管理者向けに、BEA WebLogic Integration のサンプル コンフィギュレーション ファイルを説明します。このファイルは、インストール時に自動的にカスタマイズされるので、サポートされているコンピュータシステムおよびネットワークで動作します。必要な情報を説明しながら、各サンプル ファイルを示していきます。

この付録のトピックは以下のとおりです。

- config.xml
- setEnv
- startWeblogic
- fileRealm.properties

各ファイルには、インストールプログラムによって更新されるパラメータのサンプル値が含まれています。このサンプル値は、**太字**で強調表示します。

注意： config.xml などのファイルで使用される頭辞語は、旧リリースから引き継がれているものです。WLC は B2B、WLPI は Business Process Management (BPM)、WLAI は Application Integration、WLXT は Data Integration を表しています。

config.xml

サンプルドメインは、インストール時に次に示す一連のタスクのいずれかを実行した場合に作成されます。このドメインで、WebLogic Integration サンプルアプリケーションを実行できます。

- 標準的なインストール オプションを選択した。
- カスタム インストール オプションを選択し、インストールする項目の 1 つに WebLogic Integration サンプルを選んだ。

ドメインのコンフィグレーションは、XML (eXtensible Markup Language) で定義されます。サンプルドメイン コンフィグレーションの永続ストレージは、`SAMPLES_HOME/samples/integration/config` に配置される `config.xml` によって提供されます。ここで、`SAMPLES_HOME` は、WebLogic Platform のサンプルディレクトリです。この項では、WebLogic Integration サンプルドメインの `config.xml` ファイルの内容について説明します。

`config.xml` で捕捉された設定を WebLogic Server Administration Console で表示および変更できます。2-8 ページの「WebLogic Server Administration Console の起動」の説明に従って WebLogic Server Administration Console を起動すると、サーバのホーム ページが表示されます。左ペインに表示されたナビゲーションツリーの項目は、`config.xml` ファイルで定義された要素と属性に対応していません。

`config.xml` ファイルのルート要素は Domain です。次の図で示すように、ファイルのこのセクションでは、Application Integration DBMS サンプルアダプタ、BEA WebLogic Adapter for Peregrine Power.Enterprise!、およびデフォルトの Web アプリケーションがデプロイされます (デフォルトの Web アプリケーションは、デプロイされる他の Web アプリケーションでは解決できない HTTP 要求に応答します)。

図 A-1 config.xml ファイル：パート 1

```

<?xml version="1.0" encoding="UTF-8"?>
<!--If your domain is active, please do not edit the config.xml file. Any changes
made to that file while the domain is active will not have any effect on the
domain's configuration and are likely to be lost. If your domain is inactive,
you may edit this file with an XML editor. If you do so, please refer to the BEA
WebLogic Server Configuration Reference documentation available from
http://edocs.bea.com/wls/dcs70/config_xml. In general, we recommend that changes
to your configuration file be made through the Administration Console.-->
<Domain Name="samples">
  <Application Deployed="true" Name="BEA_POWERENTERPRISE_3_0"
    Path="C:\bee\weblogic700\integration\adapters\powerenterprise\lib\BEA_POWERENTERPRISE_3_0_EAR.ear" TwoPhase="true">
    <ConnectorComponent Name="BEA_POWERENTERPRISE_3_0"
      Targets="myserver" URI="BEA_POWERENTERPRISE_3_0.ear"/>
    <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_EventRouter"
      Targets="myserver" URI="BEA_POWERENTERPRISE_3_0_EventRouter.war"/>
    <WebAppComponent Name="BEA_POWERENTERPRISE_3_0_Web"
      Targets="myserver" URI="BEA_POWERENTERPRISE_3_0_Web.war"/>
  </Application>
  <Application Deployed="true" Name="BEA_WLS_DBMS_ADK"
    Path="C:\bee\weblogic700\integration\adapters\dbms\lib\BEA_WLS_DBMS_ADK.ear" TwoPhase="true">
    <ConnectorComponent Name="BEA_WLS_DBMS_ADK" Targets="myserver" URI="BEA_WLS_DBMS_ADK.ear"/>
    <WebAppComponent Name="BEA_WLS_DBMS_ADK_Web" Targets="myserver" URI="BEA_WLS_DBMS_ADK_Web.war"/>
    <WebAppComponent Name="DbmsEventRouter" Targets="myserver" URI="BEA_WLS_DBMS_ADK_EventRouter.war"/>
  </Application>
  <Application Deployed="true" Name="DefaultWebApp_myserver"
    Path="C:\bee\weblogic700\samples\integration\config\samples\application\DefaultWebApp_myserver" TwoPhase="false">
    <WebAppComponent IndexDirectoryEnabled="true"
      Name="DefaultWebApp_myserver" Targets="myserver" URI="DefaultWebApp_myserver"/>
  </Application>

```

以下の 2 つの図で示すように、ファイルの次のセクションでは、WebLogic Integration アプリケーションに必要な EJB と Web アプリケーションをデプロイしています。

WebLogic Server では、WebLogic Integration によりエクスポートされた EAR ファイルおよび JAR ファイルの位置と、WLI application.xml ファイルおよび weblogic-application.xml ファイルの位置を確認するために、Path 属性が使用されます (application.xml ファイルおよび weblogic-application.xml ファイルは、path\META-INF ディレクトリにあります)。

WebLogic Integration J2EE コンポーネントは、エクスポートされた形式を使用してデプロイされます。エクスポートされた形式を使用した J2EE コンポーネントのデプロイの詳細については、次の URL から BEA WebLogic Server ドキュメント内の『*Web アプリケーションのアセンブルとコンフィグレーション*』を参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/webapp/basics.html>

コンポーネントがエクスポートされた形式でデプロイされる場合、DeploymentOrder 属性は、指定されません。代わりに、application.xml ファイルでの J2EE コンポーネントの順序によって、デプロイ順が決定されます。WLI アプリケーションのコンポーネントをデプロイしない場合は、application.xml ファイルのコンポーネントの要素を削除しないでください。代わりに、Targets 属性からサーバ名を削除してください (Targets="")。

図 A-2 config.xml ファイル：パート 2

Application
要素のパス
属性は実際の
インストールと
は異なる場合
があります。

WLI
Application
要素 -
WLI
アプリケーション
を構成する
全 jar と war
ファイルの定義
のパス
属性は実際の
インストールと
は異なる場合
があります。

```

<Application Deployed="true" Name="WLI"
  Path="C:\bea\weblogic700\integration\lib" TwoPhase="true">
  <EJBComponent Name="Sample BPM Plug-in" Targets="myserver" URI="U
iugin-ejb.jar"/>
  <EJBComponent Name="Sample EJB" Targets="myserver" URI="pobee
  <EJBComponent Name="WLI Repository" Targets="myserver" URI="t
  <EJBComponent Name="WLI-AI Async Processor" Targets="myserver
  <EJBComponent Name="WLI-AI BPM Plug-in" Targets="myserver" URI="U
  <EJBComponent Name="WLI-AI Event Processor" Targets="myserver
  <EJBComponent Name="WLI-AI Server" Targets="myserver" URI="wi
  <EJBComponent Name="WLI-B2B RN BPM Plug-in" Targets="myserver
  <EJBComponent Name="WLI-B2B RN MDB" Targets="myserver" URI="b
  <EJBComponent Name="WLI-D2D Startup" Targets="myserver" URI="
  <EJBComponent Name="WLI-B2B ebXML BPM Plug-in"
    Targets="myserver" URI="ebxml-bpm-plugin.jar"/>
  <EJBComponent Name="WLI-BPM Event Processor" Targets="myserve
  <EJBComponent Name="WLI-BPM Initialization" Targets="myserver
  <EJBComponent Name="WLI-BPM Plugin Manager" Targets="myserver
  <EJBComponent Name="WLI-BPM Server" Targets="myserver" URI="t
  <EJBComponent Name="WLI-DI BPM Plug-in" Targets="myserver" URI="U
  <EJBComponent Name="WLI-DI Server" Targets="myserver" URI="UI
  />

```

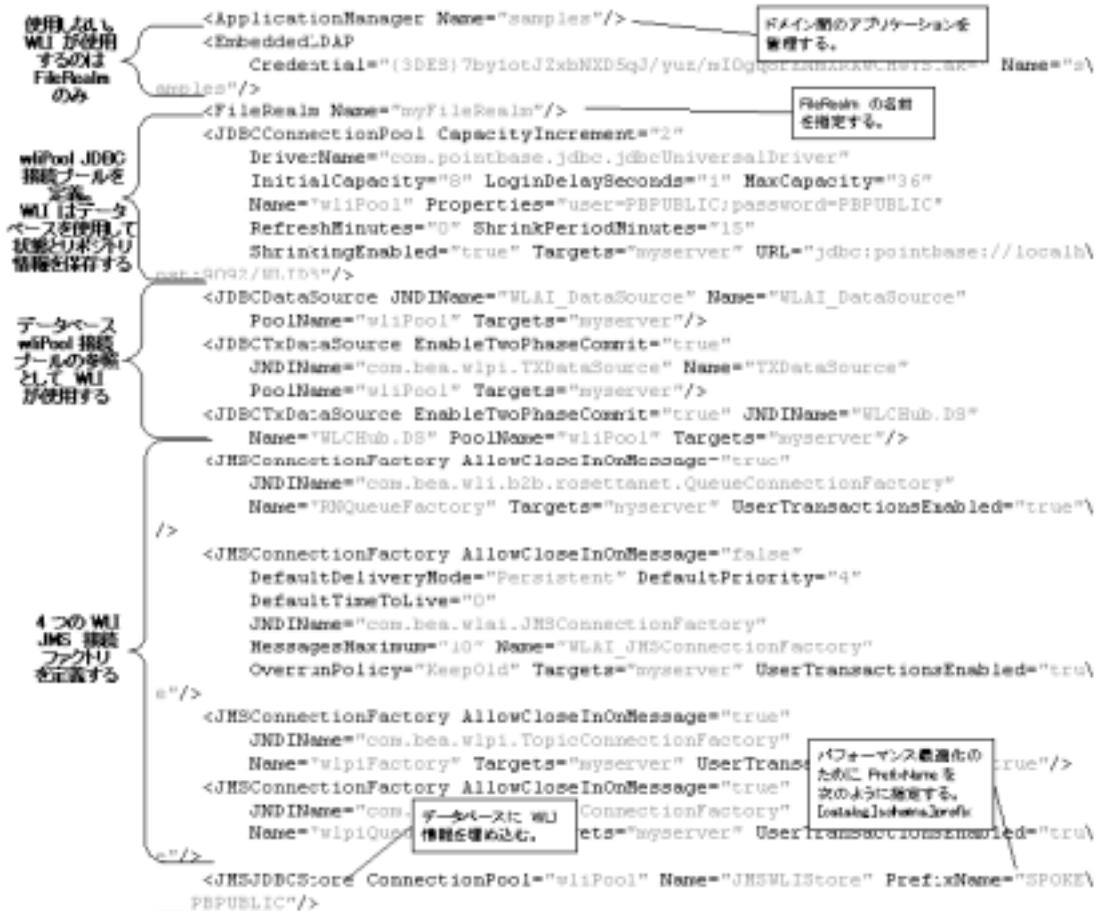
次の図に示すように、config.xml ファイルの次のセクションには、Application Integration 機能に関連する要素が記載されます。

図 A-3 config.xml ファイル : パート 3



次の図に示すように、config.xml ファイルの次のセクションでは、WebLogic Integration 用の JDBC 接続プール (wliPool)、JDBC データ ソース、および JDBC 接続ファクトリを定義しています。

図 A-4 config.xml ファイル：パート 4



次の図に示すように、WebLogic Server Administration Console のナビゲーションツリーで [サービス | JDBC | 接続プール | wliPool] を選択することにより、WebLogic Integration JDBC 接続プール (wliPool) の属性を表示したり更新したりすることもできます。

図 A-5 WebLogic Integration JDBC 接続プール



JDBC コンフィグレーション更新の詳細については、3-2 ページの「ドメインに対する新しいデータベースのコンフィグレーション」を参照してください。

また、次の図に示すように、WebLogic Server Administration Console のナビゲーション ツリーで [サービス | JMS | 接続ファクトリ] を選択して、4 つの WebLogic Integration 接続ファクトリを表示することもできます。

図 A-6 WebLogic Integration 接続ファクトリ



警告： WebLogic Integration のマニュアルで指示される場合を除いて、WebLogic Integration 接続ファクトリの属性を変更しないでください。予期しない結果が発生することがあります。たとえば、接続ファクトリの Default Delivery Mode を変更すると、システム障害またはネットワーク障害が発生した場合に WebLogic Integration を復旧できないことがあります。

パフォーマンスを最適なものにするために、JMS JDBC ストアのプレフィックス名をご使用のデータベースに合わせてカスタマイズしてください。プレフィックス名については、次の構文を使用してください。

```
[[catalog.]schema.]prefix]
```

JMS ストアプレフィックス名を更新するには、[サービス | JMS | ストア | JMSWLStore] を選択します。次の図に示す [JMSWLStore] ページが表示されます。

図 A-7 JMS ストア ページ



[プレフィックス名] フィールドに適切なプレフィックス名（使用するデータベース コンフィギュレーションに適合するスキーマプレフィックスとカタログプレフィックスを備えたプレフィックス名）を入力します。使用するデータベース コンフィギュレーションに適合するスキーマプレフィックスとカタログプレフィックスを確認するには、`DOMAIN_HOME\dbinfo\database_type\setDBVarsExt` を参照してください。`DOMAIN_HOME` は、ドメインのルートへの絶対パス名を示し、`database_type` は、使用しているデータベース（oracle、sybase、db2、mssql、またはpointbase）を意味します。

次の図に示すように、`config.xml` ファイルの次のセクションでは、WebLogic Integration の JMS Server が定義されます。

図 A-8 config.xml ファイル：パート 5



また、次の図に示すように、WebLogic Server Administration Console のナビゲーション ツリーで [サービス | JMS | 接続ファクトリ] を選択して、WebLogic Integration JMS Server (WLIJMServer) を表示することもできます。

図 A-9 WebLogic Integration JMS Server



次の図に示すように、config.xml ファイルの次のセクションには、レルム、セキュリティ、および他の要素が記載されます。

図 A-10 config.xml ファイル：パート 6



セキュリティレールの詳細については、3-48 ページの「BPM セキュリティ モデルについて」を参照してください。

次の図で示すように、config.xml ファイルの次のセクションには、Server 要素および属性、ならびに WebLogic Integration B2B シャットダウン クラスが記載されます。

図 A-11 config.xml ファイル：パート 7



ドメイン要素で指定されるドメイン名は、WebLogic Server Administration Console で指定されるドメイン名に対応します。たとえば、コンソールのナビゲーション ツリーから [myserver] を選択した場合、次の図に示すようにサーバ ページが表示されます。

図 A-12 WebLogic Server Administration Console のサーバ ページ



このページの各タブを選択すると、リスンポート、ログファイル名、ログレベルおよび出力などのサーバ コンフィグレーション パラメータを更新できます。設定のヘルプについては、右上隅の疑問符をクリックしてオンライン ヘルプを表示するか、次の URL にある WebLogic Server マニュアルを参照してください。

<http://edocs.beasys.co.jp/e-docs/wls/docs70/index.html>

setEnv

この実行ファイルは、**WebLogic Integration** の環境変数の設定に使用します。このファイルのインスタンスは、**WebLogic Integration** インストール ディレクトリ、およびその下の `bin` ディレクトリにあります。このファイルは、`startWeblogic` コマンド ファイル、および **WebLogic Integration** に付属の他のコマンド ファイルによって呼び出されます。

次の図に、サンプルドメインの `setenv` スクリプト（Windows 上で実行する場合は `setEnv.cmd`、UNIX 上で実行する場合は `setenv.sh`）の最初のセクションを示します。（このスクリプトは `WLI_HOME` ディレクトリにあります。ここで、`WLI_HOME` は、**WebLogic Integration** をインストールしたディレクトリです。）太字部分は、インストール時にインストール場所を基に設定された値です。

このコマンドとこのコマンドにより設定される環境変数の詳細については、付録 B 「**WebLogic Integration コマンド**」を参照してください。

コードリスト A-1 setEnv コマンド

```
@echo off
.
.
.
set JAVA_HOME=C:\bea\jdk131_03
set PATH=%PATH%;%JAVA_HOME%\bin
set BEA_HOME=C:\bea
set WL_HOME=C:\bea\weblogic700\server
set WLI_HOME=C:\bea\weblogic700\integration
set SAMPLES_HOME=C:\bea\weblogic700\samples
set JAVA_OPTIONS="-XX:MaxPermSize=131072k"
REM set DB2_HOME=

:checkJDK
if exist %JAVA_HOME%\lib\nul goto setCP
echo.
echo The JDK wasn't found in directory %JAVA_HOME%.
echo Please edit the setEnv.cmd script so that the JAVA_HOME
echo variable points to the location of your JDK.
goto error

:setCP
REM Set the WLI common Classpath
set WLICOMMONCP=

for %%a in (
```

```
wlicommon.jar,  
) do call :wlicmnlb %%a  
  
.  
.  
.
```

startWeblogic

この実行コマンド ファイルは **WebLogic Server** を起動します。このコマンドは、`config.xml` ファイルの内容を使用して、対応するドメインの `config.xml` ファイルで指定した **WebLogic Integration** アプリケーションとリソースをデプロイします。このファイルは、各ドメインのディレクトリにインストールされます。たとえば、**Windows** システムで、新しいドメイン (`mydomain`) をデフォルト位置に作成する場合、そのドメインの **WebLogic Integration** `startWeblogic.cmd` のディレクトリの位置は、次のとおりです。

```
BEA_HOME/user_projects/mydomain
```

このパス名の、`BEA_HOME` は、使用する **BEA Home** ディレクトリの位置を示しません。たとえば、`C:\bea` になります。

次のリストに、サンプルドメインの `startWeblogic.cmd` (**Windows**) を示します。コンフィグレーション済みのサンプルドメインの `startWebLogic.cmd` コマンドは、`SAMPLES_HOME/integration/config/samples` にあります。ここで、`SAMPLES_HOME` は、**WebLogic Platform** のサンプルディレクトリを示します。

コードリスト A-2 startWebLogic.cmd コマンド

```
@echo off  
.  
.  
.  
if /I "%WLI_HOME%" == "" call c:\bea\weblogic700\integration\setEnv.cmd  
  
setlocal  
  
call %SAMPLES_HOME%\integration\samples\bin\SetSampleData || goto :EOF  
if "%DB_TYPE%" == "pointbase" goto pointbase  
goto run
```

```

:pointbase
REM Invoke ant script to finish up work
set SCRIPT=%WLI_HOME%\lib\scripts\PointbaseChecker.xml

if exist %SCRIPT% goto ant
echo WARNING: No ant script to check for pointbase (%SCRIPT%)
goto finish

:ant
%JAVA_HOME%\bin\java -classpath %WLICP% -Xmx256m org.apache.tools.ant.Main \
-buildfile %SCRIPT% || goto error

:run
REM Remove old logs.
del /f /q %WLI_SAMPLES_HOME%\*.log* > nul 2>&1
del /f /q %WLI_SAMPLES_HOME%\logs\* > nul 2>&1

REM Change directory to where DTD files are located
cd /d %WLI_SAMPLES_HOME%

REM WLIS data directory
if not exist %WLI_SAMPLES_HOME%\data mkdir %WLI_SAMPLES_HOME%\data || goto finish

REM Start weblogic
%JAVA_HOME%\bin\java %JAVA_OPTIONS% %DB_JVMARGS% -Xmx256m -classpath %SVRCP% \
-Dbea.home=%BEA_HOME% -Dweblogic.home=%WL_HOME% \
-Dweblogic.system.home=%SAMPLES_HOME%\integration -Dweblogic.Domain=samples \
-Dweblogic.management.username=system -Dweblogic.management.password=security \
-Dweblogic.Name=myserver -Dweblogic.RootDirectory=%SAMPLES_HOME%\integration \
-Djava.security.policy=%WL_HOME%\lib\weblogic.policy \
-Dweblogic.management.discover=false \
-Dweblogic.SystemDataStoreConfigDirectory=%WLI_HOME%\lib \
-Dweblogic.servlet.ClasspathServlet.disableStrictCheck=true weblogic.Server

:finish
endlocal

```

WebLogic Server を起動する java コマンドの

`[-Dweblogic.management.password=security]` オプションは、パスワードを提供します。このオプションを削除した場合は、パスワードを入力するよう要求されます。このパスワードを変更する場合は、このオプションを削除するかまたは更新しなければなりません。

fileRealm.properties

このプロパティ ファイルは、WebLogic Server の起動時に作成される User、Group、および ACL オブジェクトを制御します。

次のリストでは、サンプルドメインの fileRealm.properties ファイルを示します。

コードリスト A-3 fileRealm.properties ファイル

```
#Fri Apr 12 22:21:45 BST 2002
acl.access.weblogic.admin.mbean.MBeanHome=wlcSamplesUser, everyone, guest
acl.boot.weblogic.server=system, everyone
acl.create.weblogic.jms.ServerSessionPool=everyone
acl.enablermonitor.WLCAdmin=admin
acl.execute.weblogic.servlet.AdminClients=system
acl.execute.weblogic.servlet.AdminConnections=system
acl.execute.weblogic.servlet.AdminEvents=system
acl.execute.weblogic.servlet.AdminJDBC=system
acl.execute.weblogic.servlet.AdminLicense=system
acl.execute.weblogic.servlet.AdminMain=system
acl.execute.weblogic.servlet.AdminProps=system
acl.execute.weblogic.servlet.AdminRealm=system
acl.execute.weblogic.servlet.AdminThreads=system
acl.execute.weblogic.servlet.AdminVersion=system
acl.execute.weblogic.servlet.Certificate=system
acl.execute.weblogic.servlet.ConsoleHelp=everyone
acl.execute.weblogic.servlet.T3AdminMain=system
acl.execute.weblogic.servlet.classes=everyone
acl.execute.weblogic.servlet.containerManaged=everyone
acl.execute.weblogic.servlet=system, everyone
acl.hubconfig.WLCAdmin=admin
acl.list.weblogic.jndi.weblogic.ejb=system, everyone
acl.list.weblogic.jndi.weblogic.fileSystem=everyone
acl.list.weblogic.jndi.weblogic.rmi=everyone
acl.list.weblogic.jndi.weblogic=system, everyone
acl.list.weblogic.jndi=everyone
acl.lockServer.weblogic.admin=system
acl.lookup.weblogic.admin.mbean.MBeanHome=wlcSamplesUser, everyone, guest
acl.lookup.weblogic.jndi.weblogic.ejb=system, everyone, guest
acl.lookup.weblogic.jndi.weblogic.fileSystem=everyone
acl.lookup.weblogic.jndi.weblogic.rmi=everyone
acl.lookup.weblogic.jndi.weblogic=everyone
```

```
acl.lookup.weblogic.jndi=everyone
acl.modify.weblogic.admin.acl=system
acl.modify.weblogic.jndi.weblogic.ejb=system,everyone,guest
acl.modify.weblogic.jndi.weblogic.fileSystem=everyone
acl.modify.weblogic.jndi.weblogic.rmi=everyone
acl.modify.weblogic.jndi.weblogic=system,everyone,guest
acl.modify.weblogic.jndi=everyone
acl.read.managedObject=system
acl.read.weblogic.workspace=system,everyone
acl.receive.weblogic.jms=everyone
acl.reserve.weblogic.jdbc.connectionPool.wliPool=wliSystem,everyone
acl.reserve.weblogic.jdbc.connectionPool=system
acl.reset.weblogic.jdbc.connectionPool.examplesPool=wlcSamplesUser,admin,guest
acl.reset.weblogic.jdbc.connectionPool.wliPool=wlcSamplesUser,wliSystem,admin,\
guest
acl.reset.weblogic.jdbc.connectionPool.wlpiPool=wliSystem,admin,guest
acl.reset.weblogic.jdbc.connectionPool=system
acl.send.weblogic.jms=everyone
acl.shrink.weblogic.jdbc.connectionPool.examplesPool=everyone
acl.shrink.weblogic.jdbc.connectionPool.wliPool=wlcSamplesUser,wliSystem,\
admin.guest
acl.shrink.weblogic.jdbc.connectionPool.wlpiPool=wliSystem,admin,guest
acl.shutdown.weblogic.admin=system
acl.unlockServer.weblogic.admin=system
acl.write.managedObject=system
acl.write.weblogic.workspace=system,everyone
```

```
group.AccountingCDE=admin,joe
group.AdministerUser=admin,joe,mary,guest,wliSystem
group.Administrators=ConfigureComponents,system
group.ConfigureComponents=admin,joe,wliSystem,mary,guest
group.ConfigureSystem=admin,joe,mary,guest,wliSystem
group.CreateTemplate=admin,joe,wliSystem,mary,guest
group.CustomerServiceCDE=admin
group.DeleteTemplate=admin,joe,wliSystem,mary,guest
group.Deployers=Administrators
group.ExecuteTemplate=admin,joe,wliSystem,mary,guest
group.MonitorInstance=admin,joe,wliSystem,mary,guest
group.Monitors=Administrators
group.Operators=Administrators
group.Role1Org1=admin,joe,mary
group.Role1Org2=admin,joe,mary
group.Role2Org1=admin,joe,mary
group.Role2Org2=admin,joe,mary
group.ShippingCDE=admin,mary
group.UpdateTemplate=admin,joe,mary,guest,wliSystem
group.adapter=admin,wlcSamplesUser,joe,hub,system,mary,guest,wliSystem
group.adminGroup=admin,joe,mary
```

A WebLogic Integration サンプル コンフィグレーション ファイル

```
group.wlcSamplesGroup=wlcSamplesUser, guest, wlisystem
```

```
group.wlpiAdministrators=admin, wlcSamplesUser, joe, wlisystem, system, mary, guest  
group.wlpiUsers=admin, joe, wlisystem, system, mary, guest
```

```
user.admin=0xab3a488db0652704287970cdf97854812f00a77b  
user.hub=0x6e4b0d35e609e96e749330f5086bb21f08a013d5  
user.joe=0xa078cb45e6f6c4eefdd1f14495ff739b5536904c  
user.mary=0xa078cb45e6f6c4eefdd1f14495ff739b5536904c  
user.system=0xab3a488db0652704287970cdf97854812f00a77b  
user.wlcSamplesUser=0x010e807f5525398753170afd0cf8694241286bc  
user.wlisystem=0x8510e22f04b4a7ebde6837902aa590a31e234be2
```

B WebLogic Integration コマンド

この節では、システム管理者および開発者を対象に、主要な **WebLogic Integration** コマンドについて説明します。この節は、次のトピックで構成されます。

- コマンド リスト – アルファベット順のコマンド リスト。詳細な説明に迅速にアクセスできるようにハイパーリンクを付けています。
- 環境変数 – コマンドにより設定または参照される環境変数のリスト。アルファベット順に並べ、簡単な説明を付けています。
- コマンドの説明 – コマンドを詳細に説明する参照ページ。

コマンド リスト

次の表に、ここで説明する各コマンドの名前と簡単な説明を示します。

表 B-1 WebLogic Integration コマンド

コマンド名	機能
bulkimporter	JAR (Java Archive) ファイルからワークフロー オブジェクトをインポートする。
bulkloader	リポジトリ データをインポート、エクスポート、または削除する。
bulkmigrator	リポジトリ データ ファイルを移行する。
checkdomain	ドメインを検証し、呼び出しプロセスのために WLI_DOMAIN を設定する。
createdb	リポジトリ テーブルを作成し、システム データをロードする。
createmssql createsybase	Microsoft SQL または Sybase データベースを作成し、そのデータベースでスクリプトを実行する。
domainname	setdomain コマンドで設定された現在のドメインを設定し、確認する。
fb	Format Builder を呼び出す。
launchSamplesPage	デフォルトの Web ブラウザで、サンプル起動ページを開く。
mdbgenerator	デプロイ可能な JAR ファイルを生成して、カスタムキューでリスンを行う。
migratedb	データベースを、WebLogic Integration 2.1 から 7.0 に移行する。
PIMigrator	ワークフロー テンプレートを移行する。

表 B-1 WebLogic Integration コマンド (続き)

コマンド名	機能
RunSamples	サンプルドメインのコンフィグレーションを行い、サンプルドメインで WebLogic Server を起動する。
setdbdata	データベース固有の変数を、現在のプロセスにロードする。
setdbtype	ドメインのデータベース タイプを設定する。
setDBVars setDBVarsExt	ドメインの基本データベース変数および拡張データベース変数を設定する。
setdomain	現在のドメインを設定する。
setenv	共通の WebLogic Integration 環境変数を設定する。
SetDomainTypeData SetSampleData	ドメイン固有の環境変数を設定する。
setupAppView setupEDIAppView	サンプルアダプタまたは EDI アダプタのアプリケーションビューを設定する
showdomain	現在のドメインを検証し、表示する。
startAiconsole startB2bconsole	Application View Console または B2B Console を起動する。
startpointbasecommander	PointBase 管理ツールを起動する。
startWeblogic	WebLogic Server を起動して、 WebLogic Integration リソースをデプロイする。
stopWeblogic	WebLogic Server をシャットダウンする。
studio	WebLogic Integration Studio を起動する。
switchdb	現在のドメイン用にコンフィグレーションが行われているデータベースを更新する。
wliconfig	WebLogic Integration データベース ウィザードを起動する。

B WebLogic Integration コマンド

表 B-1 WebLogic Integration コマンド (続き)

コマンド名	機能
worklist	WebLogic Integration Worklist を起動する。

環境変数

次の表に、WebLogic Integration コマンドにより参照される各環境変数の名前と簡単な説明を示します。

注意： このガイドでは、斜体の文字列 *DOMAIN_NAME* は、パス名のプレースホルダとして使用されます。具体的には、このドメインのルートフルパス名を意味します。斜体の等幅フォント（たとえば、*WLI_HOME*）で示される他の用語は環境変数を示しますが、*DOMAIN_HOME* は、環境変数ではありません。

表 B-2 WebLogic Integration の環境変数

変数	説明
CP	<i>SetDomainTypeData</i> コマンドにより設定され、 <i>setupAppView</i> および <i>setupEDIAppView</i> コマンドにより参照されるクラスパス。この変数は、 <i>fb</i> 、 <i>worklist</i> 、および <i>studio</i> コマンドにより、リセットされるがエクスポートされることはない。
CMNCP	<i>SetDomainTypeData</i> コマンドにより設定される共通のクラスパス。たとえば、 <i>SetSampleData</i> コマンドでは、CMNCP は、次のように設定される。 <code>CMNCP=\$WLI_SAMPLES_HOME/lib/samples.jar;\$WLI_HOME/lib/wli-aux.jar</code>
DB_CATALOG	データベース カタログの名前。 この表の <i>DB_PINGTABLE</i> 環境変数ならびに <i>setDBVarsExt</i> コマンドおよび <i>setupAppView</i> コマンドの説明を参照。
DB_CODESET	使用されているエンコーディングの種類。エンコーディングが <i>US-ASCII</i> または <i>UTF-8</i> の場合、この環境変数は設定されません。 <i>setDBVars</i> コマンドを参照。

表 B-2 WebLogic Integration の環境変数 (続き)

変数	説明
DB_COMMAND	<p>データベースのクライアントの呼び出しに必要なコマンド ライン。たとえば、現在のドメインが Oracle データベースを使用するようにコンフィグレーションされている場合、DB_COMMAND は、次のように設定される。</p> <pre>DB_COMMAND="sqlplus username/password@oracle_sid @"</pre> <p>createdb コマンド、setdbdata コマンド、および setDBVars コマンドを参照。</p>
DB_CREATE	<p>データベース固有のコマンド スクリプトを呼び出す。指定されたデータベースが存在しない場合、このコマンドはそのデータベースを作成する。createdb コマンド、setdbdata コマンド、および setDBVars コマンドを参照。</p>
DB_DIR	<p>現在のドメインの setDBVars コマンドおよび setDBVarsExt コマンドの位置。たとえば、現在のドメインが Oracle データベースを使用するようにコンフィグレーションされている場合、この変数は、次のように設定される。</p> <pre>DB_DIR=\$DBINFO/oracle</pre> <p>この表の DBINFO 環境変数ならびに setdbdata コマンドの説明を参照。</p>
DB_DRIVER	<p>コンフィグレーション済みデータベースへの接続に必要な JDBC ドライバ。たとえば、現在のドメインが Oracle データベースを使用するようにコンフィグレーションされている場合、DB_DRIVER は、次のように設定される。</p> <pre>DB_DRIVER="oracle.jdbc.driver.OracleDriver"</pre> <p>bulkloader コマンド、migratedb コマンド、および setDBVars コマンドを参照。</p>

表 B-2 WebLogic Integration の環境変数（続き）

変数	説明
DB_JVMARGS	<p>ドメインが PointBase データベースを使用するようにコンフィグレーションされている場合、この変数は次のように設定される。</p> <pre>DB_JVMARGS="-Dtransaction.isolationlevel=TRANSACTION_READ_COMMITTED -Ddatabase.pagesize=4096 -Dcache.size=2063 -Dsort.size=1024"</pre> <p>Microsoft SQL、Sybase、または Oracle データベースがコンフィグレーションされている場合、DB_JVMARGS は設定されない。</p> <p>bulkloader、createdb、migratedb、startWeblogic、stopWeblogic、setDBVars、startpointbasecommander、setupAppView、および setupEDIAppView コマンドを参照。</p>
DB_PASSWORD	<p>現在のドメイン用にコンフィグレーションされているデータベースへの接続に必要なパスワード。bulkloader、migratedb、および setDBVars コマンドを参照。</p>
DB_PINGTABLE	<p>データベース内のテーブルを指定する。たとえば、Windows システム上のサンプルドメイン用に、Microsoft SQL データベースがコンフィグレーションされている場合、DB_PINGTABLE は、setDBVarsExt コマンドにより次のように設定される。</p> <pre>DB_CATALOG=WLIDB DB_SCHEMA=dbo DB_PINGTABLE=%DB_CATALOG%.%DB_SCHEMA%.EVENT</pre> <p>次に、DB_PINGTABLE 変数が、サンプルアプリケーションビューの設定に使用される setupAppView コマンドにより、次のように呼び出される。</p> <pre>%JAVA_HOME%\bin\java . . . -cp %CP% examples.wlis.wlai.WLISAppViewDeployer . . . -pingTable %DB_PINGTABLE% . . .</pre> <p>(実際の java コマンドの一部のみを表示)。</p>

表 B-2 WebLogic Integration の環境変数 (続き)

変数	説明
DB_PROPERTIES	<p>switchdb コマンドとともに使用され、現在のドメインの JDBC 接続プールプロパティを更新する。たとえば、現在のドメインが Oracle データベースを使用するようにコンフィグレーションされている場合、DB_PROPERTIES は、次のように設定される。</p> <p>UNIX の場合 DB_PROPERTIES="user=\$DB_USER;password=\$DB_PASSWORD"</p> <p>Windows の場合 DB_PROPERTIES="user=%DB_USER%;password=%DB_PASSWORD%"</p> <p>setdbdata コマンドおよび switchDB.xml ファイルを参照。</p>
DB_SCHEMA	<p>データベースのスキーマ名。Microsoft SQL および Sybase の場合、データベース名。Oracle の場合は、ユーザ名。デフォルトの PointBase データベースの場合、次の設定が使用される。</p> <p>DB_SCHEMA=PBPUBLIC</p> <p>この表の DB_PINGTABLE 環境変数ならびに setDBVarsExt コマンドおよび setupAppView コマンドの説明を参照。</p>
DB_SCRIPT_DIR	<p>現在のドメイン用にコンフィグレーションされたデータベースで使用するスクリプトの位置。たとえば、現在のドメインのデータベースが Oracle データベースの場合、DB_SCRIPT_DIR は、次のように設定される。</p> <p>UNIX の場合 DB_SCRIPT_DIR="\$WLI_HOME/dbscripts/oracle"</p> <p>Windows の場合 DB_SCRIPT_DIR="%WLI_HOME%/dbscripts/oracle"</p> <p>createdb コマンドおよび setDBVars コマンドを参照。</p>
DB_SERVERNAME	<p>Microsoft SQL サーバの名前。</p> <p>migratedb コマンドおよび setDBVars コマンドを参照。</p>
DB_TYPE	<p>現在のドメインのデータベースの種類。有効な値は、oracle、mssql、sybase、および pointbase。</p> <p>createdb、migratedb、startpointbasecommander、startWeblogic、switchdb、setdbdata、setdbtype、および RunSamples コマンドを参照。</p>

表 B-2 WebLogic Integration の環境変数（続き）

変数	説明
DB_URL	<p>データベースドライバが要求する URL。この URL は、インストール時またはデータベース ウィザードを使用したデータベースのコンフィグレーション時に、入力した情報に基づいて構築される。たとえば、<code>DB_DRIVER="oracle.jdbc.driver.OracleDriver"</code> の場合、<code>DB_URL="jdbc:oracle:thin:@(description=(address=(host=host)(protocol=tcp)(port=1521))(connect_data=(sid=oraclesid)))"</code>。</p> <p><code>bulkloader</code> コマンド、<code>migratedb</code> コマンド、および <code>setDBVars</code> コマンドを参照。</p>
DB_USER	<p>現在のドメイン用にコンフィグレーションされているデータベースに接続するために必要なユーザ名。<code>bulkloader</code> コマンド、<code>migratedb</code> コマンド、および <code>setDBVars</code> コマンドを参照。</p>
DBINFO	<p>現在のドメイン用の <code>dbInfo</code> ディレクトリの位置。この変数は、次のように設定される。</p> <p>UNIX の場合</p> <pre>DBINFO=\$WLI_DOMAIN/dbInfo</pre> <p>Windows の場合</p> <pre>DBINFO=%WLI_DOMAIN%\dbInfo</pre> <p><code>setdbdata</code> コマンドおよび <code>switchdb</code> コマンドを参照。</p>
EVENTROUTER_URL	<p>ドメインのイベント ルータの URL。たとえば、<code>EVENTROUTER_URL="http://\$WLI_HOST_AND_PORT/DbmsEventRouter/EventRouter"</code></p> <p><code>SetDomainTypeData</code> コマンドおよび <code>setupAppView</code> コマンドを参照。</p>
JAVA	<p><code>java</code> コマンドの位置。たとえば、WebLogic Server を Solaris システム上のデフォルト位置にインストールする場合、JAVA は次のように設定される。</p> <pre>JAVA=\$JAVA_BIN/java</pre> <p><code>PIMigrator</code> コマンドを参照。</p>

表 B-2 WebLogic Integration の環境変数 (続き)

変数	説明
JAVA_HOME	JDK (Java Development Kit) のインストール先のルート ディレクトリ。たとえば、WebLogic Server を Windows システム上のデフォルト位置にインストールする場合、JAVA_HOME は次のように設定される。 JAVA_HOME=C:\bea\jdk131 bulkimporter、bulkloader、bulkmigrator、createdb、fb、mdbgenerator、migratedb、setenv、startpointbasecommander、studio、switchdb、wliconfig、worklist、startWeblogic、stopWeblogic、setupAppView、setupEDIAppView、および RunSamples コマンドを参照。
JAVA_OPTIONS	java コマンドの実行時オプション。
JAVACLASSPATH	bulkmigrator コマンドにより、呼び出されるクラスパス変数。
JDBC_DATASOURCE_JN DI_NAME	ドメインの JDBC データ ソース JNDI 名。たとえば、 JDBC_DATASOURCE_JNDI_NAME=WLAI_DataSource SetDomainTypeData コマンドおよび setupAppView コマンドを参照。
POINTBASE_HOME	PointBase データベースを使用するようにコンフィグレーションされたドメインの場合、この変数により、PointBase データベースの位置が次のように設定される。 \$WLI_DOMAIN/dbInfo/pointbase setDBVars コマンド、startpointbasecommander コマンド、および creatdb コマンドを参照。
SAMPLES_HOME	WebLogic Platform サンプルディレクトリ。たとえば、WebLogic Platform を Windows システム上のデフォルト位置にインストールする場合、SAMPLES_HOME は次のように設定される。 SAMPLES_HOME=C:\bea\weblogic700\samples createdb、setenv、startWeblogic、stopWeblogic、SetDomainTypeData、setDBVars、および RunSamples コマンドを参照。

表 B-2 WebLogic Integration の環境変数（続き）

変数	説明
SQLARGS	<p>Microsoft SQL または Sybase データベースにより要求されるオプションセット。たとえば、現在のドメイン用に Sybase データベースがコンフィグレーションされている場合、SQLARGS は、次のように設定される。</p> <pre>SQLARGS="-b -e -n -S servername -U username -P \"password\""</pre> <p>setDBVars コマンド、createmssql コマンド、および createsybase コマンドを参照。</p>
SVRCP	<p>SetDomainTypeData コマンドにより設定されるサーバクラスパス。たとえば、SetSampleData コマンドを実行する場合、SVRCP は、次のように設定される。</p> <pre>SVRCP=\$WLI\$SERVERCP:\$CMNCP</pre>
WL_HOME	<p>WebLogic Server のインストールディレクトリ。たとえば、WebLogic Platform を Windows システム上のデフォルト位置にインストールする場合、WL_HOME は次のように設定される。</p> <pre>WL_HOME=C:\bea\weblogic700\server</pre> <p>bulkimporter、mdbgenerator、setenv、startWeblogic、stopWeblogic、および RunSamples コマンドを参照。</p>
WLI_DOMAIN	<p>現在設定されているドメイン。checkdomain、createdb、migratedb、setdbdata、setdomain、domainname、showdomain、switchdb、SetDomainTypeData、および startWeblogic コマンドを参照。</p>
WLI_HOME	<p>WebLogic Integration のインストールディレクトリ。たとえば、WebLogic Platform を Windows システム上のデフォルト位置にインストールする場合、WLI_HOME は次のように設定される。</p> <pre>WLI_HOME=C:\bea\weblogic700\integration.</pre> <p>bulkimporter、bulkloader、checkdomain、createdb、fb、mdbgenerator、migratedb、RunSamples、setdomain、setenv、SetDomainTypeData、setupAppView、setupEDIAppView、showdomain、startpointbasecommander、startWeblogic、stopWeblogic、studio、switchdb、および worklist コマンドを参照。</p>

B WebLogic Integration コマンド

表 B-2 WebLogic Integration の環境変数 (続き)

変数	説明
WLI_SAMPLES_HOME	ドメインのルート。たとえば、 WLI_SAMPLES_HOME=\$SAMPLES_HOME/integration/config/samples SetDomainTypeData コマンドを参照。
WLICOMMONCP	WebLogic Integration 共通クラスパス (クライアントおよびサーバ用) setenv コマンドを参照。
WLICP	WebLogic Integration クライアント クラスパス。bulkloader、fb、migratedb、setenv、PIMigrator、studio、switchdb、worklist、SetDomainTypeData、setDBVars、および RunSamples コマンドを参照。
WLIS_HOST_AND_PORT	ドメイン管理サーバのホストおよびリスン ポート。たとえば、 WLIS_HOST_AND_PORT=localhost:7001 SetDomainTypeData、setupAppView、および setupEDIAppView コマンドを参照。
WLIS_PASSWORD	ドメインの WebLogic Server パスワード。たとえば、 WLIS_USER_NAME=system SetDomainTypeData、setupAppView、および setupEDIAppView コマンドを参照。
WLIS_USER_NAME	ドメインの WebLogic Server ユーザ名。たとえば、 WLIS_PASSWORD=security SetDomainTypeData、setupAppView、および setupEDIAppView コマンドを参照。
WLISERVERCP	WebLogic Integration サーバクラスパス。setenv、SetDomainTypeData コマンドおよび stopWeblogic コマンドを参照。

コマンドの説明

この節では、WebLogic Integration コマンドをアルファベット順に説明します。次の図に、各コマンドを説明する際の形式を簡単に示します。

図 3-14 コマンド説明の形式

コマンド名 — **bulkloader**

名前 bulkloader — リポジトリ データをインポート、エクスポート、削除する

構文 bulkloader *cfg_file*
bulkloader -load [-txAll] [-initDB] *data_file*

説明 bulkloader コマンドを使用すると、B2B コンフィグレーション データのリポジトリからのインポート、エクスポート、および削除を実行できます。 *cfg_file* の内容は、実行すべき操作を指定し、データベースアクセス情報を提供し、操作に必要であれば、リポジトリの更新に使用する XML ファイルの名前を指定します。

Bulk Loader の詳細、および *cfg_file* と *data_file* ファイルの内容については、『*Administering B2B Integration*』の「Bulk Loader の使い方」を参照してください。

オプション 次の表では、bulkloader コマンドのオプションについて概要を説明します。

表 B-3 bulkloader のオプション

オプション	説明
-load	<i>data_file</i> データを現在のドメインにコンフィグレーションされたデータベースにインポートする。
-initDB	データのインポート前にテーブルをすべて空にする。

UNIX および Windows システムでのコマンドの位置と関連ファイル

UNIX ファイル \$WLI_HOME/bulkloader
\$WLI_HOME/bin/setdbdata
\$WLI_HOME/setenv.sh

Windows ファイル %WLI_HOME%\bin\bulkloader.cmd
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\setenv.cmd

環境 このコマンドは次の環境変数を参照します。

DB_JVMARGS
JAVA_HOME
WLI_HOME
WLICP

-load オプションを使って実行すると、次の環境変数が参照されます。

コマンドの簡単な説明

コマンドの構文

コマンドオプションの説明

このコマンドにより設定または参照される環境変数

bulkimporter

名前 bulkimporter - JAR ファイルからワークフロー オブジェクトをインポートします。

構文 `bulkimporter -user bpm_user -password bpm_password -url wls_server_url -orgid org_name {-file file_name |-dir full_dir_path | -dir full_dir_path -file file_name} [-activate] [-packagepwd password] [-help] [-quiet]`

説明 bulkimporter コマンドを使用すると、WebLogic Integration Studio を使用しないで、コマンド ラインからワークフロー パッケージをインポートできます。ワークフロー パッケージには、次のワークフロー オブジェクトが含まれます。

- テンプレート定義
- ビジネス オペレーション
- ビジネス カレンダー
- イベント キー テーブル
- XML リポジトリ アイテム

オーガニゼーション、ユーザ、およびロールはインポートできません。

コマンド ラインでは、1 つ以上の jar ファイルを指定する必要がありますが、1 回のコマンドで多くの jar ファイルをインポートすることも可能です。

- `-file file_name` オプションを使用して 1 つの jar ファイル。
- `-dir full_dir_path` オプションを使用して、1 つのディレクトリ内のすべての jar ファイル。
- 1 つのコマンド ラインで、`-file file_name` オプションと `-dir full_dir_path` オプションを使用して、1 つの Jar ファイルと 1 つのディレクトリ内のすべての jar ファイル。

`-quiet` オプションを指定しない場合は、WebLogic Integration にインポートされるすべてのオブジェクトのリストが表示されます。

WebLogic Integration Studio を使用して、ワークフロー オブジェクトをインポートすることもできます。詳細については、『*WebLogic Integration Studio ユーザーズガイド*』の「ワークフロー パッケージのインポートとエクスポート」を参照してください。

オプション 次の表に、bulkimporter コマンドのオプションを簡単に示します。

表 B-3 bulkimporter のオプション

オプション	説明
<code>-user bpm_user</code>	WebLogic Integration BPM ユーザ名を指定。Studio にログインするとき使用するのと同じパスワード。デフォルトの BPM ユーザのリストについては、3-50 ページの「BPM ユーザグループおよびパーミッション」を参照。このオプションは必須。
<code>-password bpm_password</code>	<code>-user</code> オプションで指定した BPM ユーザ名のパスワードを指定。このオプションは必須。
<code>-url wls_server_url</code>	WebLogic Server URL を指定。デフォルト ポート番号の 7001 を使用するローカルマシン上の WebLogic Server のインスタンスに接続する場合は、 <code>wls_server_url</code> を <code>t3://localhost:7001</code> に設定。このオプションは必須。
<code>-orgid org_name</code>	BPM オーガニゼーション名を指定。デフォルトの BPM オーガニゼーションのリストについては、オプションを参照。このオプションは必須。
<code>-file file_name</code>	インポートするワークフロー オブジェクトを含む jar ファイル。ファイル名の最後には、必ず、拡張子 <code>.jar</code> を付ける。この <code>-dir full_dir_path</code> オプションを指定するか、あるいは <code>-file file_name</code> と <code>-dir full_dir_path</code> オプションの両方を一緒に指定する必要がある。
<code>-dir full_dir_path</code>	インポートする jar ファイルを含むディレクトリのフルディレクトリパス。このディレクトリに配置されている拡張子 <code>.jar</code> で終わるすべてのファイルがインポートされる。この <code>-file full_dir_path</code> オプションを指定するか、あるいは <code>-file file_name</code> オプションと <code>-dir full_dir_path</code> オプションの両方を一緒に指定する必要がある。
<code>-activate</code>	インポート後、ワークフローをアクティブにする。
<code>-packagepwd password</code>	ワークフロー パッケージがパスワード保護されている場合、ワークフローパスワードを指定する。
<code>-help</code>	サポートされているすべてのオプションを表示する。

表 B-3 bulkimporter のオプション (続き)

オプション	説明
-quiet	コマンドの実行結果を表示しない (コマンド実行時のメッセージは表示する)。

UNIX ファイル \$WLI_HOME/bin/bulkimporter

Windows ファイル %WLI_HOME%\bin\bulkimporter.cmd

環境 このコマンドは、次の環境変数を参照します。

WL_HOME
WLI_HOME
JAVA_HOME

WLI_HOME を設定しない場合、このコマンドは、setenv コマンドを実行します。

bulkloader

名前 bulkloader -リポジトリ データをインポート、エクスポート、または削除します。

構文 bulkloader [-wlibc] *cfg_file*
bulkloader -load [-txAll] [-initDB][*-wlibc*] *data_file*

説明 bulkloader コマンドを使用すると、リポジトリの **B2B** コンフィグレーション データをインポート、エクスポート、または削除できます。*cfg_file* は、行うアクションを指定し、データベースアクセス情報を提供します。また、操作により必要な場合は、リポジトリの更新に使用する **XML** ファイル名を提供します。

-load オプションを使用する場合、*data_file* に、リポジトリにインポートする **XML** ファイルを指定します。データベース情報は、現在のドメインの `setDBVars` コマンドの設定から取得されます。現在のドメインは、`setdomain` コマンドで設定されます。現在のドメインを確認するには、`checkdomain` コマンドを使用します。

Bulk Loader の詳細ならびに *cfg_file* および *data_file* ファイルの内容については、『*B2B Integration 管理ガイド*』の「**Bulk Loader** の操作」を参照してください。

注意： デフォルトでは、**Bulk Loader** は、**B2B** エンジンの実行中にデータをインポート、エクスポート、削除しません。**B2B** エンジンが何らかの理由で異常終了した場合、**Bulk Loader** は失敗し、次のエラー メッセージを表示します。

```
ERROR: WLC is still running or was shut down abnormally.
```

Bulk Loader コマンド ファイルを修正して、このエラーを無視しコマンドを強制実行する手順については、『*B2B Integration 管理ガイド*』の「**Bulk Loader** の操作」の「**Bulkloader** の強制実行」を参照してください。

オプション 次の表に、bulkloader コマンドのオプションを簡単に示します。

表 B-4 bulkloader のオプション

オプション	説明
-wllibc	軽量クライアント（WebLogic Integration – Business Connect）を使用するトレーディング パートナにより作成された XML ファイルをインポートする際には、-wllibc フラグを使用する。
-load	<i>data_file</i> データを現在のドメイン用にコンフィグレーションされたデータベースにインポートする。
-txAll	1 回のトランザクションでデータをインポートする。このオプションを指定しない場合、各エンティティをインポートまたは削除したあとに、トランザクションが開始される。エンティティには、トレーディング パートナ、会話定義、コラボレーション アグリーメント、ビジネス プロトコル定義、およびロジック プラグインがある。エンティティのいずれかに対するトランザクション中に無効なデータが検出されると、インポートは現行のトランザクションについてのみロールバックされ、次のトランザクションでのインポートは続行される。
-initDB	リポジトリのすべてのテーブルを空にしてから、データをインポートする。 警告： -initDB オプションを指定すると、データベース内の既存のデータが破壊されます。ロードするデータに必要なシステム データが含まれない場合、このオプションを指定するとシステムを使用できなくなります。データは WebLogic Integration によって作成されます。 Business Connect トレーディング パートナによって作成されるデータには、システム データが含まれないので、このオプションを -wllibc オプションと一緒に使用しないでください。データベース全体を再初期化する場合を除いて、-initDB を指定しないでください。

```
UNIX ファイル $WLI_HOME/bulkloader
                $WLI_HOME/bin/setdbdata
                $WLI_HOME/setenv.sh
```

Windows %WLI_HOME%\bin\bulkloader.cmd
ファイル %WLI_HOME%\bin\setdbdata.cmd
 %WLI_HOME%\setEnv.cmd

環境 このコマンドは、次の環境変数を参照します。

DB_JVMARGS
JAVA_HOME
WLI_HOME
WLICP

-load オプションを付けて実行する場合、このコマンドは次を参照します。

DB_DRIVER
DB_URL
DB_USER
DB_PASSWORD

WLI_HOME を設定しない場合、このコマンドは、setenv コマンドを実行します。

bulkmigrator

名前 bulkmigrator リポジトリのデータ ファイルを移行します。

構文 bulkmigrator *collaborate1.0_file* *wlintegration2.0_file*

説明 bulkmigrator コマンドは、WebLogic Collaborate 1.0.x (*collaborate1.0_file*) からエクスポートされた XML データ ファイルを WebLogic Integration 2.0 (*wlintegration2.0_file*) で必要な形式にコンバートします。このファイルを WebLogic Integration 7.0 にインポートできるようにするには、追加手順を実行する必要があります。

詳細については、『*BEA WebLogic Integration 移行ガイド*』の「移行の概要」の「WebLogic Integration 2.1 以前のリリースからの移行」を参照してください。

オプション なし

UNIX ファイル \$WLI_HOME/bin/bulkmigrator

Windows %WLI_HOME%\bin\bulkmigrator.cmd
ファイル

環境 このコマンドは、次の環境変数を参照します。

JAVA_HOME
JAVACLASSPATH

checkdomain

名前 checkdomain ドメインを検証し、呼び出しプロセスのために WLI_DOMAIN を設定します。

構文 checkdomain

説明 checkdomain コマンドは、setdomain および他のコマンド スクリプトと一緒に使用します。checkdomain コマンドは、他のコマンドの実行に必要なディレクトリおよびファイルがドメイン内にあることを検証し、ドメインが有効な場合は、domainname コマンドを実行して、呼び出しプロセスのために WLI_DOMAIN を設定します。

注意： domainname コマンドは、setdomain コマンドが実行されるとき、作成されます。このコマンドは、現在の WLI_DOMAIN 設定を保存します。

オプション なし

UNIX ファイル
\$WLI_HOME/bin/checkdomain
DOMAIN_HOME/checkdomain
\$WLI_HOME/bin/domainname
\$WLI_HOME/setenv.sh

Windows ファイル
%WLI_HOME%\bin\checkdomain.cmd
DOMAIN_HOME\checkdomain.cmd
%WLI_HOME%\bin\domainname.cmd
%WLI_HOME%\setEnv.cmd

環境 このコマンドは、WLI_DOMAIN を設定し、WLI_HOME を参照します。

WLI_HOME を設定しない場合、このコマンドは、setenv コマンドを実行します。

createdb

名前 createdb - リポジトリ テーブルを作成してシステム データをロードします。

構文 createdb [-f] [-noinit]

説明 createdb コマンドは、現在のドメイン用にコンフィグレーションされているデータベースで WebLogic Integration リポジトリ テーブルを作成し、必要に応じて、そのデータベースにシステム データをロードします。このコマンドは、setdbdata を呼び出して、現在のデータベース アクセス情報をロードし、bulkloader コマンドを呼び出して、データをロードします。

作成されるテーブルの内容は、現在のドメインの要件によって決まります（必要なテーブルの説明は、1-11 ページの「ドメイン コンフィグレーションの要件」の表 1-3 を参照）。たとえば、現在のドメインが B2B 機能をサポートしていない場合、B2B テーブルは作成されず、B2B システム データもロードされません。現在のドメインが WebLogic Integration サンプルドメインの場合、サンプル固有のテーブルが追加で作成されます。

注意： Microsoft SQL データベースのテーブルの作成に使用される osql クライアントは、UNIX では使用できないので、createdb コマンドを使用して、UNIX システムから Microsoft SQL データベースのテーブルを作成することはできません。

オプション 次の表に、createdb コマンドのオプションを簡単に示します。

表 B-5 createdb のオプション

オプション	説明
-f	確認を求めない。デフォルトでは、このコマンドは次に進む前にユーザに確認を求める。
-noInit	テーブルのみを作成。リポジトリにシステム データを格納しない。デフォルトでは、createdb は、システム データをリポジトリにロードする。

UNIX ファイル

```
$WLI_HOME/bin/createdb
$WLI_HOME/bin/setdbdata
$WLI_HOME/bin/bulkloader
$WLI_HOME/dbscripts/$DB_TYPE/REPOSITORY_SCHEMA_BASE.sql
$WLI_HOME/dbscripts/$DB_TYPE/REPOSITORY_SCHEMA.sql
$WLI_HOME/dbscripts/$DB_TYPE/PERSISTENCE_SCHEMA.sql
```

```

$WLI_HOME/dbscripts/$DB_TYPE/BPM_SCHEMA.sql
$SAMPLES_HOME/integration/samples/wlis/dbscripts/$DB_TYPE/WLIS_SCHEM
A.sql
$SAMPLES_HOME/integration/samples/e2e/dbscripts/$DB_TYPE/E2E_SCHEM
A.sql
$WLI_HOME/adapters/dbms/src/sql/msql/cust.sql
$WLI_HOME/adapters/dbms/src/sql/msql/evnt.sql
$WLI_HOME/adapters/dbms/src/sql/oracle/cust.sql
$WLI_HOME/adapters/dbms/src/sql/oracle/evnt.sql
$WLI_HOME/adapters/dbms/src/sql/pointbase/cust.sql
$WLI_HOME/adapters/dbms/src/sql/pointbase/evnt.sql
$WLI_HOME/adapters/dbms/src/sql/sybase/cust.sql
$WLI_HOME/adapters/dbms/src/sql/sybase/evnt.sql
$WLI_HOME/dbscripts/SystemRepData.xml
$WLI_HOME/setenv.sh

```

Windows ファイル

```

%WLI_HOME%\bin\createDB.cmd
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\bin\bulkloader.cmd
%WLI_HOME%\dbscripts\%DB_TYPE%\REPOSITORY_SCHEMA_BASE.sql
%WLI_HOME%\dbscripts\%DB_TYPE%\REPOSITORY_SCHEMA.sql
%WLI_HOME%\dbscripts\%DB_TYPE%\PERSISTENCE_SCHEMA.sql
%WLI_HOME%\dbscripts\%DB_TYPE%\BPM_SCHEMA.sql
%SAMPLES_HOME%\integration\samples\wlis\dbscripts\%DB_TYPE%\WLIS_SCH
EMA.sql
%SAMPLES_HOME%\integration\samples\e2e\dbscripts\%DB_TYPE%\E2E_SCHEM
A.sql
%WLI_HOME%\adapters\dbms\src\sql\msql\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\msql\evnt.sql
%WLI_HOME%\adapters\dbms\src\sql\oracle\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\oracle\evnt.sql
%WLI_HOME%\adapters\dbms\src\sql\pointbase\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\pointbase\evnt.sql
%WLI_HOME%\adapters\dbms\src\sql\sybase\cust.sql
%WLI_HOME%\adapters\dbms\src\sql\sybase\evnt.sql
%WLI_HOME%\dbscripts\SystemRepData.xml
%WLI_HOME%\setEnv.cmd

```

環境

createdb コマンドは、次の環境変数を参照します。

```

CLCP
DB_COMMAND
DB_CREATE
DB_JVMARGS
DB_PORT
DB_SCRIPT_DIR
DB_TYPE
JAVA_HOME
POINTBASE_HOME
SAMPLES_HOME
WLI_DOMAIN
WLI_HOME

```

WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

createmssql createsybase

名前 createmssql - Microsoft SQL データベースを作成し、そのデータベースでスクリプトを実行します。
createsybase - Sybase データベースを作成し、そのデータベースでスクリプトを実行します。

構文 createmssql *database_name sql_script*
createsybase *database_name sql_script*

説明 createmssql コマンドおよび createsybase コマンドは、指定されたデータベースが存在するかどうかをチェックします。データベースが存在する場合、指定した SQL スクリプトを実行します（この操作は、DB_COMMAND 環境変数を呼び出すのと同じです。ただしここでは、指定された *sql_script* も実行されます）。データベースが存在しない場合、作成され、指定したスクリプトは、新しく作成されたデータベースで実行されます。

たとえば、現在のドメイン用に、Microsoft SQL データベースの WLIDB がコンフィグレーションされている場合、%WLI_DOMAIN%\dbInfo\mssql\setDBVars コマンド ファイルの DB_CREATE 環境変数は次のように設定されます。

```
DB_CREATE=call %WLI_HOME%\dbscripts\mssql\createmssql WLIDB
```

DB_CREATE 環境変数は、createdb コマンドにより次のように呼び出されます。

```
%DB_CREATE%REPOSITORY_SCHEMA_BASE.sql
```

WLIDB データベースがすでに存在する場合、DB_CREATE 環境変数を呼び出すということは、DB_COMMAND 環境変数を呼び出すことと同じです。WLIDB データベースが存在しない場合、作成され、指定したスクリプトは、作成されたデータベースで実行されます。

注意: UNIX システムでは、osql クライアントを使用できないので、createmssql コマンドは使用できません。UNIX システムで、createdb コマンドを使用して Microsoft SQL データベースのテーブルを作成することはできません。

オプション なし

UNIX ファイル \$WLI_HOME/dbscripts/sybase/createsybase
ル

Windows %WLI_HOME%\dbscripts\mssql\createmssql.cmd
ファイル %WLI_HOME%\dbscripts\sybase\createsybase.cmd

環境 これらのコマンドは、次の環境変数を参照します。

SQLARGS
DB_COMMAND

domainname

名前	domainname - setdomain コマンドで設定された現在のドメインを設定し、確認します。
構文	domainname
説明	マシンが生成するこのコマンドは、現在のドメイン設定を保存します。このファイルは、setdomain コマンドの実行時に、生成されます。checkdomain コマンドは、このファイルの設定を使用して、ドメインを確認し、呼び出しプロセス用に WLI_DOMAIN を設定します。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/domainname
Windows ファイル	%WLI_HOME%\bin\domainname.cmd
環境	このコマンドは、WLI_DOMAIN 環境変数を設定します。

fb

名前	fb - Format Builder を起動します。
構文	fb
説明	このコマンドは、Format Builder クライアントを起動します。Windows の [スタート] メニューには、fb.cmd コマンドのショートカットが用意されています。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/fb \$WLI_HOME/setenv.sh
Windows ファイル	%WLI_HOME%\bin\fb.cmd %WLI_HOME%\setEnv.cmd
環境	fb コマンドは、次のように、CP 環境変数を設定しますが、エクスポートは行いません。 fb コマンドは、次の環境変数を参照します。 WLICP WLI_HOME JAVA_HOME WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

launchSamplesPage

名前 launchSamplesPage デフォルトの Web ブラウザで、サンプル起動ページを開きます。

構文 launchSamplesPage

説明 このコマンドは、`http://localhost:7001/index.html` を開きます。ローカルマシンで、サンプルドメインの管理サーバが実行されている場合、このコマンドが開くページからサンプルを実行できます。

注意： UNIX システムでは、システム パス変数に実行可能なブラウザ `netscape` を指定する必要があります

オプション なし

UNIX ファイル `$SAMPLES_HOME/integration/samples/bin/launchSamplesPage`

**Windows
ファイル** `%SAMPLES_HOME%\integration\samples\bin\launchSamplesPage.cmd`

環境 参照または設定される環境変数はありません。

mdbgenerator

名前 mdbgenerator ーデプロイ可能な JAR ファイルを生成して、カスタム キューでリスンを行います。

構文 mdbgenerator -queue *queue_name* [-min *number*] [-max *number*] [-order *number*] [-transact] [-validate] [-timeout *seconds*] [-help]

説明 MDB (Message Driven Bean) 生成ユーティリティ (mdbgenerator コマンド) は、デプロイ可能な JAR ファイルを生成します。このファイルは、カスタム JMS (Java Message Service) キューでのリスンに使用できます。

オプション 次の表に、mdbgenerator コマンドのオプションを簡単に示します。

表 B-6 mdbgenerator のオプション

オプション	説明
-queue <i>queue_name</i>	MDB を生成するカスタム キュー名。このオプションは必須。
-min <i>number</i>	順序付けされていないリスナの最少数。
-max <i>number</i>	順序付けされていないリスナの最大数。
-order <i>number</i>	順序付けされたリスナの数。31 以下の素数でなければならない。
-transact	トランザクションを必須に設定する。
-validate	XML 検証を有効にする。
-timeout <i>seconds</i>	トランザクション タイムアウトを設定する (秒単位)。このオプションは、デフォルトで、30 秒に設定されている。 transact フラグが設定されていない場合にのみ、この値を使用できる。transact フラグが設定されている場合、ユーティリティは、WebLogic Server トランザクション タイムアウト値を使用する。この値は、Administration Console で設定でき、デフォルトでは 30。
-help	コマンドの用法を示す構文を表示する。

B WebLogic Integration コマンド

UNIX ファイル \$WLI_HOME/bin/mdbgenerator
 \$WLI_HOME/setenv.sh

Windows %WLI_HOME%\bin\mdbgenerator
ファイル %WLI_HOME%\setEnv.cmd

環境 このコマンドは、次の環境変数を参照します。

WLI_HOME
WL_HOME
JAVA_HOME

WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

migratedb

名前 migratedb – WebLogic Integration 2.1 または WebLogic Integration 2.1 Service Pack 1 (SP1) のデータベースを WebLogic Integration 7.0 に移行します。

構文 migratedb

説明 migratedb コマンドは、現在のドメインでコンフィグレーションされたデータベースの移行に必要なスクリプトを作成し、このスクリプトをデータベースの対して実行します。このコマンドは、WebLogic Integration 2.1 または WebLogic Integration 2.1 SP1 データベース テーブルとデータベース スキーマを更新します。WebLogic Integration 7.0 との互換性を持つには、この手順が必要です。

詳細については、『*BEA WebLogic Integration 移行ガイド*』を参照してください。

オプション なし

UNIX ファイル

```
$WLI_HOME/bin/migratedb
$WLI_HOME/bin/checkdomain
$WLI_HOME/bin/setdbdata
$WLI_HOME/bin/bulkloader
$WLI_HOME/setenv.sh
$WLI_HOME%/dbscripts/migrate/SystemRepData.xml
```

Windows ファイル

```
%WLI_HOME%\bin\migratedb.cmd
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\bin\bulkloader.cmd
%WLI_HOME%\setEnv.cmd
%WLI_HOME%\dbscripts\migrate\SystemRepData.xml
```

環境 このコマンドは、次の環境変数を参照します。

```
DB_DRIVER
DB_NAME
DB_PASSWORD
DB_SERVERNAME
DB_TYPE
DB_URL
DB_USER
JAVA_HOME
WLI_DOMAIN
WLI_HOME
WLICP
```

WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

PIMigrator

名前	PIMigrator -ワークフロー テンプレートを移行します。
構文	PIMigrator <i>InputTemplate.xml</i> <i>OutputTemplate.xml</i>
説明	PIMigrator コマンドは、Collaborate 1.0 で使用するよう作成されたテンプレートを、WebLogic Integration 2.0/2.1 で必要な書式に更新します。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/PIMigrator \$WLI_HOME/lib/dtd/PITemplateMigration.xsl
Windows ファイル	%WLI_HOME%\bin\PIMigrator.cmd %WLI_HOME%\lib\dtd\PITemplateMigration.xsl
環境	このコマンドは、次の環境変数を参照します。 JAVA WLC_HOME WLICP

RunSamples

名前 RunSamples - サンプルデータベースのコンフィグレーションを行い、サンプルドメインで **WebLogic Server** を起動します。

構文 RunSamples [option . . .]

説明 RunSamples コマンドは、サンプルドメインを対象に次のタスクを実行します。

- サンプル データベース リポジトリ内に必要なリポジトリ テーブルを作成する。1-11 ページの「ドメイン コンフィグレーションの要件」の表 1-3 を参照。
- **B2B Integration** システム データおよびサンプル コンフィグレーション データをロードする。
- サンプルドメインで **WebLogic Integration** を起動する。
- **End-to-End** サンプル アプリケーションのアプリケーション ビューをコンフィグレーションする。
- サンプル ワークフローを含むワークフロー パッケージをインポートする。
- デフォルト **Web** ブラウザを起動し、サンプル起動ページを表示する。

デフォルト オプション (all) を使用するか、あるいはオプションを使用しないで RunSamples コマンドを実行する場合、このコマンドの動作は、サンプルドメイン用にコンフィグレーションされたデータベースに対して初めて実行されているかどうかにより異なります。詳細については、1-15 ページの「サンプルドメインのコンフィグレーションと起動」を参照してください。

オプション 次の表に、RunSamples コマンドのオプションを簡単に示します。

表 B-7 RunSamples のオプション

オプション	説明
all	サンプルの実行に必要なすべての手順を実行する。デフォルトでは、このオプションが呼び出される。
BulkLoader	サンプルの Bulk Loader を実行する。
CreateDB	サンプルのデータベースを作成する。
ImportPackages	BPM パッケージをサーバにインポートする。

表 B-7 RunSamples のオプション (続き)

オプション	説明
SampleDB	サンプル固有のデータベース テーブルを作成する。
SetupAppView	サンプルアダプタのアプリケーション ビューをセットアップする。
StartServer	サンプル サーバを起動する。

UNIX ファイル

\$SAMPLES_HOME/integration/samples/bin/RunSamples
 \$WLI_HOME/bin/setdomain
 \$SAMPLES_HOME/integration/samples/bin/SetSampleData

Windows
ファイル

%SAMPLES_HOME%\integration\samples\bin\RunSamples.cmd
 %WLI_HOME%\bin\setdomain.cmd
 %SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd

環境

このコマンドは、次の環境変数を参照します。

WLI_HOME
 WLICP
 WL_HOME
 JAVA_HOME
 DB_TYPE

WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

setdbdata

名前 setdbdata - データベース固有の変数を、現在のプロセスにロードします。

構文 setdbdata

説明 setdbdata コマンドは、次の処理により、データベース固有の変数を、現在のプロセスにロードします。

- checkdomain コマンドを呼び出して、現在のドメイン情報をロードする。
- setdbtype コマンドを呼び出して、データベースタイプを設定する。
setdbtype コマンドは、コンフィグレーションウィザードを使用してドメインを作成したとき、switchdb コマンドを実行したとき、または WebLogic Integration データベースウィザード (wliconfig コマンド) を使用してドメインのデータベースを更新するとき入力される情報を基に生成される。
- setDBVars コマンドを実行する。
- 存在する場合、setDBVarsExt コマンドを実行する (setDBVarsExt コマンドは省略可能。このコマンドが存在しなくても、エラーは発生しない)。

オプション なし

UNIX ファイル

```
$WLI_HOME/bin/setdbdata
$WLI_HOME/bin/checkdomain
DOMAIN_HOME/setdbdata
DOMAIN_HOME/checkdomain
DOMAIN_HOME/dbInfo/$DB_TYPE/setDBVars
DOMAIN_HOME/dbInfo/$DB_TYPE/setDBVarsExt
DOMAIN_HOME/dbInfo/setdbtype
```

Windows ファイル

```
%WLI_HOME%\bin\setdbdata.cmd
%WLI_HOME%\bin\checkdomain.cmd
DOMAIN_HOME\setdbdata.cmd
DOMAIN_HOME\checkdomain.cmd
DOMAIN_HOME\dbInfo\%DB_TYPE%\setDBVars
DOMAIN_HOME\dbInfo\%DB_TYPE%\setDBVarsExt
DOMAIN_HOME\dbInfo\setdbtype
```

環境 setdbtype および setDBVars により設定される変数に加えて、setdbdata コマンドは、次の変数を設定します。

```
DBINFO
DB_CREATE
DB_DIR
DB_PROPERTIES
```

注意: setDBVars で定義されていない場合、DB_CREATE は、DB_COMMAND に設定されます。

setdbtype

名前	setdbtype	ードメインのデータベースタイプを設定します。
構文	setdbtype	
説明		このコマンドは、コンフィグレーションウィザードを使用してドメインを作成したとき、switchdb コマンドを実行したとき、または WebLogic Integration データベースウィザード (wliconfig コマンド) を使用してドメインのデータベースを更新するとき入力する情報を基に生成されます。
オプション	なし	
UNIX ファイル	DOMAIN_HOME/dbInfo/setdbtype	
Windows ファイル	DOMAIN_HOME\dbInfo\setdbtype.cmd	
環境		このコマンドは、DB_TYPE 環境変数を設定します。

setDBVars

setDBVarsExt

名前 setDBVarードメインの基本データベース変数を設定します。
setDBVarsExtードメインの拡張データベース変数を設定します。

構文 setDBVars
setDBVarsExt

説明 ドメインの setDBVars コマンドおよび setDBVarsExt コマンドは、コンフィグレーションウィザードを使用してドメインを作成したとき、switchdb コマンドを実行したとき、または WebLogic Integration データベースウィザード (wliconfig コマンド) を使用してドメイン用にデータベースを更新したとき入力した情報を基に更新されます。

オプション なし

UNIX ファイル \$WLI_DOMAIN/dbInfo/\$DB_TYPE/setDBVars
\$WLI_DOMAIN/dbInfo/\$DB_TYPE/setDBVarsExt

Windows ファイル %WLI_DOMAIN%\dbInfo\%DB_TYPE%\setDBVars
%WLI_DOMAIN%\dbInfo\%DB_TYPE%\setDBVarsExt

環境 setDBVars コマンドは、次の環境変数を設定します。

```
DB_JVMARGS
DB_USER
DB_PASSWORD
DB_CODESET
DB_SERVERNAME
DB_NAME
DB_URL
DB_DRIVER
DB_SCRIPT_DIR
SQLARGS
DB_COMMAND
DB_CREATE
CLCP
```

setDBVarsExt コマンドは、次の環境変数を設定します。

```
DB_CATALOG
DB_SCHEMA
DB_PINGTABLE
```

注意： 一部の変数は、データベースのタイプに固有です。たとえば、CLCP は、PointBase 専用の変数です。

setdomain

名前 setdomain - 現在のドメインを設定します。

構文 setdomain *domain_path*

ここで、*domain_path* は、ドメインの相対パス（現在のディレクトリを基準）です。たとえば、UNIX システム上で、\$BEA_HOME ディレクトリからコマンドを実行して現在のドメインを \$BEA_HOME/user_projects/mydomain に設定する場合、次のように入力します。

```
setdomain user_projects/mydomain
```

ドメインディレクトリ（たとえば、\$BEA_HOME/user_projects/mydomain ディレクトリ）からコマンドを実行する場合、*domain_path* を指定する必要はありません。

注意：最初に setenv コマンドを実行すると、setdomain コマンドは任意のディレクトリから実行できます。

説明 setdomain コマンドは、指定したディレクトリが有効なドメインかどうかを確認します。ドメインが有効な場合、このコマンドは、WLI_DOMAIN 環境変数を指定したドメインに設定して、domainname コマンドを作成し設定を保存します。現在のドメインを、checkdomain、setdbdata、RunSamples、および switchdb コマンド用に設定する必要があります。

オプション なし

UNIX ファイル
\$WLI_HOME/bin/setdomain
\$WLI_HOME/setenv.sh
\$WLI_HOME/bin/domainname

Windows
ファイル %WLI_HOME%\bin\setdomain.cmd
%WLI_HOME%\setEnv.cmd
%WLI_HOME%\bin\domainname.cmd

環境 setdomain コマンドは、WLI_DOMAIN 環境変数を設定し、WLI_HOME 環境変数を参照します。

WLI_HOME が設定されていない場合、このコマンドによって setenv コマンドが実行されます。

setenv

名前	setenv ー共通 WebLogic Integration 環境変数を設定します。
構文	setenv
説明	setenv コマンドは、基本 WebLogic Integration 環境を設定します。
オプション	なし
UNIX ファイル	\$WLI_HOME/setenv.sh \$WLI_HOME/bin/setenv
Windows ファイル	%WLI_HOME%\setEnv.cmd %WLI_HOME%\bin\setenv.cmd %WLI_HOME%\samples\bpm_api\commandline\setEnv.cmd %WLI_HOME%\samples\bpm_api\plugin\setEnv.cmd
環境	setenv コマンドは、次の環境変数を設定します。 BEA_HOME JAVA_HOME SAMPLES_HOME WL_HOME WLI_HOME WLICOMMONCP WLICP WLISERVERCP また、このコマンドは、%JAVA_HOME%\bin を既存の PATH 環境変数に追加します。

SetDomainTypeData

SetSampleData

名前	<p><code>SetDomainTypeData</code> - <code>DomainType</code> ドメインの環境変数を設定します。</p> <p><code>SetSampleData</code> - サンプルドメインの環境変数を設定します。</p>
構文	<pre>SetDomainTypeData SetSampleData</pre>
説明	<p>The <code>SetDomainTypeData</code> コマンドは、コンフィグレーション ウィザードで作成されたカスタムドメイン用の環境変数を設定します。</p> <p>ここで、<code>DomainType</code> の値は、カスタムドメインの作成に使用したコンフィグレーション ウィザード テンプレートによって異なります。たとえば、WebLogic Integration テンプレートを使用した場合、コマンド名は <code>SetWliDomainData</code> になります。</p> <p><code>SetSampleData</code> コマンドは、WebLogic Integration サンプルドメイン用の環境変数を設定します。</p>
オプション	なし
UNIX ファイル	<pre>DOMAIN_HOME/SetDomainTypeData \$SAMPLES_HOME/integration/samples/bin/SetSampleData \$WLI_HOME/setenv.sh \$WLI_HOME/bin/setdbdata</pre>
Windows ファイル	<pre>DOMAIN_HOME\SetDomainTypeData %SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd %WLI_HOME%\setEnv.cmd %WLI_HOME%\setdbdata.cmd</pre>
環境	<p>2つのコマンド (<code>SetDomainTypeData</code> および <code>SetSampleData</code>) は、カスタムドメインと WebLogic Integration サンプルドメイン用に、それぞれ次の環境変数を設定します。</p> <pre>WLI_SAMPLES_HOME (SetSampleData) WLI_DOMAIN_HOME (SetDomainTypeData) WLIS_HOST_AND_PORT WLIS_USER_NAME WLIS_PASSWORD JDBC_DATASOURCE_JNDI_NAME EVENTROUTER_URL CMNCP CP SVRCP</pre>

The `SetDomainTypeData` コマンドは、次の環境変数を参照します。

```
WLICP  
WLISERVERCP  
WLI_HOME  
CMNCP  
SVRCP  
WLI_DOMAIN_HOME  
WLIS_HOST_AND_PORT
```

setupAppView setupEDIAppView

名前	<p>setupAppView - サンプルアダプタのアプリケーションビューをセットアップします。</p> <p>setupEDIAppView - EDI アダプタのアプリケーションビューをセットアップします。</p>
構文	<pre>setupAppView setupEDIAppView</pre>
説明	<p>setupAppView コマンドは、サンプルアダプタのアプリケーションビュー (WLISAppView) をセットアップします。setupEDIAppView コマンドは、EDI アダプタのアプリケーションビュー (EDIAppView) をセットアップします。</p> <p>両方のコマンドが、SetSampleData コマンドを実行してから、アプリケーションビューをセットアップする java コマンドが発行されます。</p>
オプション	なし
UNIX ファイル	<pre> \$SAMPLES_HOME/integration/samples/bin/setupAppView \$SAMPLES_HOME/integration/samples/bin/setupEDIAppView \$SAMPLES_HOME/integration/samples/bin/SetSampleData \$WLI_HOME/setenv.sh</pre>
Windows ファイル	<pre> %SAMPLES_HOME%\integration\samples\bin\setupAppView.cmd %SAMPLES_HOME%\integration\samples\bin\setupEDIAppView.cmd %SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd %WLI_HOME%\setEnv.cmd</pre>
環境	<p>setupAppView コマンドおよび setupEDIAppView コマンドは、次の環境変数を参照します。</p> <pre> CP DB_JVMARGS JAVA_HOME WLI_HOME WLIS_HOST_AND_PORT WLIS_PASSWORD WLIS_USER_NAME</pre> <p>また、setupAppView コマンドは、次の環境変数を参照します。</p> <pre> DB_PINGTABLE DB_CATALOG DB_SCHEMA EVENTROUTER_URL JDBC_DATASOURCE_JNDI_NAME</pre>

showdomain

名前	showdomain -現在のドメインを表示します。
構文	showdomain
説明	このコマンドは、現在のドメインが有効であることを確認してから、現在のドメイン設定を表示します。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/showdomain \$WLI_HOME/bin/checkdomain \$WLI_HOME/setenv.sh
Windows ファイル	%WLI_HOME%\bin\showdomain.cmd %WLI_HOME%\bin\checkdomain.cmd %WLI_HOME%\setEnv.cmd
環境	showdomain コマンドは、WLI_HOME 環境変数を参照します。 WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

startAiconsole

startB2bconsole

名前	startAiconsole - Application View Console を起動します。 startB2bconsole - B2B Console を起動します。
構文	startAiconsole startB2bconsole
説明	<p>startAiconsole コマンドは、デフォルトのブラウザを起動して、Application View Console の URL (http://localhost:7001/wlai) を開きます。このコマンドは、ドメインの管理サーバがローカルホスト上で実行されている場合に、Application View Console にアクセスするためだけに使用できます。</p> <p>startB2bconsole コマンドは、デフォルトブラウザを起動して、B2B Console の URL (http://localhost:7001/b2bconsole) を開きます。このコマンドは、ドメインの管理サーバが、ローカルホスト上で実行される場合に、B2B Console にアクセスするためにだけ使用できます。</p> <p>Windows の [スタート] メニューでは、これらのコマンドのショートカットが用意されています。</p> <p>注意： UNIX システムでは、システムパス変数に実行可能なブラウザ <code>netscape</code> を指定する必要があります。</p>
オプション	なし
UNIX ファイル	このコマンドは、UNIX システムでは使用できません。
Windows ファイル	%WLI_HOME%\bin\startAiconsole.cmd %WLI_HOME%\bin\startB2bconsole.cmd
環境	このコマンドは、\$WLI_HOME 環境変数を参照します。

startpointbasecommander

名前	startpointbasecommander — PointBase 管理ツールを起動します。
構文	startpointbasecommander
説明	startpointbasecommander コマンドは、現在のデータベース設定を確認し、設定が PointBase の場合に、PointBase コマンド (PointBase 管理ツール) を起動します。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/startpointbasecommander \$WLI_HOME/bin/setdbdata
Windows ファイル	%WLI_HOME%\bin\startpointbasecommander.cmd %WLI_HOME%\bin\setdbdata.cmd
環境	このコマンドは、次の環境変数を参照します。 CLCP DB_JVMARGS DB_PORT DB_TYPE JAVA_HOME POINTBASE_HOME WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

startWeblogic

名前	startWeblogic - WebLogic Server を起動して、WebLogic Integration リソースをデプロイします。
構文	startWeblogic
説明	<p>startWeblogic コマンドの機能は次のとおりです。</p> <ul style="list-style-type: none">■ setenv、setdomain、および SetDomainTypeData コマンドを実行して、ドメイン環境を設定する。■ 古いログ ファイルを削除する。■ ドメインに対してデータベース タイプが設定されているかどうかを確認する。ドメインがデフォルトの PointBase データベースを使用するようにコンフィグレーションされている場合、PointBase コマンド (PointBase 管理ツール) を起動する。■ サーバ固有のタスクを実行する。たとえば、サーバが Application Integration 機能をサポートし、Application Integration デプロイメント ディレクトリ (DOMAIN_HOME/wlai) がない場合、このコマンドは、このディレクトリを作成する。WebLogic Integration サンプル サーバ用の startWebLogic コマンドが実行されるとき、WebLogic Integration サンプル データ ディレクトリ (\$SAMPLES_HOME/integration/config/samples/data) がない場合は、このコマンドはこのディレクトリを作成する。■ ドメイン内の WebLogic Server を起動し、config.xml ファイルの指定に従って、WebLogic Integration アプリケーションとリソースをデプロイする java コマンドを実行する。
オプション	なし
UNIX ファイル ル	<pre>DOMAIN_HOME/startWeblogic DOMAIN_HOME/SetDomainTypeData DOMAIN_HOME/dbinfo/setdbdata \$SAMPLES_HOME/integration/samples/bin/SetSampleData \$WLI_HOME/setenv.sh</pre>
Windows ファイル	<pre>DOMAIN_HOME\startWeblogic.cmd DOMAIN_HOME\SetDomainTypeData.cmd DOMAIN_HOME\dbInfo\setdbdata.cmd %SAMPLES_HOME%\integration\samples\bin\SetSampleData.cmd %WLI_HOME%\setEnv.cmd</pre>

環境 startWeblogic コマンドは、次の環境変数を参照します。

```
CLCP  
DB_JVMARGS  
DB_PORT  
DB_TYPE  
JAVA_HOME  
SVRCP  
WL_HOME  
WLI_DOMAIN_HOME  
WLI_HOME
```

stopWeblogic

名前	stopWeblogic – WebLogic Server をシャットダウンします。
構文	stopWeblogic
説明	stopWeblogic コマンドは、setenv コマンドを実行して、ドメインで実行中の WebLogic Server にシャットダウン要求を発行します。
オプション	なし
UNIX ファイル	<i>DOMAIN_HOME</i> /stopWeblogic <i>\$WLI_HOME</i> /setenv.sh
Windows ファイル	<i>DOMAIN_HOME</i> \stopWeblogic.cmd <i>%WLI_HOME%</i> \setEnv.cmd
環境	stopWeblogic コマンドは、次の環境変数を参照します。 BEA_HOME DB_JVMARGS JAVA_HOME WL_HOME WLI_HOME WLISERVERCP

studio

名前	studio - WebLogic Integration Studio を起動します。
構文	studio
説明	studio コマンドは、WebLogic Integration Studio クライアントを起動します。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/studio \$WLI_HOME/setenv.sh
Windows ファイル	%WLI_HOME%\bin\studio.cmd %WLI_HOME%\setEnv.cmd
環境	studio コマンドは、CP=\$WLI_HOME/lib/wlpi-studio.jar:\$WLICP (UNIX) または CP=%WLI_HOME%\lib\wlpi-studio.jar:%WLICP% (Windows) を設定し、次の環境変数を参照します。 CP JAVA_HOME WLI_HOME WLICP WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

switchdb

名前 switchdb -現在のドメイン用にコンフィグレーションされたデータベースを更新します。

構文 switchdb mssql|oracle|pointbase|sybase

説明 switchdb コマンドは、次の処理を行うことにより、現在のドメインのデータベース コンフィグレーションを更新します。

- checkdomain を実行して現在のドメインを確認する。
- ドメインの dbInfo/setdbtype コマンドを作成または再作成する。
- 指定したデータベース タイプの setDBVars コマンドの現在の設定に基づいて、config.xml ファイルの JDBC 設定を更新する。

switchdb コマンドは、setDBVars ファイルの更新は行いません。setDBVars ファイルのデータベース コンフィグレーションの変更が必要な場合、WebLogic Integration データベース ウィザード (wlconfig コマンド) を使用して、コンフィグレーションを更新してください。

オプション なし

UNIX ファイル

```
$WLI_HOME/bin/switchdb
$WLI_HOME/bin/checkdomain
$WLI_HOME/setenv.sh
$WLI_DOMAIN/dbInfo/setdbtype
$WLI_DOMAIN/dbInfo/$DB_TYPE/setDBVars
$WLI_DOMAIN/scripts/SwitchDB.xml
$WLI_HOME/lib/scripts/SwitchDB.xml
```

Windows ファイル

```
%WLI_HOME%\bin\switchdb.cmd
%WLI_HOME%\bin\checkdomain.cmd
%WLI_HOME%\setEnv.cmd
%WLI_DOMAIN%\dbInfo\setdbtype.cmd
%WLI_DOMAIN%\dbInfo\%DB_TYPE%\setDBVars.cmd
%WLI_DOMAIN%\scripts\SwitchDB.xml
%WLI_HOME%\lib\scripts\SwitchDB.xml
```

環境 switchdb コマンドは、DBINFO および DB_TYPE 環境変数を設定し、WLI_DOMAIN および JAVA_HOME 環境変数を参照します。

WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

worklist

名前	worklist - WebLogic Integration Worklist を起動します。
構文	worklist
説明	worklist コマンドは、WebLogic Integration Worklist クライアントを起動します。
オプション	なし
UNIX ファイル	\$WLI_HOME/bin/worklist \$WLI_HOME/setenv.sh
Windows ファイル	%WLI_HOME%\bin\worklist.cmd %WLI_HOME%\setEnv.cmd
環境	worklist コマンドは、CP=\$WLI_HOME/lib/wlpi-worklist.jar:\$WLICP (UNIX) または CP=%WLI_HOME%\lib\wlpi-worklist.jar:%WLICP% (Windows) を設定し、次の環境変数を参照します。 CP JAVA_HOME WLI_HOME WLICP WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

wliconfig

名前	wliconfig – WebLogic Integration データベース ウィザードを起動します。
構文	wliconfig
説明	このコマンドは、WebLogic Integration データベース ウィザードを起動します。詳細については、3-6 ページの「データベース ウィザードの使用法」を参照してください。
オプション	現在サポートされているオプションはありません。このウィザードは、必要な情報の入力を求めます。
UNIX ファイル	<i>DOMAIN_HOME</i> /wliconfig <i>\$WLI_HOME</i> /setenv.sh
Windows ファイル	<i>DOMAIN_HOME</i> \wliconfig.cmd <i>%WLI_HOME%</i> \setEnv.cmd
環境	wliconfig コマンドは、次の環境変数を参照します。 <i>\$JAVA_HOME</i> <i>\$WLI_HOME</i> WLI_HOME が設定されていない場合、このコマンドによって、setenv コマンドが実行されます。

索引

A

Application Integration プラグイン 2-17

Application View Console

概要 1-7

起動 2-10

ポート番号の変更 3-67

B

B2B Console

概要 1-7

起動 2-25

ポート番号の変更 3-67

B2B Integration プラグイン 2-17

B2B URI エンドポイント、変更 3-66

B2B エンジン

起動 2-27

終了 2-27

B2B サンプル アプリケーション 2-6

BPM NULL 変数 3-45

BPM サンプル 2-6

BPM パスワード、変更 3-57

BPM プラグイン 2-17

C

Channel Master サンプル 2-6

config.xml ファイル A-2

D

Data Integration

サンプル 2-6

プラグイン 2-17

E

EDI サンプル 2-6

EJB

デプロイ 3-38

デプロイメント、WLI アプリケーション A-6

F

fileRealm.properties ファイル

サンプル A-18

説明 3-48

Format Builder

概要 1-7

起動 2-22

H

Hello Partner サンプル 2-6

I

I18N サポート 3-61

J

Java クラス、デプロイ 3-38

JDBC 接続パラメータ 3-6

JDBC 接続プール A-13

JMS JDBC ストアプレフィックス名 A-9

JMS JDBC ストア ページ A-10

JMS キュー、カスタマイズ 3-34

M

MDBGenerator ユーティリティ 3-34

Messaging API サンプル 2-6

Microsoft SQL Server、必要な接続パラメータ 3-6

N

NULL 変数サポート 3-45

O

Oracle、必要な接続パラメータ 3-6

R

RDBMS レルム、コンフィグレーション 3-52

RosettaNet Security サンプル 2-6

RunSamples コマンド 1-15

S

setDBVarsExt コマンド 3-9

setDBVars コマンド 3-9

setEnv サンプル A-15

startWeblogic コマンド 1-18

startWeblogic サンプル A-16

stopWeblogic コマンド 1-21

Studio

概要 1-7

起動 2-14

プラグイン 2-17

Sybase、必要な接続パラメータ 3-6

W

WebLogic Integration

起動 1-18

サンプル アプリケーション 1-24

シャットダウン スクリプトの変更 3-68

終了 1-21

データベース ウィザード 3-6

ポート番号の変更 3-62

リポジトリ 1-11

WebLogic Server

管理サーバ 1-2

管理対象サーバ 1-2

ドメイン 1-2

リスンポート番号の変更 3-62

WebLogic Server Administration Console

JDBC コンフィグレーションの更新 3-18

RDBMS レルムの更新 3-20

起動 2-8

Web ブラウザ、コンフィグレーションの要件 2-1

wlcsystem パスワード、変更 3-58

wli.bpm.server.evaluator.supportsNull オプション 3-45

WLI アプリケーション、EJB デプロイメント A-6

wlpiCachingRealm、有効化 3-52

Worklist

概要 1-7

起動 2-19

X

XML リポジトリ 1-9

Z

Zeroweight Client サンプル 2-6

あ

アダプタ、開発 2-13

アプリケーションビュー 2-13

い

インターナショナルライゼーション 3-61

か

カスタマイズ

JMS キュー 3-34

メールセッションの設定 3-59
カスタムセキュリティレール、コンフィ
グレーション 3-54
環境、setEnv コマンド 3-32
管理サーバ 1-2
管理対象サーバ 1-2
管理ログイン 1-9

き

起動

Application View Console 2-10
B2B Console 2-25
B2B エンジン 2-27
Format Builder 2-22
Studio 2-14
WebLogic Server Administration
Console 2-8
Worklist 2-19
起動完了メッセージ 1-21
起動クラス A-12
キュー、カスタマイズ 3-34

く

クラス、起動とシャットダウン A-12
クラスタ化 1-2
グループ
システム 3-51
パーミッション 3-50
ロール 3-51

こ

コマンド
RunSamples 1-15
setDBVars 3-9
setDBVarsExt 3-9
setEnv 3-32
setEnv サンプル A-15
startWeblogic 1-18
startWeblogic サンプル A-16
コンフィグレーション

JMS キュー 3-34

RDBMS レール 3-52
カスタムセキュリティレール 3-54
サンプルドメイン 1-15
セキュリティレール 3-48
代替セキュリティレール 3-51
データベース 3-6
要件 1-11
リポジトリ 1-14

コンフィグレーション済みドメイン 1-3

さ

サンプルアプリケーション
概要 2-6
実行 2-4
サンプル起動ページ 2-4
サンプルサーバ 1-15
サンプルドメイン 1-3, 1-15

し

システムグループ 3-51
システムデータのロード 3-5
システムパスワード、変更 3-57
システムログイン 1-9
シャットダウン
B2B エンジン 2-28
WebLogic Integration 1-21
スクリプトの変更 3-68
シャットダウン クラス A-12
終了
B2B エンジン 2-27
WebLogic Integration 1-21

せ

セキュリティレール
ガイドライン 3-51
コンフィグレーション 3-48
代替 3-51
接続パラメータ 3-6

た

- 代替セキュリティレーム、コンフィグレーション 3-51
- 代替文字セット 3-61

て

- データベース
 - アクセス パラメータ 3-21
 - 移行 3-6
 - 格納 1-14
 - 切り替え 3-6
 - 作成 3-6
 - 初期化 1-14, 3-5
 - 接続パラメータ 3-6
 - 変更 3-2
- データベース ウィザード 3-6
- データベース コンフィグレーション ウィザード
 - グラフィック モード 3-10
 - コンソール モード 3-16
 - モード 3-9
- データベースの移行 3-6
- データベースの切り替え 3-6
- データベースの作成 3-6
- データベース リポジトリの初期化 3-5
- テーブル、WebLogic Integration リポジトリ 1-9
- デプロイ
 - EJB 3-38
 - Java クラス 3-38
 - 非永続モードの B2B 3-68
- 電子メール、設定のカスタマイズ 3-59

と

- ドメイン
 - WebLogic Server 1-2
 - コンフィグレーション済み 1-3
 - 新規作成 3-23
 - ルート ディレクトリ 3-61

は

- パスワード
 - BPM パスワードの変更 3-57
 - wlcsystem パスワードの変更 3-58
 - システム パスワードの変更 3-57
 - デフォルト 1-9
 - 変更 3-56
- パスワードの更新 3-56
- パーミッション グループ 3-50

ひ

- ビジネス プロセス管理、セキュリティ 3-48

ふ

- ブラウザ、コンフィグレーションの要件 2-1
- プラグイン
 - Application Integration プラグイン 2-17
 - B2B Integration プラグイン 2-17
 - Data Integration プラグイン 2-17
- プロキシ、回避 2-1

へ

- 変数、NULL のサポート 3-45

ほ

- ポート番号、変更
 - Application View Console 3-67
 - B2B Console 3-67
 - B2B URI エンドポイント 3-66
 - WebLogic Integration 3-62
 - WebLogic Server リスン 3-62

め

- メッセージ、起動完了 1-21
- メッセージ フォーマット 2-22

メールセッション
設定のカスタマイズ 3-59
プロパティ 3-60

ゆ

ユーザ 3-56

よ

要素

関係 1-9
リポジトリ内 1-9

り

リポジトリ

概要 1-9
システムデータのロード 3-5
初期化 1-14
新規指定 3-2

る

ルートディレクトリ、ドメイン 3-61

ろ

ログイン、デフォルト 1-9
ロールグループ 3-51

