目次

このマニュアルの内容

20.ドメイン ログ フィルタ

21.EJB コンポーネント

22.実行時 EJB コンポーネント

23.EJB ホーム

24.EJB デプロイメント記述子

25.実行時 EJB ホーム

26.実行時エンティティ EJB

27.実行キュー

28.実行時実行キュー

29.実行スレッド

30.FileT3

31.グループ

32.JDBC 接続プール

33.実行時 JDBC 接続プール

34.JDBC データ ソース

35.JDBC マルチプール

36.JDBC トランザクション データ ソース

37.JMS 接続コンシューマ

38.JMS 接続ファクトリ

46.JMS JDBC ストア

47.実行時 JMS プロデューサ

48.JMS キュー

49.実行時 JMS

50.JMS サーバ

51.実行時 JMS サーバ

52.JMS セッション プール

53.実行時 JMS セッション プール

54.JMS ストア

55.JMS テンプレート

56.JMS トピック

57.メッセージング ブリッジ

58.JMS ブリッジ送り先

59.一般ブリッジ送り先

60.JNDIBinding

61.JNDI コンテキスト

62.Jolt

63.Jolt 接続プール

64.実行時 Jolt 接続プール

65.JTA

66.JTA トランザクション

67.LDAP レルム（廃止）

68.マシン

69.メール セッション

70.MBean

71.実行時メッセージ駆動型 EJB

72.NT レルム

73.プリファレンス

74.RDBMS レルム

75.RDBMS デプロイメント記述子

78.サーバ

79.サーブレット

80.サーブレット

81.サーブレット セッション

82.停止クラス

83.実行時 SNMP トラップ

84.実行時 SNMP

85.SNMP 属性変更

105.実行時トランザクション名

106.実行時トランザクション リソース

107.Unix マシン

108.Unix レルム

109.ユーザ

110.仮想ホスト

111.Web アプリケーション

112.実行時 Web アプリケーション コンポーネント

113.Web アプリケーション デプロイメント記述子エディタ ヘルプ

114.実行時 WLEC 接続プール

115.WLEC 接続プール

116.WLEC 接続

117.XML エンティティ定義

118.XML レジストリ

119.XML レジストリ エントリ

BEA Systems WebLogic Server Administration Console ヘルプ

著作権

Copyright © 2001 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. 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 Systems, Inc. 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 Systems, Inc. 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、市場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 Systems, Inc. は、正当性、正確さ、信頼性などの点から、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、ObjectBroker、TOP END、WebLogic、および Tuxedo は BEA Systems, Inc. の登録商標です。BEA Builder、BEA Connect、BEA Manager、BEA MessageQ、BEA Jolt、M3、eSolutions、eLink、WebLogic Enterprise、WebLogic Commerce Server、WebLogic Personalization Server、および WebLogic Server は、BEA Systems, Inc の商標です。

その他の製品名はすべて、関係各社の商標である場合があります。

BEA WebLogic Server Administration Console オンライン ヘルプ

パート番号	マニュアルの日付	ソフトウェアのバージョン
なし	2001 年 < 月 > < 日 >	WebLogic Server バージョン 6.X

著作権

Copyright © 2001 BEA Systems, Inc. All Rights Reserved.

限定的権利条項

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

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

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

商標または登録商標

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

その他の商標はすべて、関係各社がその権利を有します。

BEA WebLogic Server Administration Console オンライン ヘルプ

マニュアルの日付	ソフトウェアのバージョン
2001 年 12 月 19 日	BEA WebLogic Server 6.1

このマニュアルの内容

このマニュアルでは、BEA WebLogic Serverô Administration Console の機能と BEA WebLogic Server ドメインを管理する手順について説明します。

このマニュアルの内容は以下のとおりです。

- Administration Console のノードごとに章が分かれています。
- 各章には、そのノードで設定できる属性の情報、またはそのノードで利用できる情報、エンティティの作成およびデプロイメントの手順、プログラミングと管理についての追加情報へのリンクなどが含まれています。

対象読者

このマニュアルは、1 つまたは複数の WebLogic Server ドメインを管理するシステム管理者またはオペレータを対象としています。Web 技術に読者が精通していることを前提として書かれています。

e-docs Web サイト

BEA 製品のドキュメントは、BEA の Web サイトで入手できます。BEA 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 WebLogic Server 製品ドキュメント ページ（http://edocs.beasys.co.jp/e-docs/）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 トピックずつ印刷できます。

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

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

関連情報

BEA の Web サイトでは WebLogic Server の全ドキュメントを提供しています。

サポート情報

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

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

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

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

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
〔Ctrl〕+〔Tab〕	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、Java クラス、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。
	例 :
	`import java.util.Enumeration;`
	`chmod u+w *`
	`config/examples/applications`
	`.java`
	`config.xml`
	`float`

斜体の等幅テキスト	コード内の変数を示す。 例 : `String CustomerName;`

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。 例 : LPT1 BEA_HOME OR
{ }	構文の中で複数の選択肢を示す。
[ ]	構文の中で任意指定の項目を示す。例 : `java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]`
|	構文の中で任意指定の項目を示す。例 : `java weblogic.deploy [list|deploy|undeploy|update] password {application} {source}`
. . .	コマンドラインで以下のいずれかを示す。 ♦ 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
. . .	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。

1 ACL

以下に、Administration Console を使用して、アクセス制御リスト (ACL) の作成や管理に必要な属性を設定する手順を説明します。ACL の詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

新しい ACL の作成

1. 左ペインの [ACL] ノードをクリックします。右ペインに [ACL] テーブルが表示され、ドメイン内のすべての ACL が示されます。
2. [新しい ACL 名] フィールドに値を入力します。
3. [作成] ボタンをクリックします。右ペインにダイアログが表示され、新しい ACL の作成に関連する情報が示されます。
4. [パーミッション]、[ユーザ]、[グループ] フィールドに値を入力して、どの許可を割り当てるのか、およびそれらの許可を受けるのはどのユーザとグループなのかを指定します。各フィールドでは、複数の値をスペースで区切ります。
5. [パーミッションを付与] ボタンをクリックして、ACL を作成します。

新しい ACL の作成の詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

ACL の変更

1. 左ペインの [ACL] ノードをクリックします。右ペインに [ACL] テーブルが表示され、ドメイン内のすべての ACL が示されます。

2. 変更する ACL の名前をクリックします。
3. [パーミッション]、[ユーザ]、および [グループ] フィールドの値を変更して、どの許可を割り当てるのか、およびそれらの許可を受けるのはどのユーザとグループなのかを指定します。各フィールドでは、複数の値をスペースで区切ります。
4. [パーミッションを付与] ボタンをクリックして、ACL を変更します。

ACL の変更の詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

[パーミッション]

属性	説明	値の範囲	デフォルト値
[パーミッション]	特定の種類のアクセスを個人またはグループに許可する。	[write]、[read]、[modify]	NULL
[ユーザ]	特定の種類のアクセスを個人に許可する。	[write]、[read]、[modify]	NULL
[グループ]	特定の種類のアクセスをグループに許可する。	[write]、[read]、[modify]	NULL

ACL の詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

2　ACL のパーミッション

以下の表に、アクセス制御リスト（ACL）でのパーミッションの設定に使用する属性を示します。ACL の詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

[パーミッション]

属性	説明	値の範囲	デフォルト値
[パーミッション]	特定の種類のアクセスを個人またはグループに許可する。	[write]、[read]、[modify]	NULL
[ユーザ]	特定の種類のアクセスを個人に許可する。	[write]、[read]、[modify]	NULL
[グループ]	特定の種類のアクセスをグループに許可する。	[write]、[read]、[modify]	NULL

ACL の詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

3　アプリケーション

以下に、Administration Console を使用して、新しいアプリケーションのインストールやデプロイに必要な属性を設定する手順を説明します。アプリケーションの詳細については、『管理者ガイド』の「アプリケーションのデプロイ」を参照してください。

新しいアプリケーションのインストール

1. 左ペインの [アプリケーション] ノードをクリックします。右ペインに [アプリケーション] テーブルが表示され、ドメインにインストールされているすべてのアプリケーションが示されます。
2. [新しい Application のコンフィグレーション] リンクをクリックして、[新しい Application の作成] ページを開きます。
3. コンフィグレーション情報を次のように入力します。
   - [名前] フィールドにアプリケーションの名前を入力します。
   - [パス] フィールドに .ear、.jar、または .war アプリケーションのパスを入力します。
   - 新しいアプリケーションをデプロイする場合は、[デプロイ] ボックスを選択します。
4. [作成] ボタンをクリックして、アプリケーションをインストールします。新しいアプリケーションが左ペインの [アプリケーション] ノードの下に追加されます。

アプリケーションの詳細については、『管理者ガイド』の「アプリケーションのデプロイ」を参照してください。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	アプリケーションの名前を返す。	文字列	なし
[パス]	アプリケーションの完全なパスを返す。	文字列	なし
[デプロイ]	アプリケーションのデプロイメント ステータスをレポートする。		なし

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

アプリケーションの詳細については、『管理者ガイド』の「アプリケーションのデプロイ」を参照してください。

4 基本レルム

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前

5　キャッシング レルム

以下に、Administration Console を使用して、キャッシング レルムのコンフィグレーションや管理に必要な属性を設定する手順を説明します。キャッシング レルムについては、『管理者ガイド』の「セキュリティの管理」を参照してください。

新しいキャッシング レルムのコンフィグレーション

1. 左ペインの [キャッシング レルム] ノードをクリックします。右ペインに [キャッシング レルム] テーブルが表示され、ドメインで定義されているすべてのキャッシング レルムが示されます。
2. [新しい Caching Realm のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するキャッシング レルムの設定に関連するタブが示されます。
3. [名前] および [基本レルム] 属性フィールドに値を入力します。[キャッシュで大文字 / 小文字を区別] チェックボックスをクリックすると、大文字 / 小文字を区別できます。
4. [作成] をクリックして、[名前] フィールドに指定した名前でキャッシング レルムのインスタンスを作成します。新しいインスタンスが左ペインの [キャッシング レルム] ノードの下に追加されます。
5. [ACL]、[認証]、[グループ]、[ユーザ]、および [パーミッション] タブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

キャッシング レルムのクローンの作成

1. 左ペインの [キャッシング レルム] ノードをクリックします。右ペインに [キャッシング レルム] テーブルが表示され、ドメインで定義されているすべてのキャッシング レルムが示されます。
2. クローンを作成するキャッシング レルムの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、キャッシング レルムのクローンの作成に関連するタブが示されます。
3. [名前] および [基本レルム] 属性フィールドに値を入力します。[キャッシュで大文字 / 小文字を区別] チェックボックスをクリックすると、大文字 / 小文字を区別できます。
4. 右下隅の [作成] ボタンをクリックして、[名前] フィールドに指定した名前でキャッシング レルムのインスタンスを作成します。新しいインスタンスが左ペインの [キャッシング レルム] ノードの下に追加されます。
5. [ACL]、[認証]、[グループ]、[ユーザ]、および [パーミッション] タブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

キャッシング レルムの削除

1. 左ペインの [キャッシング レルム] ノードをクリックします。右ペインに [キャッシング レルム] テーブルが表示され、ドメインで定義されているすべてのキャッシング レルムが示されます。
2. 削除するキャッシング レルムの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、キャッシング レルムを削除します。[キャッシング レルム] ノードの下のキャッシング レルム アイコンが削除されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	アクティブなセキュリティ レルムを表示する。このフィールドは変更不可。	文字列	MyCachingRealm
[基本レルム]	キャッシング レルムと共に使用される代替セキュリティ レルムまたはカスタム セキュリティ レルムの名前。	リスト	なし
[キャッシュで大文字 / 小文字を区別]	指定したセキュリティ レルムで大文字 / 小文字を区別するかどうかを定義する。デフォルトでは、このフィールドは有効になっており、大文字 / 小文字は区別される。大文字 / 小文字を区別しないレルム（Windows NT および LDAP のセキュリティ レルムなど）を使用するには、このフィールドを無効にする必要がある。	ブール 選択されている = True 選択されていない = False	True

[ACL]

属性	説明	値の範囲	デフォルト値
[ACL キャッシュを有効化]	ACL キャッシュを有効化するオプション。	ブール 選択されている = True 選択されていない = False	True
[ACL キャッシュ サイズ]	キャッシュする ACL ルックアップの最大数。	最高のルックアップ性能を実現する適切な数	211
[成功時の ACL キャッシュ生存時間]	成功したルックアップの結果を保持する秒数。	整数	60
[失敗時の ACL キャッシュ生存時間]	失敗したルックアップの結果を保持する秒数。	整数	10

[認証]

属性	説明	値の範囲	デフォルト値
[認証キャッシュを有効化]	認証キャッシュが有効な場合に True に設定される。	ブール 選択されている = True 選択されていない = False	True
[認証キャッシュ サイズ]	キャッシュする認証要求の最大数を設定する。	最高のルックアップ性能を実現する適切な数	211
[成功時の認証キャッシュ生存時間]	成功したルックアップの結果を保持する秒数を設定する。	整数	60
[失敗時の認証キャッシュ生存時間]	失敗したルックアップの結果を保持する秒数を設定する。	整数	10

[グループ]

属性	説明	値の範囲	デフォルト値
[グループ キャッシュを有効化]	グループ キャッシュが有効な場合に True に設定される。	ブール 選択されている = True 選択されていない = False	True
[グループ キャッシュ サイズ]	キャッシュするグループ ルックアップの最大数を設定する。	最高のルックアップ性能を実現する適切な数	211
[成功時のグループ キャッシュ生存時間]	成功したルックアップの結果を保持する秒数を設定する。	整数	60
[失敗時のグループ キャッシュ生存時間]	失敗したルックアップの結果を保持する秒数を設定する。	整数	10
[グループ メンバシップ キャッシュ生存時間]	更新前にグループのメンバを保存する秒数を設定する。	整数	300

[ユーザ]

属性	説明	値の範囲	デフォルト値
[ユーザ キャッシュを有効化]	ユーザ キャッシュが有効な場合に True に設定される。	ブール 選択されている = True 選択されていない = False	True
[ユーザ キャッシュ サイズ]	キャッシュするユーザ ルックアップの最大数を設定する。	最高のルックアップ性能を実現する適切な数	211
[成功時のユーザ キャッシュ生存時間]	成功したルックアップの結果を保持する秒数を設定する。	整数	60
[失敗時のユーザ キャッシュ生存時間]	失敗したルックアップの結果を保持する秒数を設定する。	整数	10

[パーミッション]

属性	説明	値の範囲	デフォルト値
[パーミッション キャッシュを有効化]	許可キャッシュが有効な場合に True に設定される。	ブール 選択されている = True 選択されていない = False	True
[パーミッション キャッシュ サイズ]	キャッシュする許可ルックアップの最大数を設定する。	最高のルックアップ性能を実現する適切な数	211
[成功時のパーミッション キャッシュ生存時間]	成功したルックアップの結果を保持する秒数を設定する。	整数	60
[失敗時のパーミッション キャッシュ生存時間]	失敗したルックアップの結果を保持する秒数を設定する。	整数	10

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

キャッシング レルムについては、『管理者ガイド』の「セキュリティの管理」を参照してください。

6 スタートアップ クラスとシャットダウン クラスのデプロイメント

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前
- 型

7　クラスタ

以下に、Administration Console を使用して、クラスタのコンフィグレーションや管理に必要な属性を設定する手順を説明します。クラスタについての詳細は、『管理者ガイド』の「WebLogic Server とクラスタのコンフィグレーション」および『WebLogic Server クラスタ ユーザーズ ガイド』を参照してください。

クラスタのコンフィグレーション

1. 左ペインの [クラスタ] ノードをクリックします。右ペインに [クラスタ] テーブルが表示され、ドメインで定義されているすべてのクラスタが示されます。
2. [新しい Cluster のコンフィグレーション] リンクをクリックします。右ペインにダイアログが表示され、新しく作成するクラスタの設定に関連するタブが示されます。
3. [名前]、[デフォルトのロードバランス アルゴリズム]、および [サービス期間しきい値] 属性フィールドに値を入力します。
4. [クラスタ アドレス] に値を入力します。プロダクション システムでは、このアドレスは、クラスタに参加している管理サーバの個々の IP アドレスにマップされる DNS ホスト名にする必要があります。開発（非プロダクション）システムでは、このアドレスには、DNS ホスト名、あるいはシングル アドレス ホスト名または IP アドレスのカンマ区切りリストを指定できます。クラスタのアドレス形式の詳細については、『WebLogic Server クラスタ ユーザーズ ガイド』の「クラスタの DNS 名」を参照してください。
5. 右下隅の [作成] ボタンをクリックして、[名前] フィールドに指定した名前でクラスタを作成します。新しいクラスタが左ペインの [クラスタ] ノードの下に追加されます。

6. [マルチキャスト] タブをクリックして、[送信遅延] 属性フィールドまたは [生存時間] を変更するか、割り当てられているデフォルト値をそのまま使用します。マルチキャストアドレスの有効な範囲は 224.0.0.1 から 239.255.255.255 です。
7. 各タブの右下隅の [適用] ボタンをクリックして、これまでの変更を保存します。

クラスタのクローンの作成

1. 左ペインの [クラスタ] ノードをクリックします。右ペインに [クラスタ] テーブルが表示され、ドメインで定義されているすべてのクラスタが示されます。
2. クローンを作成するクラスタの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、クラスタのクローンの作成に関連するタブが示されます。
3. [名前]、[クラスタ アドレス]、[デフォルトのロードバランス アルゴリズム]、および [サービス期間しきい値] 属性フィールドに値を入力します。
4. 右下隅の [クローン] ボタンをクリックして、[名前] フィールドに指定した名前でクラスタのインスタンスを作成します。新しいインスタンスが左ペインの [クラスタ] ノードの下に追加されます。
5. [マルチキャスト] タブをクリックして、属性フィールドを変更するか、割り当てられているデフォルト値をそのまま使用します。
6. 各タブの右下隅の [適用] ボタンをクリックして、これまでの変更を保存します。

クラスタの削除

1. 左ペインの [クラスタ] ノードをクリックします。右ペインに [クラスタ] テーブルが表示され、ドメインで定義されているすべてのクラスタが示されます。

2. 削除するクラスタの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、クラスタを削除します。[クラスタ] ノードの下のクラスタ アイコンが削除されます。

クラスタへのサーバの割り当て

1. 左ペインの [クラスタ] の下のインスタンス ノードをクリックして、サーバを割り当てるクラスタを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サーバ] タブをクリックします。
3. [選択可] カラムで、クラスタに割り当てる 1 つまたは複数のサーバを選択します。
4. 移動コントロールをクリックして、選択したサーバを [選択済み] カラムに移動します。
5. 右下隅の [適用] をクリックして、割り当てを保存します。

クラスタを構成するサーバのモニタ

1. 左ペインの [クラスタ] の下のインスタンス ノードをクリックして、サーバをモニタするクラスタを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [このクラスタを構成するサーバをモニタ] テキスト リンクをクリックします。右ペインにサーバ テーブルが表示され、このクラスタに割り当てられているすべてのサーバが示されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	クラスタの名前を返す。	カンマとスペースの含まれていない、256 文字までの英数字	NULL
[クラスタアドレス]	このクラスタに接続するためにクライアントで使用されるアドレスを返す。複数の IP アドレスに対応する DNS ホスト名、あるいは単一アドレスのホスト名または IP アドレスで構成されるカンマ区切りのリストのいずれか。	複数の IP アドレスに対応する DNS ホスト名、あるいは単一アドレスのホスト名または IP アドレスで構成されるカンマ区切りのリストのいずれか。	NULL
[デフォルトのロードバランスアルゴリズム]	デフォルト クラスタ ロード アルゴリズムとは、負荷を分散するアルゴリズムのこと。最後のサーバにトランザクションが割り当てられている場合に、負荷が最初のサーバに戻される。	[round-robin]、[Weight-based Round Robin]、[Random]、および [Parameter-based Routing]	round-robin
[サービス期間しきい値]	2 つの競合するサービスがある場合に、それらの一方が他方より古いと判断される基準となる存続期間の差を秒数で示す。	整数（秒数）	180

[マルチキャスト]

属性	説明	値の範囲	デフォルト値
[マルチキャスト アドレス]	クラスタ メンバーが互いに通信するために使用するマルチキャスト アドレスを設定する。必ず、正しい形式の IP アドレスを使用する。	224.0.0.1 ～ 239.255.255.255	NULL
[マルチキャスト送信遅延]	OS レベルでのバッファのオーバーフローを防止するために、マルチキャストによるメッセージの断片の送信が遅延されるミリ秒数を返す。	整数	12
[マルチキャスト生存期間]	マルチキャストの TTL の値を返す。	整数	1

[サーバ]

属性	説明	値の範囲	デフォルト値
[このクラスタへのサーバを選択]	このクラスタに参加するサーバをユーザが選択できる。	リスト	NULL

[モニタ]

属性	説明	値の範囲	デフォルト値
[このクラスタに登録されているサーバ数]	このクラスタで動作するように設定されているサーバの数を返す。	整数	0
[現在このクラスタを構成するサーバ数]	このクラスタで動作しているサーバの数を返す。	整数	0

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

8 実行時クラスタ

[モニタ] タブを使用して、次の実行時クラスタの統計をモニタすることができます。

:

統計	説明
[クラスタ]	管理ドメイン内のクラスタの名前。
[名前]	クラスタの個々のサーバ インスタンスの名前。
[マシン]	個々のサーバ インスタンスが稼動しているマシン名 (オプション) 。
[サーバ数]	クラスタに存在するサーバ数。
[再送要求数]	JNDI 更新がクラスタのメンバに再送された回数。個々のサーバ インスタンスは、クラスタのマルチキャスト アドレスを通じて更新のブロードキャストを受け取れなかったことを検知した場合、JNDI ツリーに更新を要求することがある。
[送信したフラグメント数]	個々のサーバがクラスタにブロードキャストしたマルチキャスト フラグメントの総数。
[受信したフラグメント数]	個々のサーバがクラスタのマルチキャスト アドレスを通じて受信したマルチキャスト フラグメントの総数。
[失われたマルチキャスト メッセージ数]	サーバが受信に失敗したマルチキャスト メッセージの総数。この値は、サーバ自身が 入ってくるメッセージのシーケンス ナンバーを比較して決定する。
[Foreign Fragments Dropped Count]	別のクラスタで生成され、このクラスタのメンバ宛てでなかったためにドロップされたマルチキャスト フラグメントの数。
[サーバ]	このクラスタのサーバ インスタンスのリスト。

統計	説明
[セカンダリ分散]	ステートフル オブジェクトのインメモリ レプリケーションを使用するクラスタでは、この列にリモート サーバの名前のリストが表示される。このローカル サーバは、ここに挙げられたリモート サーバのために、レプリケーションされたオブジェクトのセカンダリ コピーを保持する。サーバ名の後の数値は、ローカル サーバがそのリモート サーバのために保持しているセカンダリ オブジェクト レプリカの数を表す。
[プライマリ数]	ステートフル オブジェクトのインメモリ レプリケーションを使用するクラスタでは、この値はローカル サーバが保持しているプライマリ オブジェクトの数を表す。

9 コンポーネント

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- マシン
- ソース情報
- サーブレット
- セッション
- 最大セッション数
- 総セッション数

10 BEA WebLogic J2EE コネクタ アーキテクチャの属性記述

ra.xml の属性

デプロイメント記述子エディタで操作する ra.xml の属性

属性	説明	値の範囲	デフォルト値
[Connector Description]	省略可能な要素。セキュリティ コントラクトおよび認証メカニズムを実装するための、リソース アダプタ固有の要件を指定する。	文字列	NULL
[Connector Display Name]	ツールでの表示に使う短い名前を指定する。表示名は一意である必要はない。	文字列	NULL
[Small Icon]	小さいサイズ（16 × 16 ピクセル）のアイコン画像ファイルの名前を指定する。ファイル名は、リソース アダプタの rar ファイル内部での相対パスで指定する。JPEG または GIF 形式の画像を使用できる。	JPEG GIF	

属性	説明	値の範囲	デフォルト値
[Large Icon]	大きなサイズ（32 × 32 ピクセル）のアイコン画像ファイルの名前を指定する。ファイル名は、リソース アダプタの rar ファイル内部での相対パスで指定する。JPEG または GIF 形式の画像を使用できる。	JPEG GIF	
[Connector Vendor Name]	リソース アダプタの開発元ベンダの名前を指定する。	文字列	NULL
[Connector Spec Version]	このリソース アダプタが対応しているコネクタ アーキテクチャ仕様のバージョンを指定する。デプロイヤはこの情報を参考にしてリソース アダプタをコンフィグレーションし、対応するコネクタ アーキテクチャ仕様のデプロイメント要件および実行時要件を満たすことができる。	文字列	NULL
[Connector Eis Type]	EIS のタイプについての情報を指定する。たとえば、EIS のタイプとして、バージョン情報を除いた EIS の製品名を指定することがある。	文字列	NULL
[Connector Version]	文字列で表したリソース アダプタのバージョンで、リソース アダプタの開発元によって指定される。	文字列	NULL
[License Description]	テキストによるライセンス要素の説明。この要素は省略可能である。	文字列	NULL
[License Required]	リソース アダプタをデプロイし、使用するためにライセンスが必要かどうかを指定する。	True False	NULL
[Resourceadapter Managedconnectionfactory Class]	javax.resource.spi.Managed-ConnectionFactory インターフェースを実装する Java クラスの完全修飾名を指定する。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[Resourceadapter Connectionfactory Interface]	リソース アダプタによってサポートされる ConnectionFactory インターフェースの完全修飾名を指定する。	文字列	NULL
[Resourceadapter Connectionfactory Impl Class]	リソース アダプタに固有の ConnectionFactory インターフェースを実装する ConnectionFactory クラスの完全修飾名を指定する。	文字列	NULL
[Resourceadapter Connection Interface]	リソース アダプタによってサポートされる Connection インターフェースの完全修飾名を指定する。	文字列	NULL
[Resourceadapter Connection Impl Class]	リソース アダプタに固有の Connection インターフェースを実装する Connection クラスの完全修飾名を指定する。	文字列	NULL
[Transaction Support]	リソース アダプタによって提供されるトランザクション サポートのレベルを指定する。	NoTransaction LocalTransaction XATransaction	NULL
[Reauthentication Support]	リソース アダプタの実装が、既存の Managed-Connection インスタンスの再認証をサポートするかどうかを指定する。	True False	NULL

ra.xml のコンフィグレーション プロパティ

属性	説明	値の範囲	デフォルト値
[Description]	プロパティの範囲または定義済みの値を記述するために使用する。	文字列	NULL
[Config Property Name]	コンフィグレーション プロパティの名前を指定する。コネクタ アーキテクチャでは、すべて java.lang.String 型である定義済みプロパティの集合が用意されている。リソース アダプタ プロバイダはこのプロパティ集合を拡張して、リソース アダプタとその基盤である EIS に固有のプロパティを取り入れることができる。	ServerName PortNumber UserName Password ConnectionURL	NULL
[Config Property Type]	ManagedConnection-Factory インスタンスに必要とされるコンフィグレーション プロパティの完全修飾 Java 型を指定する。	java.lang.Boolean java.lang.String java.lang.Integer java.lang.Double java.lang.Byte java.lang.Short java.lang.Long java.lang.Float java.lang.Character	NULL
[Config Property Value]	コンフィグレーション エントリの値を指定する。	不定	NULL

ra.xml の認証メカニズム

ra.xml の認証メカニズム

属性	説明	値の範囲	デフォルト値
[Description]	認証メカニズムのタイプをテキストで記述する。	文字列	NULL
[Authentication Mechanism Type]	リソース アダプタによってサポートされる認証メカニズムを指定する。このサポートは、基盤の EIS インスタンスではなくリソース アダプタを対象としたものである。必要に応じて、セキュリティ コントラクトおよび認証メカニズムをサポートするための、リソース アダプタ固有の要件についての説明を指定する。メカニズムのタイプとして BasicPassword を指定する場合、javax.resource.spi.security.PasswordCredential インターフェースのサポートが必要である。Kerbv5 を指定する場合、javax.resource.spi.security.GenericCredential インターフェースのサポートが必要である。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[Credential Interface]	資格を表現するためにリソース アダプタの実装がサポートするインターフェースを指定する。この要素は、セキュリティ コントラクトの一部として使用する Credential インターフェースを検索する目的で、アプリケーション サーバによって使用されることを想定している。	文字列	NULL

ra.xml のセキュリティ パーミッション

属性	説明	値の範囲	デフォルト値
[Description]	テキストによるセキュリティ パーミッションの説明を指定する。	String	NULL
[Security Permission Spec]	セキュリティ ポリシー ファイルの構文に従ってセキュリティ パーミッションを指定する。	文字列	NULL

weblogic-ra.xml の属性

デプロイメント記述子エディタで操作する weblogic-ra.xml の属性

属性	説明	値の範囲	デフォルト値
[Description]	親の要素についてのテキストによる説明を指定する。説明目的の要素には、デプロイされる接続ファクトリについてデプロイヤが注記しておきたい情報をすべて記述しておくことが望ましい。この要素は省略可能である。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[Connection Factory Name]	リソース アダプタの特定のデプロイメントと、それに対応する接続ファクトリに関連付けられる論理名を定義する。connection-factory-name 要素の値は、ra-link-ref 要素を介して、デプロイ対象のその他のリソース アダプタで使用することができる。これにより、デプロイ対象の複数の接続ファクトリで、共通のデプロイ対象リソース アダプタの利用およびコンフィグレーション仕様の共有が可能になる。この要素は必ず指定する。	文字列	NULL
[JNDI 名]	接続ファクトリ オブジェクトを WebLogic JNDI 名前空間にバインドするために使われる名前を定義する。クライアントの EJB およびサーブレットは、WebLogic に固有のデプロイメント記述子に定義済みのリファレンス記述子要素で、これと同じ JNDI 名を使用する。この要素は必ず指定する。	文字列	NULL
[初期容量]	WebLogic Server がデプロイメントの間に取得を試みる管理対象接続の初期数を示す。この要素は省略可能である。この値を指定しない場合、WebLogic は定義済みのデフォルト値を使用する。	整数	1

属性	説明	値の範囲	デフォルト値
[最大容量]	この要素は、WebLogic Server で許容する管理対象接続の最大数を示す。管理対象接続がこの数に達しているときに接続割り当て要求が新しく出されると、呼び出し側に ResourceAllocationException が返される。この要素は省略可能である。この値を指定しない場合、WebLogic は定義済みのデフォルト値を使用する。	整数	10
[増加容量]	管理対象の接続プールのサイズ変更の間に、WebLogic Server が追加で取得を試みる管理対象接続の数を指定する。この要素は省略可能である。この値を指定しない場合、WebLogic は定義済みのデフォルト値を使用する。	整数	1
[縮小可]	システム リソースを制御する手段として、未使用の管理対象接続を接続プールで再利用するかどうかを指定する。この要素は省略可能である。この値を指定しない場合、WebLogic は定義済みのデフォルト値を使用する。	True False	True
[縮小間隔]	接続プール管理において、未使用の管理対象接続の再利用を試みるまでの待ち時間を指定する。この要素は省略可能である。この値を指定しない場合、WebLogic は定義済みのデフォルト値を使用する。	整数	1

属性	説明	値の範囲	デフォルト値
[Ra Link Ref]	複数のデプロイ対象接続ファクトリを、単一のデプロイ対象リソース アダプタに論理的に関連付けるための要素。独立してデプロイされている接続ファクトリを識別する値をこの要素に指定すると、新しくデプロイされる接続ファクトリは、デプロイ済みのリソース アダプタを、この要素に指定された接続ファクトリとの間で共有するようになる。また、指定された接続ファクトリのデプロイメントで定義されている値はすべて、特に指定のない限り、この新しくデプロイされる接続ファクトリに継承される。この要素は省略可能である。	文字列	NULL
[Native Lib Dir]	このリソース アダプタのデプロイメントに存在するすべてのネイティブ ライブラリに対して使われるディレクトリ位置を指定する。デプロイメント処理の一環として、見つかったすべてのネイティブ ライブラリが、この要素に指定された位置にコピーされる。WebLogic Server の実行時にこれらのライブラリが見つかるように、プラットフォームごとに必要な作業を行うのは管理者の責任である。この要素は、ネイティブ ライブラリが存在する場合には必ず指定する。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[ログを有効化]	ManagedConnectionFactory および ManagedConnection に関して、ログの書き込みを有効または無効にする。この要素を true に設定すると、ManagedConnectionFactory または ManagedConnection のどちらかから生成される出力が、log-filename 要素に指定されたファイルに送られる。この要素は省略可能である。この値を指定しない場合、WebLogic は定義済みのデフォルト値を使用する。	True False	False
[Log Filename]	ManagedConnectionFactory または ManagedConnection から生成された出力が送られるログ ファイルの名前を指定する。ファイル名は完全なアドレスで指定する必要がある。この要素は省略可能である。	文字列	NULL

weblogic-ra.xml のコンフィグレーション プロパティ

属性	説明	値の範囲	デフォルト値
[Description]	コンフィグレーション プロパティについてのテキストによる説明を指定する。	文字列	NULL
[Config Property Name]	この要素は、対応する config-property-name 要素とともに、ra.xml の config-entry 要素に対応する名前を識別する。	文字列	NULL
[Config Property Type]	この要素は、対応する config-property-name 要素とともに、ra.xml の config-entry 要素に対応する型を識別する。	文字列	NULL
[Config Property Value]	この要素は、対応する config-property-name 要素とともに、ra.xml の config-entry 要素に対応する値を識別する。		NULL

weblogic-ra.xml のセキュリティ プリンシパル マップ エントリ

属性	説明	値の範囲	デフォルト値
[Initiating Principal]	security-principal-map 要素は、既知の WebLogic 実行時開始プリンシパル（*initiating-principal*）に基づいて、リソース アダプタおよび EIS での認可処理のために適切なリソース プリンシパル（resource-principal）の値を定義するためのメカニズムを提供する。このマップでは、開始プリンシパルと、それに対応するリソース プリンシパルのユーザ名およびパスワードの定義済み集合を指定することができる。このユーザ名とパスワードは、管理対象接続および接続ハンドルの割り当て時に使われる。	0 以上	
[Resource Username]	resource-principal によって識別されるユーザ名を指定する。管理対象接続および接続ハンドルの割り当て時に使われる。		
[Resource Userpassword]	resource-principal によって識別されるパスワードを指定する。管理対象接続および接続ハンドルの割り当て時に使われる。	文字列	NULL

11 接続

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- リモート アドレス
- プロトコル
- 接続時間
- 受信されたバイト数
- 送信されたバイト数
- 受信されたメッセージ
- 送信されたメッセージ

12 [コネクタ]

Administration Console を使用したリソース アダプタのデプロイメント

WebLogic Server Administration Console を使用してリソース アダプタをデプロイするには次の手順に従います。

1. WebLogic Server を起動します。
2. Administration Console を開きます。
3. 作業を行うドメインを開きます。
4. 左ペインで、[デプロイメント] の下の [コネクタ] を選択します。デプロイ済みのコネクタ（リソース アダプタ）が右ペインの [リソース コネクタ] テーブルに表示されます。
5. [新しい Connector Component のコンフィグレーション] を選択します。
6. 次の情報を入力します。
   - [名前]ó 必要に応じてコネクタ コンポーネントのデフォルト名を変更します。
   - [Path]ó リソース アダプタ `.rar` ファイル、またはリソース アダプタ展開ディレクトリ形式を含むディレクトリのフルパスを入力します。例：`c:\myaps\components\myResourceAdapter.rar`
   - [デプロイ]ó リソース アダプタ `.rar` ファイルを作成時にデプロイされたかどうかを示します。
7. [作成] ボタンをクリックします。

新しいリソース アダプタが右ペインの [デプロイメント] テーブルに表示されるようになりました。

Administration Console を使用したデプロイ済みリソース アダプタの参照

デプロイ済みのリソース アダプタを Administration Console で参照するには次の手順に従います。

1. Administration Console の左ペインで、[デプロイメント] の下の [コネクタ]（リソース アダプタ）を選択します。
2. 右ペインの [リソース コネクタ] テーブルでデプロイ済みのコネクタのリストを参照します。

Administration Console を使用したデプロイ済みリソース アダプタのアンデプロイ

WebLogic Server Administration Console からデプロイ済みのリソース アダプタをアンデプロイするには次の手順に従います。

1. Administration Console の左ペインで、[デプロイメント] の下の [コネクタ]（リソース アダプタ）を選択します。
2. [リソース コネクタ] テーブルでアンデプロイするコネクタを選択します。
3. [コンフィグレーション] タブで、[デプロイ] チェックボックスの選択を解除します。
4. [適用] をクリックします。

リソース アダプタをアンデプロイしても、リソース アダプタ名は WebLogic Server から削除されません。リソース アダプタは、Server セッションが終了するまでアンデプロイされた状態が続きます。ただし、アンデプロイ後にリソース アダプタを変更した場合を除きます。サーバを再起動するまで、deploy 引数でデプロイメント名を再利用することはできません。次の項で説明するように、デプロイメントの更新にそのデプロイメント名を再使用できます。

Administration Console を使用したデプロイ済みリソース アダプタの更新

WebLogic Server にデプロイ済みのリソース アダプタの .rar ファイルまたはデプロイメント ディレクトリの内容を更新した場合、更新内容は以下のいずれかを実行するまで WebLogic Server に反映されません。

- サーバを起動します(.rar またはディレクトリを自動的にデプロイする場合)。

  または、

- WebLogic Server Administration Console を使用して、リソース アダプタのデプロイメントを更新します。

WebLogic Server Administration Console から、次の操作を行います。

1. Administration Console の左ペインで、[デプロイメント] の下の [コネクタ]（リソース アダプタ）を選択します。
2. [リソース コネクタ] テーブルで更新するコネクタを選択します。
3. 必要に応じてコネクタ名とデプロイ ステータスを更新します。
4. [適用] をクリックします。

詳細については、『WebLogic J2 EE コネクタ アーキテクチャ』を参照してください。

[コンフィグレーション]

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	アプリケーションの名前を指定する。	文字列	NULL
[URI]			
[パス]	アプリケーションの絶対パスを設定する。		
[デプロイ順]	このアプリケーションがデプロイされる順序を設定する。		
[デプロイ]	アプリケーションのデプロイメントを有効にするか、それとも無効にするかを設定する。		選択されていない

13 実行時コネクタ接続プール

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- [リモート アドレス]
- [プロトコル]
- [接続時間]
- [受信バイト数]
- [送信バイト数]
- [受信メッセージ数]
- [送信メッセージ数]

14 ConsoleLink

[プリファレンス]

[一般]

属性	説明	値の範囲	デフォルト値
[自動更新間隔]	コンソールの自動更新時間を設定する。	整数（秒数）	10
[最後に選択したタブの位置を保存]	最後に選択したタブの位置に戻る機能を有効または無効にする。	ブール 有効 = 選択 無効 = 未選択	選択
[ナビゲーション ツリーを使用]	コンソールでナビゲーション ツリーを使用する機能を有効または無効にする。	ブール 有効 = 選択 無効 = 未選択	選択

15 変換

以前のバージョンの Weblogic Server を使用していた場合は、weblogic.properties ファイルをドメインに変換する必要があります。1 つの XML コンフィグレーション ファイルで定義されているリソースが、1 つのドメインを構成します。

この節では、.properties ファイルを変換する以下の方法を説明します。

- Administration Console による変換

Administration Console による変換

変換プロセスの最初のステップでは、WebLogic がインストールされているルート ディレクトリを確認します。変換するすべての weblogic.properties ファイルは、このルートの下になければなりません。また、それらのファイルは、このサーバが動作している同じマシンに配置されている必要があります。

サーバのファイル システムをナビゲートするリンクを使用すると、weblogic のルートを見つけることができます。ルートが見つかったら、その隣のアイコンをクリックして、次のステップに進みます。

変換ユーティリティでは、サーバおよびクラスタとして識別されているエンティティのリストが提供されます。

WebLogic Server Administration Console で変換ユーティリティを使用するには

- 右ペインの [weblogic.properties のコンバート] をクリックします。
- ペインの左側のアイコンを使用し、ツリーをナビゲートすることで、ルート ディレクトリ（グローバルの weblogic.properties ファイルの格納場所）を選択します。
- クラスタおよびサーバの weblogic.properties ディレクトリを選択します。
- [コンバート] ボタンをクリックします。

アプリケーションに必要なドメインがすべて作成されるまで、weblogic.properties ファイルの変換を続けます。

SSL セキュリティ ファイル

SSL セキュリティ ファイルは、従来のプロパティ ファイルでは次のように設定されています。

```
weblogic.security.certificate.server=democert.pem
weblogic.security.key.server=demokey.pem
weblogic.security.certificate.authority=ca1024.pem
weblogic.security.clientRootCA=ca.pem
```

これらのファイルは、config\<NewDomain>\<ServerName>（サーバ別のディレクトリ）にコピーされ、config.xml で次のように反映されます。

```
<Server Name=....

 <SSL
ServerCertificateFileName=".\config\<NewDomain>\democert.pem"
ServerKeyFileName=".\config\<NewDomain>\demokey.pem"
TrustedCAFileName=".\config\<NewDomain>\ca.pem"
ServerCertificateChainFileName=".\config\<NewDomain>\ca1024.pem"
....

 >

 </Server>
```

weblogic.properties で指定されている SSL セキュリティ ファイルが従来のサーバ別ディレクトリにない場合、それらのファイルは config.xml で設定されません。したがって、config\<NewDomain>\<ServerName>（サーバ別のディレクトリ）にコピーして、config.xml で設定する必要があります。

サーブレット

weblogic.properties で登録されているすべてのサーブレットが、1 つの Web アプリケーションに変換されます。

変換ツールでは、web.xml や weblogic.xml などの必要なファイルが、次のディレクトリの 1 つに作成されます。

1. config\<NewDomain>\applications\DefaultWebApp_<ServerName>\WEB-INF（プロパティ ファイルにすでに宣言されたものがなければ、デフォルトの Web アプリケーションとして作成される）
2. config\<NewDomain>\<Server_Name>\WEB-INF（デフォルトの Web アプリケーションがすでに宣言されている場合）

weblogic の内部サーブレットを除いて、従来の weblogic.properties で weblogic.httpd.register として個別に登録されているすべてのサーブレット クラスは、web.xml で指定されているとおりに新しい Weblogic Server のツリー構造にコピーする必要があります。

たとえば、web.xml で次のように指定されている場合、

```
<servlet>
<servlet-class>weblogic.hello.HelloworldServlet</servlet-class>
</servlet>
```

サーブレット クラスの Hellowordlservlet.class は、対応する WEB-INF\classes\weblogic\hello ディレクトリにコピーする必要があります。カレント ディレクトリは、Weblogic Server が新しいコンフィグレーションで起動されるディレクトリです。

EJB の jar ファイルと Web アプリケーションの war ファイル

weblogic.properties の weblogic.ejb.deploy および weblogic.httpd.webApp.<webAppName> で相対的なディレクトリが指定されている場合は、それらの jar ファイルと war ファイルを新しい Weblogic Server のツリー構造にコピーする必要があります。

例 :

```
weblogic.ejb.deploy=weblogic\ejb\HelloEJB.jar
```

この jar ファイルは、.\weblogic\ejb\ ディレクトリにコピーする必要があります。

「.」は、サーバが新しいコンフィグレーションで起動されるディレクトリを示しています。

16 [このビューをカスタマイズ ...]

[Table Show]

1. [Table Show List] に項目を追加するには、[選択可] リストから項目を選択し、矢印ボタンを使って [選択済み] リストに移動します。
2. [Table Show List] から項目を削除するには、[選択済み] リストから項目を選択し、矢印ボタンを使って [選択可] リストに移動します。
3. [適用] をクリックして、変更を有効にします。

[Table Sort]

1. [Table Sort List] に項目を追加するには、[選択可] リストから項目を選択し、矢印ボタンを使って [選択済み] リストに移動します。
2. [Table Sort List] から項目を削除するには、[選択済み] リストから項目を選択し、矢印ボタンを使って [選択可] リストに移動します。
3. [適用] をクリックして、変更を有効にします。

17 カスタム レルム

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	AccountingRealm など、カスタム セキュリティ レルムの名前を指定する。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[レルム クラス名]	セキュリティ レルムが所属する Java クラスの名前を返す。Java クラスは、WebLogic Server の CLASSPATH に含まれている必要がある。	この属性は変更不可	NULL
[コンフィグレーション情報]	セキュリティ ストアへの接続に必要な情報。	key=value のリスト	NULL
[パスワード]	カスタム セキュリティ レルムと共に使用されるセキュリティ ストアのパスワードを設定する。	文字列	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

18 デプロイメント記述子エディタ

デプロイメント記述子エディタの詳細については、以下の節を参照してください。

BEA WebLogic J2EE コネクタ アーキテクチャの属性記述

EJB デプロイメント記述子

RDBMS デプロイメント記述子

Web アプリケーション デプロイメント記述子エディタ ヘルプ

19 ドメイン

以下の手順では、Administration Console を利用してドメイン情報表示用の属性を設定する方法について説明します。ドメインの詳細については、『WebLogic Server 管理者ガイド』の「ログ メッセージを使用した WebLogic Server の管理」を参照してください。

ドメインの状態の表示

1. 左ペインで、ユーザの「ドメイン」を表すインスタンス ノード（インストール時に WebLogic 管理ドメイン名に mydomain を指定し、サーバ名に myserver を指定した場合は myDomain）をクリックします。右ペインにドメイン ダイアログが表示され、このドメインに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [ドメインの状態を参照します] テキスト リンクをクリックします。右ペインにドメインの状態ビューが表示され、[サーバ] テーブルとこのドメインのログが示されます。

ドメイン ログの表示

1. 左ペインで、ユーザの「ドメイン」を表すインスタンス ノード（myDomain）を右クリックします。ポップアップ メニューが開きます。
2. [ドメインログを参照します] をクリックします。右ペインにログが表示されます。

アプリケーション管理設定の編集

1. 左ペインで、ユーザの「ドメイン」を表すインスタンス ノード (myDomain) をクリックします。右ペインにドメイン ダイアログが表示され、このドメインに関連するタブが示されます。
2. [アプリケーション] タブをクリックします。
3. [自動更新間隔] 属性フィールドの値を編集します。[自動デプロイを有効化] チェックボックスをクリックすると、アプリケーションはドメインにインストールされた時点で自動的にデプロイされることになります。
4. [適用] をクリックして、変更を保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[ドメイン名]	当該コンフィグレーションの名前を返す。	カンマとスペースを含まない 256 文字までの英数字	mydomain
［コンソール有効化］	コンソールを有効または無効にするのに用いられる。		選択されている
［コンソール コンテキスト パス］	WebLogic Server コンソールのコンテキストパスを設定するのに用いられる。		console

[JTA]

属性	説明	値の範囲	デフォルト値
[タイムアウト秒数]	強制ロールバックされるまで、トランザクションがアクティブ状態を継続できる時間（秒単位）。	整数	30 秒
[トランザクションを保持する最長時間]	トランザクションコーディネータがトランザクションの完了を試み続ける最長時間（秒単位）。	整数	86400 秒
[beforeCompletion の反復上限]	強制ロールバックされるまでに処理される `beforeCompletion` コールバックの回数。	整数	10
[最大トランザクション数]	個々のサーバ上で一度にアクティブになりうるトランザクションの最大数。	整数	10000
[ユニーク名の最大数]	サーバが一度にトラッキングできるユニークなトランザクション名の最大数。	整数	1000

属性	説明	値の範囲	デフォルト値
[ヒューリスティックを無視]	ヒューリスティックな結果が得られたトランザクションをすべて無視するようにトランザクション マネージャがリソースに指示するかどうかを指定するブール値。	True = 選択されていない False = 選択されている	選択されていない

[SNMP]

属性	説明	値の範囲	デフォルト値
[有効化]	SNMP サービスは SNMP エージェント機能を提供し、管理サーバの一部を構成する。SNMP エージェントはドメインのすべての WebLogic リソースを監視する。コンフィグレーションの変更を有効にするには、管理サーバを再起動する必要がある。	True = 有効 False = 無効	False
[SNMP ポート]	このポートは、WebLogic SNMP エージェントが SNMP マネージャから着信する要求をリスンするポート。	有効なリスン ポート	161

属性	説明	値の範囲	デフォルト値
[MIB データの更新間隔]	SNMP エージェントはすべての属性値のキャッシュを維持管理し、そのキャッシュから属性値を取得することでマネージャからの要求に応答する。[MIB データの更新間隔] は、SNMP エージェントがこのキャッシュを完全に更新する間隔（秒単位）。SNMP エージェントは更新を行う際に、WebLogic SNMP MIB に記述されているすべての WebLogic 属性を取得する。	整数（秒単位）	120
[サーバ状態チェック間隔係数]	SNMP エージェントは、[MIB データの更新間隔] にこの数字を乗じて、ドメイン内の管理対象サーバが稼働しているかどうかを判定するためのチェックを行う頻度を決定する。なお、エージェントはこの値を MIB キャッシュから取得する。[サーバ 状態 チェック間隔係数] が 1 ならば、WebLogic SNMP エージェントは、[MIB データの更新間隔] に定義されている間隔で、管理対象サーバが稼働しているかどうかを調べる。	正の整数	MIB キャッシュから取得

属性	説明	値の範囲	デフォルト値
[コミュニティ プレフィックス]	この文字列は、SNMP マネージャと通信するための平文パスワードの役目を果たす SNMP コミュニティ名を作成するために使われる。SNMP マネージャから送られるコミュニティ プレフィックスがこの属性に設定されている値と一致しない場合には、SNMP エージェントは要求側に authenticationFailuretrap を返す。	文字列 値が `community_prefix@server_name` の形式の場合、エージェントは指定された管理対象サーバのデータだけを返す。値が `community_prefix@domain_name` の形式の場合、エージェントはドメイン内の全サーバのデータを返す。SNMP マネージャが `community_prefix` だけを送信する場合、エージェントは管理サーバのデータだけを取り出す。	Null
[デバッグ レベル]	管理者へ送られるデバッグ メッセージのレベルを設定する。この値が 0 に設定されると、デバッグ メッセージは作成されない。値が正であれば、エージェント コードの実行内容を記述したメッセージが作成される。この値が大きいほど、メッセージの内容は詳細になる。	0 ～ 3 の整数	0

[ログ]

属性	説明	値の範囲	デフォルト値
[ファイル名]	ログ ファイルの名前を返す。	文字列	config/mydomain/logs/wl-domain.log
[ローテーション タイプ]	使用するローテーションのタイプを選択できるようにする。	リスト	なし
[最小ファイルサイズ]	1 ファイルのサイズのしきい値を設定する。	整数	500
[ファイルローテーション間隔]	1 ファイルの制限時間を設定する。	整数	24
[ファイル数の制限]	ファイル制限の状態を判定するために使用する。	選択されている = 有効 選択されていない = 無効	選択されていない
[ファイル数]	作成するファイル数の上限を設定する。	整数	7

[アプリケーション]

属性	説明	値の範囲	デフォルト値
[自動デプロイを有効化]	自動デプロイメント機能を設定する。	選択されている = 有効 選択されていない = 無効	選択されている
[自動更新間隔]	アプリケーションの自動更新の間隔を設定する。	整数	3000

[セキュリティ]

[一般]

属性	説明	値の範囲	デフォルト値
[監査プロバイダ クラス]	監査プロバイダを指定できるようにする。	有効な監査プロバイダ クラスの名前	Null
[ゲスト不可]	デフォルトのユーザ「Guest」を無効にする。	ブール True = 選択されている False = 選択されていない	選択されている

[ファイル レルム]

属性	説明	値の範囲	デフォルト値
[キャッシング レルム]	セキュリティを設定または確認したいキャッシング レルムを選択できるようにする。	リスト	何も選択されていない
[最大ユーザ数]	ファイル レルムでサポートされるユーザの最大数	最大値は 10,000、最小値は 1。最大値は適用されない。つまり、最大値を設定すると警告が出る。	1000
[最大グループ数]	ファイル レルムでサポートされるグループの最大数	最大値は 10,000、最小値は 1。最大値は適用されない。つまり、最大値を設定すると警告が出る。	1000
[最大 ACL 数]	ファイル レルムでサポートされる ACL の最大数	最大値は 10,000、最小値は 1。最大値は適用されない。つまり、最大値を設定すると警告が出る。	1000

[パスワード]

属性	説明	値の範囲	デフォルト値
[最小パスワード文字数]	パスワードに必要な文字数を設定する。パスワードは 8 文字以上でなければならない。	8 文字以上	8
[ロックアウト有効化]	この属性を選択すると、無効なログインが試みられたときにユーザ アカウントがロックされる。	ブール 有効 = 選択されている 無効 = 選択されていない	選択されている

属性	説明	値の範囲	デフォルト値
[ロックアウトしきい値]	ユーザが何回ログインを試みて失敗するとアカウントがロックされるかを示す回数。それ以上そのアカウントにアクセスしようとすると（ユーザ名とパスワードの組合せが正しくても）セキュリティ例外が発生する。システム管理者によって明示的にロックが解除されるか、あるいはロックアウト継続時間が経過したあとに別のログインが試みられるまで、アカウントはロックされたままになる。なお、[ロックアウト リセット遅延] 属性で定義された時間内に無効なログインを繰り返し試みないかぎり、失敗した回数（[LockoutThreshold] 属性の値を上限とする）には数えられない点に注意。	1 ～ 99,999	5
[ロックアウト遅延]	[ロックアウト リセット遅延] 属性で指定された時間内に無効なログインが所定の回数以上試みられたときにユーザ アカウントがロックされる時間（分単位）を指定。	0 ～ 999999 分	30

属性	説明	値の範囲	デフォルト値
[ロックアウト リセット遅延]	どれだけの時間（分単位）内に無効なログインの試みが繰り返されるとユーザのアカウントがロックされるかを指定する。このフィールドで定義された時間内に、[ロックアウトしきい値] フィールドに定義された回数だけ無効なログインを試みると、アカウントがロックされる。	整数	5 分
[ロックアウト キャッシュ サイズ]	使用しなかった無効なログインの試みを一時的に保存しておくためのキャッシュのサイズを設定するのに使われる。	0 〜 10	5

[詳細設定]

属性	説明	値の範囲	デフォルト値
[接続フィルタ]	セキュリティ レルムの接続フィルタを選択できるようにする。	リスト	何も選択されていない
[結果バッチサイズ]			
[すべてのログをチェック]			

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域	英数字の文字列	Null

ドメインの詳細については、『WebLogic Server 管理者ガイド』の「ログ メッセージを使用した WebLogic Server の管理」を参照してください。

20 ドメイン ログ フィルタ

以下の手順は、Administration Console を使ってドメイン ログ フィルタをコンフィグレーションおよび管理する方法を説明します。ドメイン ログ フィルタの詳細にについては、『WebLogic Server 管理ガイド』の「ログ メッセージを使用した WebLogic Server の管理」を参照してください。

新しいドメイン ログ フィルタの作成

1. 左ペインの [ドメイン ログ フィルタ] ノードをクリックします。右ペインに [ドメイン ログ フィルタ] テーブルが表示され、ドメイン内のすべてのログ フィルタが示されます。
2. [新しい Domain Log Filter の作成] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するドメイン ログ フィルタの設定に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力、および [重大度] リストで値を選択します。
4. [作成] ボタンをクリックして、[名前] フィールドで指定した名前でドメイン ログ フィルタのインスタンスを作成します。新しいインスタンスが左ペインの [ドメイン ログ フィルタ] ノードの下に追加されます。
5. [対象] タブをクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、これまでの変更を保存します。

ドメイン ログ フィルタのクローンの作成

1. 左ペインの [ドメイン ログ フィルタ] ノードをクリックします。右ペインに [ドメイン ログ フィルタ] テーブルが表示され、ドメイン内のすべてのログ フィルタが示されます。
2. クローンを作成するドメイン ログ フィルタの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、ドメイン ログ フィルタのクローンの作成に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力、および [重大度] リストで値を選択します。
4. [作成] ボタンをクリックして、[名前] フィールドで指定した名前でドメイン ログ フィルタのインスタンスを作成します。新しいインスタンスが左ペインの [ドメイン ログ フィルタ] ノードの下に追加されます。
5. [対象] タブをクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、これまでの変更を保存します。

ドメイン ログ フィルタの削除

1. 左ペインの [ドメイン ログ フィルタ] ノードをクリックします。右ペインに [ドメイン ログ フィルタ] テーブルが表示され、ドメイン内のすべてのログ フィルタが示されます。
2. 削除するドメイン ログ フィルタの行の [削除] アイコンをクリックします。右ペインにダイアログが表示され、削除要求を確認するよう求められます。
3. [はい] をクリックして、ドメイン ログ フィルタを削除します。[ドメイン ログ フィルタ] ノードの下のドメイン ログ フィルタ アイコンが削除されます。

[コンフィグレーション]

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	ドメイン ログ フィルタの名前を設定する。	カンマの含まれていない、256 文字までの英数字の文字列	MyDomain Log Filter
[重大度]	ドメイン ログ フィルタで報告される重要度を設定する。	リスト	Error

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[Server]	ドメイン ログ フィルタでメッセージを受信するサーバのリストを提供する。	リスト	NULL

[サブシステム]

属性	説明	値の範囲	デフォルト値
[サブシステム名]	ドメイン ログ フィルタでメッセージを受信するサブシステムのリスト。	リスト	NULL

[ユーザ]

属性	説明	値の範囲	デフォルト値
[ユーザ ID]	ID のリストからユーザがユーザ ID を選択できるようにする。	リスト	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

ドメイン ログ フィルタの詳細については、『WebLogic Server 管理者ガイド』の「ログ メッセージを使用した WebLogic Server の管理」を参照してください。

21 EJB コンポーネント

以下では、新しい EJB をインストールまたはコンフィグレーションするための属性を Administration Console を使用して設定する手順を説明します。EJB については、『WebLogic エンタープライズ JavaBeans プログラマーズ ガイド』を参照してください。

[新しい EJB のインストール]

1. 左ペインの [デプロイメント] ノードを展開します。
2. [EJB] をダブルクリックします。すべてのデプロイ済み EJB を示す [EJB デプロイメント] テーブルが右ペインに表示されます。
3. [新しい EJB をインストール] をクリックし、サーバに新しい EJB をインストールします。[アプリケーションのアップロードとインストール] ダイアログが右ペインに表示されます。
4. テキスト入力フィールドに . jar ファイルのパスを入力するか、[参照] ボタンをクリックしてファイル システムを参照し、インストールする .jar ファイルを選択します。
5. [アップロード] をクリックして、.jar ファイルをインストールします。新しい EJB が左ペインの [EJB] ノードの下に追加されます。

[新しい EJB のコンフィグレーション]

1. 左ペインの [デプロイメント] ノードを展開します。
2. [EJB] をダブルクリックします。すべてのデプロイ済み EJB を示す [EJB デプロイメント] テーブルが右ペインに表示されます。
3. [新しい EJB のコンフィグレーション] をクリックし、新しいコンポーネントを作成します。右ペインに表示される EJB コンポーネントで、新しい EJB をコンフィグレーションします。
4. EJB コンポーネント ペインの [一般] タブで以下を設定します。
   a. EJB コンポーネントの名前を [名前] フィールドに入力します。
   b. EJB コンポーネントで使用するパスと Uniform Resource Identifier (URI) を [パス] フィールドに入力します。
   c. コンポーネントをサーバにデプロイする順番を [デプロイ順] フィールドに入力します。
   d. [デプロイ] チェックボックスをクリックし、コンポーネントのデプロイメント ステータスを設定します。
5. コンパイラ オプションをコンフィグレーションするには、EJB コンポーネント ペインの [EJBC オプション] タブで、以下を設定します。
   a. 使用する Java コンパイラを [Java コンパイラ] フィールドに入力します。
   b. 生成されたファイルを格納する場所のパスを [テンポラリ パス] フィールドに入力します。
   c. rmic オプションを [追加 Rmic オプション] フィールドに入力します。
   d. [生成したソースファイルを残す] チェックボックスをオンにすると、生成したソース ファイルを残すことができます。
6. [作成] をクリックし、コンポーネントを作成します。

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	EJB コンポーネントの名前を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[URL]	EJB コンポーネントの URL を返す。	有効な URL	NULL
[アプリケーション]	アプリケーションの名前を返す。	文字列	NULL
[インスタンス数]	EJB コンポーネントのインスタンス数を返す。	整数	NULL
[Java コンパイラ]	このコンポーネントに使用する Java コンパイラを設定する。	有効な Java コンパイラ	javac
[テンポラリ パス]	`ejbc` によって生成されるファイルの格納場所を設定する。	文字列	NULL
[追加 Rmic オプション]	ユーザが rmic オプションを設定できるようにする。	文字列	NULL
[生成したソース ファイルを残す]	生成されたソース ファイルを保存する機能を有効 / 無効にする。	ブール 選択されている = 有効 選択されていない = 無効	選択されていない

属性	説明	値の範囲	デフォルト値
[パス]	このコンポーネントのパスと、アプリケーション コンポーネントを指す Uniform Resource Identifier (URI) を設定する。 パスには、展開ディレクトリ形式でのコンポーネントの直接名、または EAR ファイル名を指定できる。	文字列	NULL
[デプロイ]	コンポーネントのデプロイメントの状態を設定する。	ブール デプロイ = 選択されている アンデプロイ = 選択されていない	選択
[デプロイ順]	ユーザがコンポーネントをサーバにデプロイする順番を設定できる。		NULL

[対象]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	この EJB がデプロイされるサーバを選択できる。	リスト	未選択
[対象クラスタ]	この EJB がデプロイされるクラスタを選択できる。	リスト	未選択

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

EJB については、『WebLogic エンタープライズ JavaBeans プログラマーズ ガイド』を参照してください。

22 実行時 EJB コンポーネント

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- アプリケーション
- 名前
- URI

23 EJB ホーム

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- アプリケーション
- 名前
- URI

25　EJB デプロイメント記述子

weblogic-ejb-jar.xml デプロイメント記述子

この章では、バージョン 6.1 の WebLogic Server に固有の XML 要素について説明します。これらの要素は `weblogic600-ejb-jar.xml` ファイルを構成し、WebLogic Server EJB コンテナ内のデプロイメント記述子を定義するために使われるものです。これらの要素は、WebLogic Server Administration Console のほぼ同じ名前のフィールドに対応します。これらのデプロイメント記述子はバージョン 2.0 の EJB に対して使用します。

allow-concurrent-calls

値の範囲	`true	false`
デフォルト値	`false`
要件	ステートフル セッション Bean のインスタンスがメソッド呼び出しを処理している間に、（同時実行される）別のメソッド呼び出しがサーバに到着したとき、サーバは `RemoteException` を送出する必要がある。
親要素	`weblogic-enterprise-bean` `stateful-session-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`allow-concurrent-calls` 要素では、ステートフル セッション Bean のインスタンスでメソッドの同時呼び出しを許可するかどうかを指定します。デフォルトでは、`allows-concurrent-calls` の値は `false` です。この要素の値を `true` に設定すると、メソッドの同時呼び出しが EJB コンテナによってブロックされ、直前の呼び出しが完了した時点で次のメソッドの処理に移行します。

cache-type

値の範囲	NRU | LRU
デフォルト値	NRU
要件	
親要素	weblogic-enterprise-bean stateful-session-cache
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

この要素では、EJB のキャッシュ タイプを設定します。

connection-factory-jndi-name

値の範囲	
デフォルト値	config.xml 内の weblogic.jms.MessageDrivenBeanConnectionFactory
要件	ステートフル セッション Bean のインスタンスがメソッド呼び出しを処理している間に、（同時実行される）別のメソッド呼び出しがサーバに到着したとき、サーバは RemoteException を送出する必要がある。
親要素	weblogic-enterprise-bean message-driven-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

connection-factory-jndi-name 要素では、メッセージ駆動型 Bean がキューおよびトピックを作成するためにルックアップする JMS 接続ファクトリの JNDI 名を指定します。この要素を指定しない場合、config.xml 内の weblogic.jms.MessageDrivenBeanConnectionFactory がデフォルト値となります。

concurrency-strategy

値の範囲	`Exclusive	Database	ReadOnly`
デフォルト値	`Database`
要件	省略可能な要素。エンティティ EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`entity-cache`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

この要素では、エンティティ Bean への同時アクセスをコンテナでどのように管理するかを指定します。この要素には、次の 3 つの値のいずれかを設定できます。

- `Exclusive` は、WebLogic Server バージョン 3.1 ～ 5.1 でのデフォルトのロック動作です。Bean がトランザクションと関連付けられているとき、WebLogic Server はキャッシュされたエンティティ EJB のインスタンスに排他的ロックをかけます。EJB のインスタンスに対するほかのリクエストは、トランザクションが完了するまでブロックされます。
- `Database` を指定すると、WebLogic Server はエンティティ EJB に対するロック要求を、基盤のデータストアに委ねます。同時実行性オプションが `Database` のとき、WebLogic Server はトランザクションに関係するエンティティ EJB の中間結果をキャッシュしません。
- `ReadOnly` は、エンティティ EJB が変更されないことを指定します。WebLogic Server は `read-timeout-seconds` パラメータの設定に基づいて、同時実行性が `ReadOnly` に設定された Bean に対して `ejbLoad()` メソッドを呼び出します。

db-is-shared

値の範囲	`true	false`
デフォルト値	`true`
要件	省略可能な要素。エンティティ EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`db-is-shared` 要素は、エンティティ Bean だけに適用されます。この要素の値を `true` に設定すると、WebLogic Server は EJB のデータをトランザクションの間に変更できないことを想定し、各トランザクションの開始時にデータを再ロードします。`false` に設定すると、WebLogic Server は永続ストア内の EJB データに排他的にアクセスできることを想定します。

delay-updates-until-end-of-tx

値の範囲	true | false
デフォルト値	true
要件	省略可能な要素。エンティティ EJB に対してのみ有効。
親要素	weblogic-enterprise-bean, entity-descriptor, persistence
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

トランザクションの完了時に、トランザクションを構成する全 Bean の永続ストアを更新するには、この要素を true（デフォルト）に設定します。このように設定すると、不必要な更新が回避されるためパフォーマンスが向上するのが一般的です。ただしこの設定では、データベース トランザクション内部でのデータベース更新の順序は保持されません。

データストアで分離レベル TRANSACTION_READ_UNCOMMITTED を使用する場合に、ほかのデータベース ユーザが、進行中のトランザクションの中間結果を確認できるようにすることができます。この場合、各メソッド呼び出しの完了時に Bean の永続ストアが更新されるように、delay-updates-until-end-of-tx の値を false に設定します。

注意： delay-updates-until-end-of-tx を false に設定すると、各メソッドの呼び出し後にデータベースの更新はデータベースに送られるだけで、コミットはされません。更新はトランザクションが完了した時点でデータベースにコミットされるか、またはロールバックされます。

description

値の範囲	なし
デフォルト値	なし
要件	なし
親要素	weblogic-enterprise-bean, transaction-isolation method
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

description 要素には、親の要素について説明するテキストを指定します。

destination-jndi-name

値の範囲	有効な JNDI 名
デフォルト値	なし
要件	message-driven-descriptor に必要。
親要素	weblogic-enterprise-bean message-driven-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

destination-jndi-name 要素では、メッセージ駆動型 Bean を、WebLogic Server JNDI ツリーにデプロイされた実際の JMS キューまたはトピックと関連付けるために使われる JNDI 名を指定します。

ejb-name

値の範囲	ejb-jar.xml に定義された EJB の名前
デフォルト値	なし
要件	method スタンザ内で必須の要素。名前は NMTOKEN の辞書ルールに従う必要がある。
親要素	weblogic-enterprise-bean transaction-isolation method
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

ejb-name 要素では、WebLogic Server が分離レベル プロパティを適用する EJB の名前を指定します。この名前は、ejb-jar ファイルのデプロイメント記述子によって割り当てられます。同じ ejb.jar ファイルに定義されるエンタープライズ Bean 間で一意な名前を指定する必要があります。エンタープライズ Bean のコードは名前に依存しないため、アプリケーションをアセンブルする過程で、エンタープライズ Bean の機能を変更せずに名前を変更できます。デプロイメント記述子に定義される EJB 名と、デプロイヤがエンタープライズ Bean のホームに割り当てる JNDI 名との間に構造的な関係はありません。

ejb-reference-description

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	weblogic-enterprise-bean reference-description
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

resource-description スタンザは、ejb-jar.xml に定義されるリソース参照を、WebLogic Server で利用可能な実際のリソースの JNDI 名にマッピングします。

- ejb-ref-name には、リソース参照名を指定します。これは、EJB プロバイダがデプロイメント ファイル ejb-jar.xml の内部に定義する参照です。
- jndi-name には、WebLogic Server で利用可能な実際のリソース ファクトリの JNDI 名を指定します。

ejb-ref-name

値の範囲	なし
デフォルト値	なし
要件	省略可能な要素。

親要素	`weblogic-enterprise-bean`
	`reference-description`
	`ejb-reference-description`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`resource-description` スタンザは、`ejb-jar.xml` に定義されるリソース参照を、WebLogic Server で利用可能な実際のリソースの JNDI 名にマッピングします。

ejb-local-reference-description

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	`weblogic-enterprise-bean` `reference-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

ejb-local-reference-description 要素は、EJB ローカルの参照において Bean によって参照される EJB の、WebLogic Server における JNDI 名をマッピングするために使用します。

enable-call-by-reference

値の範囲	`true	false`
デフォルト値	`true`
要件	省略可能な要素。

親要素	`weblogic-enterprise-bean`
	`reference-description`
	`ejb-reference-description`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

デフォルトでは、同じサーバの内部から呼び出される EJB メソッドでは、参照によって引数を渡します。この方式ではパラメータがコピーされないため、メソッド呼び出しのパフォーマンスが向上します。

`enable-call-by-reference` 要素の値を `False` に設定すると、EJB メソッドへのパラメータは、EJB 1.1 仕様に準拠する形式（値渡し）でコピーされます。EJB がサーバの内部からではなくリモートで呼び出されるときは常に、値渡しが必要です。

entity-cache

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	entity-cache 要素は省略可能で、エンティティ EJB に対してのみ有効である。
親要素	weblogic-enterprise-bean, entity-descriptor
デプロイメントファイル	weblogic-ejb-jar.xml

機能

entity-cache 要素では、WebLogic Server の内部でエンティティ EJB のインスタンスをキャッシュするために使われる以下のオプションを定義します。

- max-beans-in-cache
- idle-timeout-seconds
- read-timeout-seconds
- concurrency-strategy

entity-clustering

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。クラスタ内の EJB に対してのみ有効。
親要素	`weblogic-enterprise-bean`, `entity-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`entity-clustering` 要素では以下のオプションを使用して、エンティティ Bean が WebLogic クラスタにどのようにレプリケートされるかを指定します。

- home-is-clusterable
- home-load-algorithm
- home call-router-class-name

entity-descriptor

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	*.jar* 内の各エンティティ EJB に対して、1 つの `entity-descriptor` スタンザが必要である。
親要素	`weblogic-enterprise-bean`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`entity-descriptor` 要素では、エンティティ Bean に適用される以下のデプロイメント パラメータを指定します。

- pool
- entity-cache
- lifecycle
- persistence
- entity-clustsering

finders-load-bean

値の範囲	true | false
デフォルト値	`true`
要件	省略可能な要素。CMP エンティティ EJB に対してのみ有効。
親要素	`weblogic-enterprise-bean`, `entity-descriptor`, `persistence`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`finders-call-ejbload` 要素は、ファインダ メソッドへの呼び出しによって Bean への参照が返された後で、WebLogic Server が EJB をキャッシュにロードするかどうかを決定します。この要素の値を `true` に設定すると、Bean への参照がファインダによって返された場合に、WebLogic は Bean を直ちにキャッシュにロードします。`false` に設定すると、メソッドが最初に呼び出されるまで、WebLogic Server は Bean をキャッシュに自動的にロードしません。この動作は EJB 1.1 仕様に準拠しています。

home-call-router-class-name

値の範囲	有効なルータ クラス名
デフォルト値	NULL
要件	省略可能な要素。クラスタ内のエンティティ EJB およびステートフル セッション EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean, entity-descriptor, entity-clustering`
	および
	`weblogic-enterprise-bean stateful-session-descriptor stateful-session-clustering`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`home-call-router-class-name` 要素では、Bean のメソッド呼び出しをルーティングするために使用するカスタム クラスの名前を指定します。このクラスは `weblogic.rmi.extensions.CallRouter()` を実装しなければなりません。この要素の値を指定した場合、各メソッド呼び出しの前にこのクラスのインスタンスが呼び出されます。このとき、ルータ クラスでメソッド パラメータに基づいてルーティング先のサーバを選択する機会が設けられます。ルータ クラスはサーバ名または NULL のどちらかを返します。NULL が返された場合、サーバの選択は現在のロード アルゴリズムに基づいて行われます。

home-is-clusterable

値の範囲	`true	false`
デフォルト値	`true`
要件	省略可能な要素。クラスタ内のエンティティ EJB およびステートフル セッション EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean,`
	`  entity-descriptor,`
	`  entity-clustering`
	および
	`weblogic-enterprise-bean`
	`  stateful-session-descriptor`
	`  stateful-session-clustering`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`home-is-clusterable` 要素の値が `true` のとき、クラスタ内の複数の WebLogic Server から EJB をデプロイすることができます。ホーム スタブへの呼び出しは、この Bean がデプロイされるサーバ間でロードバランシングされ、Bean が動作しているサーバに到達できない場合、呼び出しは自動的に、Bean が動作している別のサーバへとフェイルオーバされます。

home-load-algorithm

値の範囲	round-robin | random | weight-based
デフォルト値	weblogic.cluster.defaultLoadAlgorithm の値
要件	省略可能な要素。クラスタ内のエンティティ EJB およびステートフル セッション EJB に対してのみ有効。
親要素	weblogic-enterprise-bean, entity-descriptor, entity-clustering および weblogic-enterprise-bean stateful-session-descriptor stateful-session-clustering
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

home-load-algorithm 要素では、EJB ホームのレプリカ間でロードバランシングを行うためのアルゴリズムを指定します。このプロパティを定義しない場合、WebLogic Server はサーバ プロパティ weblogic.cluster.defaultLoadAlgorithm によって指定されるアルゴリズムを使用します。

home-load-algorithm には、以下のいずれかの値を設定できます。

- round-robin: Bean が動作しているサーバ間で、逐次的な方式でロードバランシングが行われます。
- random: EJB ホームのレプリカは、Bean が動作しているサーバを無作為に選択してデプロイされます。
- weight-based: EJB ホームのレプリカは、その時点での各サーバの作業負荷に応じてホスト サーバにデプロイされます。

idle-timeout-seconds

値の範囲	1 から *maxSeconds* までの範囲内
デフォルト値	600
要件	省略可能な要素。
親要素	weblogic-enterprise-bean, entity-descriptor, entity-cache および weblogic-enterprise-bean, stateful-session-descriptor, stateful-session-cache
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

idle-timeout-seconds 要素では、ステートフル EJB をキャッシュに保持する最長の時間を定義します。この時間が経過した後で、キャッシュ内の Bean の数が max-beans-in-cache の制限に達すると、WebLogic Server は Bean のインスタンスを削除します。削除された Bean インスタンスはパッシブ化されます。

initial-beans-in-free-pool

値の範囲	0 から *maxBeans* までの範囲内
デフォルト値	0
要件	省略可能な要素。ステートレス セッション EJB に対してのみ有効。
親要素	weblogic-enterprise-bean, pool
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

initial-beans-in-free-pool の値を指定する場合、プールの初期サイズを設定します。WebLogic Server は、起動時にすべての Bean クラスについて、指定された数の Bean インスタンスをフリー プールに格納します。このようにフリー プールを満たしておくことにより、新しいインスタンスを生成しなくても Bean に対しての初期リクエストを処理できるため、EJB の初期応答時間が改善されます。

initial-context-factory

値の範囲	true | false
デフォルト値	weblogic.jndi.WLInitialContextFactory
要件	ステートフル セッション Bean のインスタンスがメソッド呼び出しを処理している間に、（同時実行される）別のメソッド呼び出しがサーバに到着したとき、サーバは RemoteException を送出する必要がある。
親要素	weblogic-enterprise-bean message-driven-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

initial-context-factory 要素では、コンテナが自らの接続ファクトリを作成するために使用する contextFactory を指定します。この要素の値を指定しない場合のデフォルト値は weblogic.jndi.WLInitialContextFactory です。

is-modified-method-name

値の範囲	エンティティ EJB の有効なメソッド名
デフォルト値	なし
要件	省略可能な要素。エンティティ EJB に対してのみ有効。
親要素	`weblogic-enterprise-bean`, `entity-descriptor`, `persistence`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`is-modified-method-name` 要素では、EJB が格納されるときに WebLogic Server が呼び出すメソッドを指定します。指定されたメソッドは、`boolean` 型の値を返す必要があります。メソッドを指定しない場合、WebLogic Server は常に、EJB に変更が加えられたことを想定して EJB を保存します。

メソッドを定義して適切に設定することにより、EJB 1.1 準拠の Bean と、Bean 管理永続性を使用する Bean のパフォーマンスを改善できます。ただし、メソッドの戻り値に誤りがあると、データの不整合に関する問題が生じる可能性があります。

注意： EJB 2.0 仕様に基づく、バージョン 2.0 の CMP エンティティ EJB については、`isModified()` は不要になりました。ただし、BMP EJB およびバージョン 1.1 の CMP EJB では、引き続きこのメソッドが必要です。永続性がコンテナによって管理される EJB 2.0 のエンティティ Bean をデプロイすると、WebLogic Server は EJB のどのフィールドに変更が行われたかを自動的に検出し、該当するフィールドだけを基盤のデータストアに書き込みます。

isolation-level

値の範囲	Serializable | ReadCommitted | ReadUncommitted | RepeatableRead
デフォルト値	なし
要件	省略可能な要素。
親要素	weblogic-enterprise-bean, transaction-isolation
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

isolation-level 要素では、EJB のデータベース操作のすべてに適用される分離レベルを指定します。isolation-level には、以下の値を設定できます。

- TRANSACTION_READ_UNCOMMITTED: トランザクションで、未コミットの更新をほかのトランザクションから確認できます。
- TRANSACTION_READ_COMMITTED: トランザクションで、コミット済みの更新だけをほかのトランザクションから確認できます。
- TRANSACTION_REPEATABLE_READ: トランザクションがデータの一部をいったん読み取ると、同じデータが繰り返し読み取られたときに、ほかのトランザクションによって読み取り後にデータが変更された場合でも、最初に読み取られた同じ値が返されます。
- TRANSACTION_SERIALIZABLE: このトランザクションを複数回にわたって同時実行するときと、シリアル方式で複数回実行するときの効果が同じです。

それぞれの分離レベルの効用およびサポート状況の詳細については、使用しているデータベースのドキュメントを参照してください。

jndi-name

値の範囲	有効な JNDI 名
デフォルト値	なし
要件	`resource-description` および `ejb-reference-description` に必要。

親要素	`weblogic-enterprise-bean`
	および
	`weblogic-enterprise-bean`
	`reference-description`
	`resource-description`
	および
	`weblogic-enterprise-bean`
	`reference-description`
	`ejb-reference-description`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`jndi-name` 要素では、WebLogic Server で利用可能な実際の EJB、リソース、または参照の JNDI 名を指定します。

local-jndi-name

値の範囲	有効な JNDI 名
デフォルト値	なし
要件	Bean がローカル ホームを持つ場合に必要。
親要素	`weblogic-enterprise-bean`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`local-jndi-name` 要素では、Bean のローカル ホームの JNDI 名を指定します。Bean にリモート ホームとローカル ホームの両方が存在する場合、それぞれのホームに対応する 2 つの JNDI 名を Bean に定義する必要があります。

lifecycle

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	`lifecycle` 要素は省略可能である。

親要素	`weblogic-enterprise-bean, entity-descriptor`
	および
	`weblogic-enterprise-bean stateful-session-descriptor`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`lifecycle` 要素では、WebLogic Server の内部での、ステートフル EJB およびエンティティ EJB のライフサイクルに影響するオプションを定義します。現時点では、`lifecycle` 要素の子要素は `passivation-strategy` の 1 つだけです。

max-beans-in-cache

値の範囲	1 から *maxBeans* までの範囲内
デフォルト値	100
要件	省略可能な要素。
親要素	weblogic-enterprise-bean, entity-descriptor, entity-cache および weblogic-enterprise-bean stateful-session-descriptor stateful-session-cache
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

max-beans-in-cache 要素では、このクラスのオブジェクトを最大何個までメモリに展開できるかを指定します。オブジェクトの数が max-bean-in-cache に達すると、WebLogic Server はクライアントによって最近使われていない EJB の一部をパッシブ化します。max-beans-in-cache の値は、EJB が WebLogic Server キャッシュから削除されるタイミングにも影響します。

max-beans-in-free-pool

値の範囲	0 から *maxBeans* までの範囲内
デフォルト値	*max Int*
要件	省略可能な要素。ステートレス セッション EJB に対してのみ有効。
親要素	weblogic-enterprise-bean, transaction-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

WebLogic Server は、すべてのステートレス セッション Bean クラスおよびメッセージ駆動型 Bean クラスについて、EJB のフリー プールを管理します。max-beans-in-free-pool 要素では、このプールのサイズを定義します。デフォルトでは、max-beans-in-free-pool の制限は設定されていません。フリー プール内の Bean の数は、利用可能なメモリ容量だけに制限されます。

message-driven-descriptor

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	
親要素	`weblogic-enterprise-bean`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`message-driven-descriptor` 要素は、WebLogic Server における JMS の送り先にメッセージ駆動型 Bean を関連付けます。この要素では、以下のデプロイメント パラメータを指定します。

- pool
- destination-jndi-name
- initial-context-factory
- provider-url
- connection-factory-jndi-name

method

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。複数の EJB メソッドをコンフィグレーションするために、複数の `method` 要素を指定可能。
親要素	`weblogic-enterprise-bean` `transaction-isolation`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`method` 要素では、エンタープライズ Bean のホーム インターフェースまたはリモート インターフェースのメソッド、またはメソッドの集合を定義します。

method-intf

値の範囲	Home | Remote
デフォルト値	なし
要件	省略可能な要素。
親要素	weblogic-enterprise-bean transaction-isolation method
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

method-intf 要素では、WebLogic Server が分離レベル プロパティを適用する EJB インターフェースを指定します。この要素は、EJB のホーム インターフェースとリモート インターフェースでシグネチャが一致するメソッドを区別する必要がある場合にのみ使用します。

method-name

値の範囲	ejb-jar.xml ｜ * に定義された EJB の名前
デフォルト値	なし
要件	method スタンザ内の必須要素。
親要素	weblogic-enterprise-bean transaction-isolation method
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

method-name 要素では、WebLogic Server が分離レベル プロパティを適用する個別の EJB メソッドの名前を指定します。EJB のホーム インターフェースおよびリモート インターフェースの全メソッドを指定するには、アスタリスク（*）を使用します。

method-name を指定する場合、指定された ejb-name でメソッドが利用可能でなければなりません。

method-param

値の範囲	メソッド パラメータの Java 型名（完全修飾形式）
デフォルト値	なし
要件	method-params 内の必須要素。
親要素	weblogic-enterprise-bean transaction-isolation method method-params
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

method-param 要素では、メソッド パラメータの Java 型名を完全修飾形式で指定します。

method-params

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能なスタンザ。
親要素	weblogic-enterprise-bean transaction-isolation method
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

method-params スタンザは、個々のメソッド パラメータの Java 型名を定義する 1 つ以上の要素で構成されます。

passivation-strategy

値の範囲	`default	transaction`
デフォルト値	`default`
要件	省略可能な要素。エンティティ EJB に対してのみ有効。
親要素	`weblogic-enterprise-bean, entity-descriptor, lifecycle`
デプロイメントファイル	`weblogic-ejb-jar.xml`

機能

`passivation-strategy` 要素では、WebLogic Server キャッシュ内のエンティティ EJB の中間状態を保持するかどうかを指定します。

persistence

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	コンテナ管理による永続性サービスを利用するエンティティ EJB に対してのみ必要。
親要素	`weblogic-enterprise-bean`, `entity-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`persistence` 要素では、以下のオプションを定義します。これらのオプションは、WebLogic Server で使用するエンティティ EJB の永続性タイプ、トランザクション コミット時の動作、`ejbLoad()` および `ejbStore()` の動作を決定します。

- is-modified-method-name
- delay-updates-until-end-of-tx
- finders-load-bean
- persistence-type
- db-is-shared
- persistence-use

persistence-type

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	コンテナ管理による永続性サービスを利用するエンティティ EJB に対してのみ必要。
親要素	`weblogic-enterprise-bean`, `entity-descriptor`, `persistence`
デプロイメントファイル	`weblogic-ejb-jar.xml`

機能

`persistence-type` 要素では、エンティティ EJB で利用できる永続性サービスを定義します。`weblogic-ejb-jar.xml` ファイルには、複数の永続性サービスを使って EJB をテストする目的で、複数の `persistence-type` スタンザを定義することができます。デプロイメント時には、`persistence-use` に定義される永続性タイプだけが実際に使われます。

`persistence-type` 要素には、永続性タイプを識別する以下の子要素が含まれます。

- type-identifier
- type-version
- type-storage

persistence-use

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	コンテナ管理による永続性サービスを利用するエンティティ EJB に対してのみ必要。
親要素	`weblogic-enterprise-bean`, `entity-descriptor`, `persistence`
デプロイメントファイル	`weblogic-ejb-jar.xml`

機能

`persistence-use` 要素は `persistence-type` に似ていますが、デプロイメント時に実際に使われる永続性サービスを定義する点が異なります。`persistence-use` では、`persistence-type` の構造内で定義される `type-identifier` および `type-version` の各要素の値によってサービスを識別します。

persistent-store-dir

値の範囲	完全修飾形式のファイル システム パス
デフォルト値	なし
要件	省略可能な要素。
親要素	weblogic-enterprise-bean stateful-session-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

persistent-store-dir 要素では、パッシブ化されたステートフル セッション Bean インスタンスの状態を WebLogic Server が格納する、ファイル システム内のディレクトリを指定します。

pool

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	`weblogic-enterprise-bean` `stateless-session-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`pool` 要素では、ステートレス セッション EJB およびメッセージ駆動型 EJB について、WebLogic Server のフリー プールの動作をコンフィグレーションします。以下のオプションを指定できます。

- max-beans-in-free-pool
- initial-beans-in-free-pool

principal-name

値の範囲	有効な WebLogic Server プリンシパル名
デフォルト値	なし
要件	security-role-assignment スタンザで、少なくとも 1 つの principal-name の定義が必要である。各 role-name に対して、複数の principal-name を定義できる。
親要素	weblogic-enterprise-bean security-role-assignment
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

principal-name 要素では、指定された role-name に適用される、実際の WebLogic Server プリンシパルの名前を指定します。

provider-url

値の範囲	有効な名前
デフォルト値	なし
要件	`initial-context-factory` および `connection-factory-jndi-name` とともに使用する。
親要素	`weblogic-enterprise-bean` `message-driven-descriptor`
デプロイメントファイル	`weblogic-ejb-jar.xml`

機能

`provider-url` 要素では、InitialContext によって使われる URL プロバイダを指定します。通常、これは host : port の形式です。

read-timeout-seconds

値の範囲	0 から *maxSeconds* までの範囲内
デフォルト値	600
要件	省略可能な要素。エンティティ EJB に対してのみ有効。
親要素	weblogic-enterprise-bean, entity-descriptor, entity-cache
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

read-timeout-seconds 要素では、読み取り専用のエンティティ Bean に対する ejbLoad() の呼び出しの間の秒数を指定します。デフォルトでは、read-timeout-seconds は 0 に設定されており、WebLogic Server は Bean がキャッシュに移動されるときにのみ ejbLoad() を呼び出します。

reference-descriptor

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	`weblogic-enterprise-bean`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`reference-descriptor` 要素は、`ejb-jar.xml` ファイルに定義された参照を、WebLogic Server で利用可能な実際のリソース ファクトリおよび EJB の JNDI 名にマッピングします。

replication-type

値の範囲	InMemory | None
デフォルト値	なし
要件	省略可能な要素。クラスタ内のステートフル セッション EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean`
	`stateful-session-descriptor`
	`stateful-session-clustering`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`replication-type` 要素では、クラスタを構成する複数の WebLogic Server インスタンスの間で、ステートフル セッション EJB の状態をレプリケートするかどうかを指定します。この要素の値として `InMemory` を選択すると、EJB の状態はレプリケートされます。`None` を選択すると、状態はレプリケートされません。

res-env-ref-name

値の範囲	ejb-jar.xml ファイルに定義された有効なリソース環境参照名
デフォルト値	なし
要件	なし
親要素	weblogic-enterprise-bean reference-descriptor resource-env-description
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

res-env-ref-name 要素では、リソース環境参照の名前を指定します。

res-ref-name

値の範囲	ejb-jar.xml ファイルに定義された有効なリソース参照名
デフォルト値	なし
要件	ejb-jar.xml 内のリソース参照を EJB で指定する場合に必須の要素。
親要素	weblogic-enterprise-bean reference-description resource-description
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

res-ref-name 要素では、リソース ファクトリ参照の名前を指定します。これは、EJB プロバイダがデプロイメント ファイル ejb-jar.xml の内部で定義する参照です。

resource-env-description

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	`weblogic-enterprise-bean` `reference-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`resource-env-description` 要素は、`ejb-jar.xml` ファイルに定義されたリソース環境参照を、WebLogic Server で利用可能な実際のリソースの JNDI 名にマッピングします。

resource-description

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	`weblogic-enterprise-bean` `reference-description`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`resource-description` 要素は、`ejb-jar.xml` ファイルに定義されたリソース参照を、WebLogic Server で利用可能な実際のリソースの JNDI 名にマッピングします。

role-name

値の範囲	ejb-jar.xml に定義された EJB ロール名
デフォルト値	なし
要件	security-role-assignment 内で必須の要素。
親要素	weblogic-enterprise-bean security-role-assignment
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

role-name 要素は、EJB プロバイダがデプロイメント ファイル ejb-jar.xml に定義するアプリケーション ロール名を識別します。スタンザ内でこの要素に続く principal-name 要素は、この要素で指定された role-name に WebLogic Server プリンシパルをマッピングします。

run-as-identity-principal

値の範囲	ejb-jar.xml に定義されるアイデンティティとして使われるプリンシパル。
デフォルト値	なし
要件	security-role-assignment 内で必須の要素。
親要素	weblogic-enterprise-bean
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

run-as-identity-principal 要素では、ejb-jar.xml ファイルで security-identity.run-as-specified-identity が設定された Bean のアイデンティティとして使われるプリンシパルを指定します。

この要素に設定するプリンシパル名は、run-as-specified--identity ロールにマッピングされるプリンシパルのいずれかでなければなりません。

security-role-assignment

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	ejb-jar.xml にアプリケーション ロールが定義される場合に必須の要素。
親要素	weblogic-ejb-jar
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

security-role-assignment スタンザは、ejb-jar.xml ファイルに定義されたアプリケーション ロールを、WebLogic Server で利用可能なセキュリティ プリンシパルの名前にマッピングします。

stateful-session-cache

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	stateful-session-cache スタンザは省略可能で、ステートフル セッション EJB に対してのみ有効である。
親要素	weblogic-enterprise-bean, stateful-session-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

stateful-session-cache 要素では、WebLogic Server 内部のステートフル セッション EJB のインスタンスをキャッシュするために使われる、以下のオプションを定義します。

- home-is-clusterable
- home-load-algorithm
- home-call-router-class-name
- replication-type

stateful-session-clustering

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。クラスタ内のステートフル セッション EJB に対してのみ有効。
親要素	`weblogic-enterprise-bean`, `stateful-session-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`stateful-session-clustering` スタンザ要素では、クラスタ内のステートフル セッション EJB のインスタンスをレプリケートするときの、WebLogic Server の動作を決定する以下のオプションを指定します。

- max-beans-in-cache
- idle-timeout-seconds
- cache-type

stateful-session-descriptor

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	.jar 内のステートフル セッション EJB のそれぞれに対して、1 つの stateful-session-descriptor スタンザが必要である。
親要素	weblogic-enterprise-bean
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

stateful-session-descriptor 要素では、WebLogic Server 内のステートフル セッション EJB に適用される以下のデプロイメント パラメータを指定します。

- stateful-session-cache
- lifecycle
- persistent-store-dir
- stateful-session-clustering
- allow-concurrent-calls

stateless-bean-call-router-class-name

値の範囲	有効なルータ クラス名
デフォルト値	なし
要件	省略可能な要素。クラスタ内のステートレス セッション EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean`,
	`stateless-session-descriptor`
	`stateless-clustering`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`stateless-bean-call-router-class-name` 要素では、Bean のメソッド呼び出しをルーティングするために使用するカスタム クラスの名前を指定します。このクラスは `weblogic.rmi.extensions.CallRouter()` を実装しなければなりません。この要素の値を指定した場合、各メソッド呼び出しの前にこのクラスのインスタンスが呼び出されます。このとき、ルータ クラスでメソッド パラメータに基づいてルーティング先のサーバを選択する機会が設けられます。ルータ クラスはサーバ名または NULL のどちらかを返します。NULL が返された場合、サーバの選択は現在のロード アルゴリズムに基づいて行われます。

stateless-bean-is-clusterable

値の範囲	`true	false`
デフォルト値	`true`
要件	省略可能な要素。クラスタ内のステートレス セッション EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean,`
	`stateless-session-descriptor`
	`stateless-clustering`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`stateless-bean-is-clusterable` 要素の値が `true` のとき、クラスタ内の複数の WebLogic Server から EJB をデプロイすることができます。ホーム スタブへの呼び出しは、この Bean がデプロイされるサーバ間でロードバランシングされ、Bean が動作しているサーバに到達できない場合、呼び出しは自動的に、Bean が動作している別のサーバへとフェイルオーバされます。

stateless-bean-load-algorithm

値の範囲	round-robin | random | weight-based
デフォルト値	weblogic.cluster.defaultLoadAlgorithm の値
要件	省略可能な要素。クラスタ内のステートレス セッション EJB に対してのみ有効。
親要素	weblogic-enterprise-bean, stateless-session-descriptor stateless-clustering
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

stateless-bean-load-algorithm 要素では、EJB ホームのレプリカ間でロードバランシングを行うために使用するアルゴリズムを指定します。このプロパティを定義しない場合、WebLogic Server はサーバ プロパティ weblogic.cluster.defaultLoadAlgorithm によって指定されるアルゴリズムを使用します。

stateless-bean-load-algorithm には、以下のいずれかの値を設定できます。

- round-robin: Bean が動作しているサーバ間で、逐次的な方式でロードバランシングが行われます。
- random: EJB ホームのレプリカは、Bean が動作しているサーバを無作為に選択してデプロイされます。
- weight-based: EJB ホームのレプリカは、その時点での各サーバの作業負荷に応じてホスト サーバにデプロイされます。

stateless-bean-methods-are-idempotent

値の範囲	`true	false`
デフォルト値	`false`
要件	省略可能な要素。クラスタ内のステートレス セッション EJB に対してのみ有効。

親要素	`weblogic-enterprise-bean,`
	`stateless-session-descriptor`
	`stateless-clustering`

デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

この要素は `true` または `false` のどちらかに設定できます。同じ引数を使って同じメソッドを繰り返し呼び出すときの結果が、メソッドを 1 回呼び出すときとまったく同じになるように Bean が記述されている場合にのみ、`stateless-bean-methods-are-idempotent` を true に設定します。このように設定すると、フェイルオーバ処理で、障害を起こしたサーバ上で呼び出しが実際に完了したかどうかを調べる必要なしに、失敗した呼び出しを再試行することができます。このプロパティを `true` に設定すると、Bean が動作している別のサーバが到達可能である限り、どのような障害からも Bean スタブを自動的に回復できます。

stateless-clustering

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。クラスタ内のステートレス セッション EJB に対してのみ有効。
親要素	`weblogic-enterprise-bean`, `stateless-session-descriptor`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`stateless-clustering` 要素では、クラスタ内のステートレス セッション EJB のインスタンスをレプリケートするときの、WebLogic Server の動作を決定する以下のオプションを指定します。

- stateless-bean-is-clusterable
- stateless-bean-load-algorithm
- stateless-bean-call-router-class-name
- stateless-bean-methods-are-idempotent

stateless-session-descriptor

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	*.jar* 内のステートレス セッション EJB のそれぞれに対して、1 つの `stateless-session-descriptor` 要素が必要である。
親要素	`weblogic-enterprise-bean`
デプロイメント ファイル	`weblogic-ejb-jar.xml`

機能

`stateless-session-descriptor` 要素では、WebLogic Server 内のステートレス セッション EJB のキャッシング、クラスタリング、永続性などに関係するデプロイメント パラメータを定義します。

transaction-descriptor

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	weblogic-enterprise-bean
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

transaction-descriptor 要素では、WebLogic Server でのトランザクション動作を定義するオプションを指定します。現時点で、このスタンザに含まれる要素は trans-timeout-seconds だけです。

transaction-isolation

値の範囲	なし（XML スタンザ）
デフォルト値	なし（XML スタンザ）
要件	省略可能な要素。
親要素	transaction-description
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

transaction-isolation 要素では、EJB に関してメソッドレベルでのトランザクション分離設定を定義します。

trans-timeout-seconds

値の範囲	0 から *maxSeconds* までの範囲内
デフォルト値	30
要件	省略可能な要素。すべての EJB に対して有効。
親要素	weblogic-enterprise-bean, transaction-descriptor
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

trans-timeout-seconds 要素では、EJB のコンテナによって開始されるトランザクションについて、最長の継続期間を指定します。トランザクションが trans-timeout-seconds の秒数を超過すると、WebLogic Server はトランザクションをロールバックします。

trans-timeout-seconds の値を指定しない場合、コンテナによって開始されるトランザクションは、デフォルトで 5 分後にタイムアウトします。

type-identifier

値の範囲	有効な文字列
デフォルト値	なし
要件	コンテナ管理による永続性サービスを利用するエンティティ EJB に対してのみ必要。
親要素	weblogic-enterprise-bean, entity-descriptor, persistence persistence-type および weblogic-enterprise-bean, entity-descriptor, persistence persistence-use
デプロイメントファイル	weblogic-ejb-jar.xml

機能

type-identifier 要素には、エンティティ EJB の永続性タイプを識別するテキストを設定します。WebLogic Server の RDBMS ベースの永続性では、識別子 WebLogic_CMP_RDBMS が使われます。BEA 以外のベンダが提供する永続性メカニズムを利用する場合、type-identifier の有効な値については、各ベンダが提供するドキュメントを参照してください。

type-storage

値の範囲	有効な文字列
デフォルト値	なし
要件	コンテナ管理による永続性サービスを利用するエンティティ EJB に対してのみ必要。
親要素	weblogic-enterprise-bean, entity-descriptor, persistence persistence-type
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

type-storage 要素には、この永続性タイプのデータが格納されるファイルの絶対パスを定義します。このパスには、EJB のデプロイメント ファイル（.jar）またはデプロイメント ディレクトリの最上位を起点としたファイル位置を指定しなければなりません。

WebLogic Server の RDBMS ベースの永続性では一般に、weblogic-cmp-rdbms-jar.xml という名前の XML ファイルを使用して、Bean の永続性データを格納します。このファイルは、.jar ファイルの META-INF サブディレクトリに格納されます。

type-version

値の範囲	有効な文字列
デフォルト値	なし
要件	コンテナ管理による永続性サービスを利用するエンティティ EJB に対してのみ必要。

親要素	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-type`
	および
	`weblogic-enterprise-bean,`
	`entity-descriptor,`
	`persistence`
	`persistence-use`

デプロイメントファイル	`weblogic-ejb-jar.xml`

機能

`type-version` 要素は、指定された永続性タイプのバージョンを識別します。

注意： WebLogic Server の RDBMS ベースの永続性を利用する場合、指定するバージョンは、WebLogic Server のリリースに対応する RDBMS 永続性バージョンと**厳密**に一致しなければなりません。無効なバージョンを指定すると、以下のエラーが発生します。

```
weblogic.ejb.persistence.PersistenceSetupException: Error
initializing the CMP Persistence Type for your bean: No installed
Persistence Type matches the signature of (identifier
'Weblogic_CMP_RDBMS', version 'version_number').
```

weblogic-ejb-jar

値の範囲	
デフォルト値	
要件	
親要素	weblogic-enterprise-bean,
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

weblogic-ejb-jar は、EJB デプロイメント記述子の weblogic コンポーネントのルート要素です。

weblogic-enterprise-bean

値の範囲	
デフォルト値	
要件	
親要素	weblogic-enterprise-bean,
デプロイメント ファイル	weblogic-ejb-jar.xml

機能

weblogic-enterprise-bean 要素には、WebLogic Server で利用可能な Bean のデプロイメント情報が定義されます。

この章では、バージョン 6.1 の WebLogic Server に固有の XML 要素について説明します。これらの要素は weblogic510-ejb-jar.xml ファイルを構成し、WebLogic Server EJB コンテナ内のデプロイメント記述子を定義するために使われるものです。これらの要素は、WebLogic Server Administration Console のほぼ同じ名前のフィールドに対応します。これらのデプロイメント記述子は、バージョン 1.1 の EJB に対して使用します。

caching-descriptor

caching-descriptor スタンザでは、WebLogic Server キャッシュ内の EJB の数と、EJB がパッシブ化されるかまたはプールに格納されるまでの時間を指定します。スタンザ全体は、スタンザ内の各要素と同様に省略可能です。どの要素も定義しない場合、WebLogic Server はデフォルト値を使用します。

max-beans-in-free-pool

注意： この要素は、ステートレス セッション EJB に対してのみ有効です。

WebLogic Server は、すべての Bean クラスについて EJB のフリー プールを管理します。この省略可能の要素では、プールのサイズを定義します。デフォルトでは、max-beans-in-free-pool の制限は設定されていません。フリー プール内の Bean の数は、利用可能なメモリ容量だけに制限されます。

initial-beans-in-free-pool

注意： この要素は、ステートレス セッション EJB に対してのみ有効です。

initial-bean-in-free-pool の値を指定すると、WebLogic Server の起動時に、指定された数の Bean インスタンスがプールに格納されます。このようにフリープールを満たしておくことにより、新しいインスタンスを生成しなくても Bean に対しての初期リクエストを処理できるため、EJB の初期応答時間が改善されます。

initial-bean-in-free-pool 要素の値を定義しない場合、デフォルト値の 0 が使われます。

max-beans-in-cache

注意： この要素は、ステートフル セッション EJB およびエンティティ EJB に対してのみ有効です。

この要素では、このクラスのオブジェクトをメモリ上に何個まで展開できるかを指定します。オブジェクトの数が max-bean-in-cache に達すると、WebLogic Server はクライアントによって最近使われていない EJB の一部をパッシブ化します。max-beans-in-cache の値は、EJB が WebLogic Server キャッシュから削除されるタイミングにも影響します。

max-beans-in-cache のデフォルト値は 100 です。

idle-timeout-seconds

idle-timeout-seconds 要素では、ステートフル EJB をキャッシュに保持する最長の時間を定義します。この時間が経過した後で、キャッシュ内の Bean の数が max-beans-in-cache の制限に達すると、WebLogic Server は Bean インスタンスを削除する場合があります。

要素の値を定義しない場合、idle-timeout-seconds のデフォルト値は 600 です。

cache-strategy

cache-strategy 要素には、以下のいずれかの値を設定できます。

- Read-Write
- Read-Only

デフォルト値は Read-Write です。

read-timeout-seconds

read-timeout-seconds 要素では、読み取り専用のエンティティ Bean に対する ejbLoad() の呼び出しの間の秒数を指定します。デフォルトでは、read-timeout-seconds は 600 秒に設定されます。この値を 0 に設定すると、WebLogic Server は Bean がキャッシュに移動されるときにのみ ejbLoad を呼び出します。

persistence-descriptor

persistence-descriptor スタンザでは、エンティティ EJB の永続性オプションを指定します。

is-modified-method-name

is-modified-method-name 要素では、EJB が格納されるときに WebLogic Server が呼び出すメソッドを指定します。指定されたメソッドは、boolean 型の値を返す必要があります。メソッドを指定しない場合、WebLogic Server は常に、EJB に変更が加えられたことを想定して EJB を保存します。

メソッドを定義して適切に設定することにより、パフォーマンスが向上する可能性があります。ただし、メソッドの戻り値に誤りがあると、データの不整合に関する問題が生じる可能性があります。

delay-updates-until-end-of-tx

トランザクションの完了時に、トランザクションを構成する全 Bean の永続ストアを更新するには、このプロパティを true（デフォルト）に設定します。このように設定すると、不必要な更新が回避されるためパフォーマンスが向上するのが一般的です。ただしこの設定では、データベース トランザクション内部でのデータベース更新の順序は保持されません。

データストアで分離レベル TRANSACTION_READ_UNCOMMITTED を使用する場合に、ほかのデータベース ユーザが、進行中のトランザクションの中間結果を確認できるようにすることができます。この場合、各メソッド呼び出しの完了時に Bean の永続ストアが更新されるように、delay-updates-until-end-of-tx の値を false に設定します。

注意： delay-updates-until-end-of-tx を false に設定すると、各メソッドの呼び出し後にデータベースの更新はデータベースに送られるだけで、コミットはされません。更新はトランザクションが完了した時点でデータベースにコミットされるか、またはロールバックされます。

persistence-type

persistence-type には、EJB で利用可能な永続性サービスを定義します。複数の永続性サービスを使ったテストを行う目的で、weblogic-ejb-jar.xml ファイルに複数の persistence-type エントリを定義することができます。デプロイメント時には、persistence-use に定義された永続性タイプだけが使われます。

persistence-type 要素は、サービスのプロパティを定義する以下の子要素で構成されます。

- type-identifier 要素には、指定された永続性タイプを識別するテキストを指定します。たとえば、WebLogic Server の RDBMS 永続性は識別子 WebLogic_CMP_RDBMS を使用します。
- type-version 要素は、指定された永続性タイプのバージョンを識別します。

注意： 指定するバージョンは、WebLogic Server のリリースに対応する RDBMS 永続性のバージョンと**厳密**に一致しなければなりません。無効なバージョンを指定すると、エラーが発生します。

```
weblogic.ejb.persistence.PersistenceSetupException: Error
initializing the CMP Persistence Type for your bean: No installed
Persistence Type matches the signature of (identifier
'Weblogic_CMP_RDBMS', version 'version_number').
```

- type-storage 要素には、この永続性タイプのデータが格納されるファイルの絶対パスを定義します。このパスには、EJB のデプロイメント ファイル（*.jar*）またはデプロイメント ディレクトリの最上位を起点としたファイル位置を指定しなければなりません。

  WebLogic Server の RDBMS ベースの永続性では一般に、weblogic-cmp-rdbms-jar.xml という名前の XML ファイルを使用して、Bean の永続性データを格納します。このファイルは、*.jar* ファイルの META-INF サブディレクトリに格納されます。

db-is-shared

db-is-shared 要素は、エンティティ Bean だけに適用されます。この要素の値を true（デフォルト値）に設定すると、WebLogic Server は EJB データをトランザクションの間に変更できないことを想定し、各トランザクションの開始時にデータを再ロードします。false に設定すると、WebLogic Server は永続ストア内の EJB データに排他的にアクセスできることを想定します。

stateful-session-persistent-store-dir

stateful-session-persistent-store-dir 要素では、パッシブ化されたステートフル セッション Bean インスタンスの状態が格納される、ファイル システム内のディレクトリを指定します。

persistence-use

persistence-use プロパティは persistence-type に似ていますが、デプロイメント時に実際に使われる永続性サービスを定義する点が異なります。persistence-use では、persistence-type の構造内で定義される type-identifier および type-version の各要素の値によってサービスを識別します。

clustering-descriptor

clustering-descriptor スタンザでは、WebLogic Server クラスタにデプロイされる EJB に関して、レプリケーションのプロパティおよび動作を定義します。clustering-descriptor スタンザとその子要素は省略可能であり、シングル サーバ システムには適用されません。

home-is-clusterable

この要素は true または false のどちらかに設定できます。home-is-clusterable 要素の値が true のとき、クラスタ内の複数の WebLogic Server から EJB をデプロイすることができます。ホーム スタブへの呼び出しは、この Bean がデプロイされるサーバ間でロードバランシングされ、Bean が動作しているサーバに到達できない場合、呼び出しは自動的に、Bean が動作している別のサーバへとフェイルオーバされます。

home-load-algorithm

home-load-algorithm 要素では、EJB ホームのレプリカ間でロードバランシングを行うためのアルゴリズムを指定します。このプロパティを定義しない場合、WebLogic Server はサーバ プロパティ weblogic.cluster.defaultLoadAlgorithm によって指定されるアルゴリズムを使用します。

home-load-algorithm には、以下のいずれかの値を設定できます。

- round-robin: Bean が動作しているサーバ間で、逐次的な方式でロードバランシングが行われます。
- random: EJB ホームのレプリカは、Bean が動作しているサーバを無作為に選択してデプロイされます。
- weight-based: EJB ホームのレプリカは、その時点での各サーバの作業負荷に応じてホスト サーバにデプロイされます。

home-call-router-class-name

home-call-router-class-name 要素では、Bean のメソッド呼び出しをルーティングするために使用するカスタム クラスを指定します。このクラスは weblogic.rmi.extensions.CallRouter() を実装しなければなりません。この要素の値を指定すると、各メソッド呼び出しの前にこのクラスのインスタンスが呼び出されます。このとき、ルータ クラスでメソッド パラメータに基づいてルーティング先のサーバを選択する機会が設けられます。ルータ クラスはサーバ名または NULL のどちらかを返します。NULL が返された場合、サーバの選択は現在のロード アルゴリズムに基づいて行われます。

stateless-bean-is-clusterable

このプロパティは home-is-clusterable に似ていますが、ステートレス セッション EJB だけに適用される点が異なります。

stateless-bean-load-algorithm

このプロパティは home-load-algorithm に似ていますが、ステートレス セッション EJB だけに適用される点が異なります。

stateless-bean-call-router-class-name

このプロパティは home-call-router-class-name に似ていますが、ステートレス セッション EJB だけに適用される点が異なります。

stateless-bean-methods-are-idempotent

この要素は true または false のどちらかに設定できます。同じ引数を使って同じメソッドを繰り返し呼び出すときの結果が、メソッドを 1 回呼び出すときとまったく同じになるように Bean が記述されている場合にのみ、stateless-bean-methods-are-idempotent を true に設定します。このように設定すると、フェイルオーバ処理で、障害を起こしたサーバ上で呼び出しが実際に完了したかどうかを調べる必要なしに、失敗した呼び出しを再試行することができます。このプロパティを true に設定すると、Bean が動作している別のサーバが到達可能である限り、どのような障害からも Bean スタブを自動的に回復できます。

注意： このプロパティは、ステートレス セッション EJB だけに適用されます。

transaction-descriptor

transaction-descriptor スタンザは、WebLogic Server でのトランザクションの動作を定義する子要素で構成されます。現時点で、このスタンザを構成する要素は次のとおりです。

trans-timeout-seconds

trans-timeout-seconds 要素では、EJB のコンテナによって開始されるトランザクションについて、最長の継続期間を指定します。トランザクションが trans-timeout-seconds の秒数を超過すると、WebLogic Server はトランザクションをロールバックします。

trans-timeout-seconds の値を指定しない場合、コンテナによって開始されるトランザクションは、デフォルトで 5 分後にタイムアウトします。

reference-descriptor

reference-descriptor スタンザは、ejb-jar.xml ファイルに定義された参照を、WebLogic Server で利用可能な実際のリソース ファクトリおよび EJB の JNDI 名にマッピングします。

reference-descriptor スタンザの内部にはさらに、リソース ファクトリ参照および EJB 参照を定義するための 1 つ以上のスタンザ構造が含まれます。

resource-description

以下の子要素で、個別の resource-description を定義します。

- res-ref-name には、リソース参照名を指定します。これは、EJB プロバイダがデプロイメント ファイル ejb-jar.xml の内部で定義する参照です。
- jndi-name には、WebLogic Server で利用可能な実際のリソース ファクトリの JNDI 名を指定します。

ejb-reference-description

以下の子要素で、個別の ejb-reference-description を定義します。

- res-ref-name には、EJB 参照名を指定します。これは、EJB プロバイダがデプロイメント ファイル ejb-jar.xml の内部で定義する参照です。
- jndi-name には、WebLogic Server で利用可能な実際の EJB の JNDI 名を指定します。

transaction-isolation

transaction-isolation スタンザでは、EJB メソッドについてトランザクションの分離レベルを指定します。このスタンザは、広い範囲の EJB メソッドに適用される、1 つ以上の isolation-level 要素で構成されます。

isolation-level

isolation-level は、特定の EJB メソッドに適用される、有効なトランザクション分離レベルを定義します。isolation-level には、以下の値を設定できます。

- TRANSACTION_READ_UNCOMMITTED: トランザクションで、未コミットの更新をほかのトランザクションから確認できます。
- TRANSACTION_READ_COMMITTED: トランザクションで、コミット済みの更新だけをほかのトランザクションから確認できます。
- TRANSACTION_REPEATABLE_READ: トランザクションがデータの一部をいったん読み取ると、同じデータが繰り返し読み取られたときに、ほかのトランザクションによって読み取り後にデータが変更された場合でも、最初に読み取られた同じ値が返されます。
- TRANSACTION_SERIALIZABLE: このトランザクションを複数回にわたって同時実行するときと、シリアル方式で複数回実行するときの効果が同じです。

それぞれの分離レベルの効用およびサポート状況の詳細については、使用しているデータベースのドキュメントを参照してください。

method

method スタンザは、分離レベルが適用される EJB メソッドを定義します。method では以下の子要素によって、さまざまな種類のメソッドを定義します。

- description はメソッドについての説明で、省略可能な要素です。
- ejb-name は、WebLogic Server が分離レベル プロパティを適用する EJB を識別します。
- method-intf は省略可能な要素で、指定されたメソッドが EJB のホーム インターフェースとリモート インターフェースのどちらに位置するかを示します。この要素の値は「Home」または「Remote」のどちらかでなければなりません。method-intf を指定しない場合、両方のインターフェースのメソッドに分離を適用できます。
- method-name には EJB メソッドの名前を指定します。アスタリスク（*）を使って、すべての EJB メソッドを指定することもできます。

- method-params は省略可能なスタンザで、メソッドの各パラメータの Java 型を列挙します。各パラメータの型は、method-params スタンザを構成する個別の method-param 要素を使用して、順番に列挙する必要があります。

security-role-assignment

security-role-assignment スタンザは、ejb-jar.xml ファイルに定義されたアプリケーション ロールを、WebLogic Server で利用可能なセキュリティ プリンシパルの名前にマッピングします。-

security-role-assignment には、以下の要素ペアを 1 つ以上定義できます。

- role-name は、EJB プロバイダがデプロイメント ファイル ejb-jar.xml に定義するアプリケーション ロール名です。
- principal-name には、実際の WebLogic Server プリンシパルの名前を指定します。

enable-call-by-reference

デフォルトでは、同じサーバの内部から呼び出される EJB メソッドでは、参照によって引数を渡します。この方式ではパラメータがコピーされないため、メソッド呼び出しのパフォーマンスが向上します。

enable-call-by-reference 要素の値を false に設定すると、EJB メソッドへのパラメータは、EJB 1.1 仕様に準拠する形式（値渡し）でコピーされます。EJB がサーバの内部からではなくリモートで呼び出されるときは常に、値渡しが必要です。

25 実行時 EJB ホーム

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- アプリケーション
- 名前
- URI

26 実行時エンティティ EJB

以下の統計によって、エンティティ EJB をモニタすることができます。

統計	説明
Idle Beans Count	プール内にある使用可能なアイドル中の Bean の数。
Beans In Use Count	現在のセッションで使用中 (Active/Ready 状態など) の Bean の数。 beans-in-use の値が beans-in-cache の値よりも若干大きくなる場合もある。ファインダー メソッドやホーム メソッドを実行するときには匿名インスタンスを使用するが、匿名インスタンスを使用すると、beans-in-cache の値は変わらず、beans-in-use の値が増える。ただし、その差はそれほど大きくはならない。
Waiter Total Count	スレッドがプール内の Bean を要求して待機した回数。
Timeout Total Count	タイムアウトになったトランザクションの総数。
Cached Beans Current Count	現在キャッシュされている Bean の数。
Cache Access Count	キャッシュへのアクセス回数。
Cache Hit Count	検索された Bean がキャッシュ内で見つかった回数。
Activation Count	アクティブにされた Bean の数。
Passivation Count	パッシブにされた Bean の数。
Lock Entries Current Count	現在ロックされているエントリの数。

統計	説明
Lock Manager Access Count	Lock Manager のアクセス回数。排他的ロックが指定されている Bean のみ該当。

27 実行キュー

新しい実行キューの作成

1. Administration Console の左ペインに表示されているサーバのリストから、実行キューを作成する対象のサーバを選択します。
2. [モニタ] タブをクリックします。
3. [一般] タブで、[すべてのアクティブなキューのモニタ ...] をクリックします。
4. [Execute Queues のコンフィグレーション ...] をクリックします。
5. [新しい Execute Queue のコンフィグレーション] をクリックします。
6. [コンフィグレーション] タブで、デフォルトの値をそのままにするか、あるいは値を編集します。
7. [作成] をクリックします。

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	実行キューの名前を設定する。		
[スレッド優先順位]	キューのスレッドの優先順位を設定する。		
[スレッド数]	キューの最大スレッド数を設定する。		

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定できる。	文字列	NULL

28 実行時実行キュー

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前
- アイドル状態の実行スレッドの数
- 最も古い保留状態のリクエストの日付
- 処理されたリクエストの数
- 保留状態のリクエストの数

29 実行スレッド

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- キュー
- 名前
- ユーザ
- トランザクション
- 総リクエスト数

30 FileT3

以下に、Administration Console を使用して、FileT3 のコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「WebLogic Server とクラスタのコンフィグレーション」を参照してください。

新しい FileT3 のコンフィグレーション

1. 左ペインの [FileT3] ノードをクリックします。ドメインで定義されているすべての FileT3 を示す [FileT3 コンフィグレーション] テーブルが右ペインに表示されます。
2. [新しい FileT3 のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する FileT3 の設定に関連するタブが示されます。
3. [名前] および [パス] 属性フィールドに値を入力します。
4. [作成] をクリックして、FileT3 のインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [FileT3] ノードの下に追加されます。
5. [適用] をクリックしてこれまでの変更を保存します。

FileT3 のクローンの作成

1. 左ペインの [FileT3] ノードをクリックします。ドメインで定義されているすべての FileT3 を示す [FileT3 コンフィグレーション] テーブルが右ペインに表示されます。

2. クローンを作成する FileT3 の行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、FileT3 のクローンの作成に関連するタブが示されます。
3. [名前] および [パス] 属性フィールドに値を入力します。
4. [クローン] をクリックして、FileT3 のインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [FileT3] ノードの下に追加されます。
5. [適用] をクリックして変更を保存します。

FileT3 の削除

1. 左ペインの [FileT3] ノードをクリックします。ドメインで定義されているすべての FileT3 を示す [FileT3 コンフィグレーション] テーブルが右ペインに表示されます。
2. 削除する FileT3 の行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして FileT3 を削除します。[FileT3] ノードの下の FileT3 アイコンが削除されます。

FileT3 の割り当て

1. 左ペインで、ファイルに割り当てる [FileT3] の下のインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. FileT3 に割り当てる 1 つまたは複数の対象を [選択可] カラムで選択します。

b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。

c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	FileT3 コンポーネントの名前を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	MyFileT3
[パス]	ユーザが FileT3 コンポーネントのパスを設定できるようにする。	有効なパス	NULL

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	FileT3 コンポーネントの対象として使用するサーバをユーザが選択できるようにする。	リスト	NULL

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	FileT3 コンポーネントの対象として使用するクラスタをユーザが選択できるようにする。	リスト	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

31 グループ

以下に、Administration Console を使用して、グループの作成や管理に必要な属性を設定する手順を説明します。グループについては、『管理者ガイド』の「セキュリティの管理」を参照してください。

新しいグループの作成

1. 左ペインの [グループ] ノードをクリックします。ドメイン内のすべてのグループを示す [グループ] テーブルが右ペインに表示されます。
2. [新しいグループ名] 属性フィールドに値を入力します。
3. [作成] をクリックしてグループを作成します。新しいグループを操作するためのコントロールを示すダイアログが右ペインに表示されます。
4. [ユーザ] および [グループ] 属性フィールドに値を入力して、ユーザとグループをそれぞれ追加します。各フィールドでは、複数のユーザおよび複数のグループの区切りにはスペースを使用します。追加するユーザおよびグループはデータベース内に存在している必要があります。
5. [選択したメンバを削除] をクリックします。右ペインのダイアログが更新され、[メンバ] フィールドに新規ユーザおよびグループが表示されます。

グループの削除

1. 左ペインの [グループ] ノードをクリックします。ドメイン内のすべてのグループを示す [グループ] テーブルが右ペインに表示されます。
2. 削除するグループの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。

3. [はい] をクリックしてグループを削除します。[グループ] ノードの下のグループアイコンが削除されます。

グループからのユーザの削除

1. 左ペインの [グループ] ノードをクリックします。ドメイン内のすべてのグループを示す [グループ] テーブルが右ペインに表示されます。
2. 削除するユーザを持つグループの名前をクリックします。選択したグループを操作するためのコントロールを示すダイアログが右ペインに表示されます。
3. 削除するユーザの名前を選択します。
4. [選択したメンバを削除] をクリックします。右ペインのダイアログが更新され、選択したユーザが [メンバ] フィールドから削除されます。

グループからのグループの削除

1. 左ペインの [グループ] ノードをクリックします。ドメイン内のすべてのグループを示す [グループ] テーブルが右ペインに表示されます。
2. 削除するグループを持つグループの名前をクリックします。選択したグループを操作するためのコントロールを示すダイアログが右ペインに表示されます。
3. 削除するグループの名前を選択します。
4. [選択したメンバを削除] をクリックします。右ペインのダイアログが更新され、選択したグループが [メンバ] フィールドから削除されます。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	グループの名前を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[メンバ]	グループ内のメンバを返す。		NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

グループについては、『管理者ガイド』の「セキュリティの管理」を参照してください。

32 JDBC 接続プール

以下の手順では、Administration Console を利用して接続プールのコンフィグレーションおよび管理用の属性を設定する方法について説明します。接続プールはデータベース接続性の基本部分を成すものです。詳細については、下記のマニュアルを参照してください。

- 『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」
- 『WebLogic JDBC プログラミング ガイド』

JDBC 接続プールのコンフィグレーション

1. [JDBC] ノードをクリックして展開します。
2. [接続プール] ノードをクリックします。右ペインに [JDBC 接続プール] テーブルが表示され、ドメインに定義されているすべての接続プールが示されます。
3. [新しい JDBC Connection Pool のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する接続プールのコンフィグレーションに関連するタブが示されます。
4. [名前]、[URL]、[ドライバ クラス名]、[プロパティ]、[パスワード]、および [オープン文字列のパスワード] の各属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドに指定した名前の接続プール インスタンスを作成します。新規インスタンスが左ペインの [接続プール] ノードの下に追加されます。
6. [接続] タブおよび [テスト] タブをクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
7. [適用] をクリックして、変更を保存します。

データベース接続性の設定で次に行うべき作業については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

JDBC 接続プールのクローンの作成

1. [JDBC] ノードをクリックして展開します。
2. [接続プール] ノードをクリックします。右ペインに [JDBC 接続プール] テーブルが表示され、ドメインに定義されているすべての接続プールが示されます。
3. クローンを作成したい接続プールの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、接続プールのクローン作成に関連するタブが示されます。
4. [名前]、[URL]、[ドライバクラス名]、[プロパティ] 、および [パスワード] の各属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドに指定した名前の接続プール インスタンスを作成します。新しいインスタンスが左ペインの [接続プール] ノードの下に追加されます。
6. [接続] タブおよび [テスト] タブをクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
7. [適用] をクリックして、変更を保存します。

JDBC 接続プールの削除

1. [JDBC] ノードをクリックして展開します。
2. [接続プール] ノードをクリックします。右ペインに [JDBC 接続プール] テーブルが表示され、ドメインに定義されているすべての接続プールが示されます。
3. 削除したい接続プールの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。

4. [はい] をクリックして接続プールを削除します。[接続プール] ノードの下の接続プール アイコンが削除されます。

サーバ、クラスタへの JDBC 接続プールの割り当て

1. 左ペインで、[接続プール] の下のインスタンス ノードをクリックして、割り当て対象のプールを指定します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] タブ、あるいは [クラスタ] タブで次の手順を実行します。
   a. 接続プールの割り当て先となる対象を [選択可] カラムで任意の数だけ選択します。
   b. 移動コントロールをクリックして、選択した対象を [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

以下の表では、データベース接続性の基本部分を成す接続プールをコンフィグレーションおよび管理するために Administration Console で設定される属性について説明します。接続性の設定に必要な手順については、「JDBC 接続の管理」のコンフィグレーション手順および『WebLogic JDBC プログラミング ガイド』を参照してください。

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	接続プールの名前を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[URL]	データベースの URL（形式は JDBC ドライバのマニュアルで指定されているとおり）。この URL は、物理データベース接続を作成するために JDBC ドライバに渡される。		
[ドライバ クラス名]	JDBC ドライバ クラス名を表す文字列。これは、当該接続プール用に WebLogic Server と DBMS 間の物理接続を作成するのに使われる JDBC 2 層ドライバの完全パッケージ名である。`java.sql.Driver` インターフェースを実装するクラスの名前でなければならない。絶対パス名については、該当する JDBC ドライバのマニュアルを参照のこと。		

属性	説明	値の範囲	デフォルト値
[プロパティ]	この JDBC ドライバに渡されるプロパティのリスト。		
[パスワード]	この値は、（名前と値の組として）プロパティに定義されているあらゆるパスワードに優先する。この属性は、物理データベース接続の作成時に 2 層 JDBC ドライバに渡される。この値は暗号化形式で config.xml に格納されるので、これを使えば、そのファイルに平文のパスワードが保存されないようにすることができる。		NULL
[オープン文字列のパスワード]	この属性に設定した値は、オープン文字列のパスワードをオーバーライドする。 このパスワードは、XA の物理データベース接続を作成するためのオープン文字列で使用される。 値は、暗号化されて config.xml に格納される。		

[接続]

属性	説明	値の範囲	デフォルト値
[ログイン遅延時間]	各データベース接続を作成するまでにかかる遅延時間（秒数）。この遅延は、最初にプールが作成されるときにも、プールの生存期間中に物理データベース接続が作成されるときにも発生する。データベース サーバの中には、複数の接続要求が短時間に連続して発生すると処理できないものがある。このプロパティを使えば、短い遅延を組み込んでデータベース サーバがすべての要求を処理できるようにすることが可能。	整数（秒）	0
[初期容量]	プールのコンフィグレーション時に作成する物理データベース接続の数。この数の接続を作成できなければ、当該接続プールは作成できない。これは、プールにおいて使用可能な状態に保たれる最小の物理接続数でもある。	整数。最小値 = 0	1

属性	説明	値の範囲	デフォルト値
[最大容量]	当該接続プールが保有できる物理データベース接続の最大数。各種の JDBC ドライバとデータベース サーバにはおそらく、作成可能な物理接続の数に対する制限がある。	整数。可能な最小値 = 1	1
[増加容量]	プール容量を拡張する際の増加量。要求に応えるために使用できる物理接続がもうない場合には、接続プールは追加の物理データベース接続をこの数だけ作成し、プールに追加する。プールは、[最大容量] に設定されている物理接続の最大数を絶対に超過しないようにする。	整数。最小値 = 0	1
[縮小可]	True に設定すると、プールの縮小が有効になる。使用されていない接続が検出されたときにプールが［初期容量］に縮小できるかどうかを指定。	ブール 選択されている = true 選択されていない = false	true
[縮小間隔]	要求に合わせてインクリメンタルに容量が増大した接続プールを縮小するまでの待ち時間（分）を設定。縮小が行われるようにするには、ShrinkingEnabled を true に設定しておく必要がある。	整数（分単位）。最小値 = 1	15

属性	説明	値の範囲	デフォルト値
[更新間隔]	接続更新間隔を設定。未使用の接続はすべて、TestTableName を使ってテストされる。このテストに合格しない接続は閉じられ、有効な物理データベース接続の再確立を試みる際に再開される。TestTableName が設定されていなければ、テストは実行されない。	整数（分単位）。 最小値 = 0	0
［ローカルトランザクションのサポート］	XA 接続プールにのみ適用され、XA ドライバ以外の場合には無視される。XA ドライバがグローバル トランザクションのない SQL をサポートする場合には true に設定。	ブール 選択されている = true 選択されていない = false	選択されていない = false

[テスト]

属性	説明	値の範囲	デフォルト値
[テストテーブル名]	物理データベース接続のテスト時に使われるテーブルの名前を設定。接続のテストに使われるデフォルトの SQL は TestTableName からの「select count」。TestTableName が存在し、データベース ユーザからアクセスして接続可能でなければならない。大半のデータベース サーバでは、この SQL を最適化してテーブル スキャンを避けるが、それでも、行がほとんどあるいはまったくないことがわかっているテーブルの名前を TestTableName に設定したほうがよい。 TestTableName が「SQL」で始まる場合には、その先頭のトークンの後の残りの文字列は、接続のテストに使われるリテラル SQL 文とみなされる。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[リザーブされたときに接続をテスト]	リザーブされている接続をテストするかどうかの決定に使われる。true に設定された場合には、WebLogic Server はクライアントに接続を提供したあと、接続をテストする。このテストが行われると、クライアントがプールに接続を要求した場合、その要求に応えるまでに短い遅延が生じることになるが、クライアントはそのおかげで有効な接続を確実に受け取ることができるようになる（DBMS が利用可能およびアクセス可能であると仮定）。	ブール 選択されている = True 選択されていない = False	False
[リリースされたときに接続をテスト]	リリースされる接続をテストするかどうかを決定するために使われる。true に設定された場合には、WebLogic Server は接続をテストしてから接続プールに返す。プール内のすべての接続がすでに使用中で、かつクライアントが接続を待っている場合、接続のテスト中はクライアントの待ち時間が少しだけ長くなる。	ブール 選択されている = True 選択されていない = False	False

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	このデプロイメントの対象サーバを設定する。	文字列	デフォルトは [Lweblogic.management.configuration.TargetMBean;@2c84d9

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	このデプロイメントの対象クラスタを設定する。	文字列	デフォルトは [Lweblogic.management.configuration.TargetMBean;@2c84d9

[モニタ]

属性	説明	値の範囲	デフォルト値
[サーバでデプロイされました]	プールのデプロイ先としてコンフィグレーションするサーバの数を示す整数。	整数	
[Instances Currently Active]	当該プールに対して現在アクティブになっているサーバの数を示す整数。	整数	

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	文字列	NULL

詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」および『WebLogic JDBC プログラミング ガイド』を参照してください。

33 実行時 JDBC 接続プール

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- マシン
- 最大接続数
- 最大待ち時間 (秒)

34 JDBC データ ソース

以下の手順では、Administration Console を利用してデータソースのコンフィグレーションおよび管理用の属性を設定する方法について説明します。データソースの詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

JDBC データ ソースのコンフィグレーション

1. [JDBC] ノードをクリックして展開します。
2. [データ ソース] ノードをクリックします。右ペインに [JDBC データ ソース] テーブルが表示され、ドメインに定義されているすべてのデータソースが示されます。
3. [新しい JDBC Data Source のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するデータソースのコンフィグレーションに関連するタブが表示されます。
4. [名前]、[JNDI 名]、および [プール名] の各属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドに指定した名前のデータ ソース インスタンスを作成します。新しいインスタンスが左ペインの [データ ソース] ノードの下に追加されます。

データベース接続性の設定で次に行うべき作業については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

JDBC データ ソースのクローンの作成

1. [JDBC] ノードをクリックして展開します。
2. [データ ソース] ノードをクリックします。右ペインに [JDBC データ ソース] テーブルが表示され、ドメインに定義されているすべてのデータ ソースが示されます。
3. クローンを作成したいデータ ソースの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、データ ソースのクローン作成に関連するタブが示されます。
4. [名前]、[JNDI 名]、および [プール名] の各属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドに指定した名前のデータ ソース インスタンスを作成します。新しいインスタンスが左ペインの [データ ソース] ノードの下に追加されます。

JDBC データ ソースのすべてのインスタンスのモニタ

1. [JDBC] ノードをクリックして展開します。
2. 左ペインの [データ ソース] ノードをクリックします。右ペインに [JDBC データ ソース] テーブルが表示され、ドメインに定義されているすべてのデータ ソースが示されます。
3. モニタしたいデータ ソースの行の [すべてのインスタンスをモニタ] アイコンをクリックします。右ペインにダイアログが表示され、サーバ ドメイン全体でデプロイされている、そのデータ ソースのインスタンスがすべて示されます。

JDBC データ ソースの割り当て

1. 左ペインで、[データ ソース] の下のインスタンス ノードをクリックして、割り当てたいソースを指定します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ]、[グループ] および [クラスタ] の各タブについて、次の手順を実行します。
   a. データ ソースに割り当てたい対象を [選択可] カラムで任意の数だけ選択します。
   b. 移動コントロールをクリックして、選択した対象を [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

以下の表では、データ ソースのコンフィグレーションと管理に使われる属性について説明します。データ ソースの詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	当該データ ソースの名前を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	Null
[JNDI 名]	当該データ ソースに関連付けられる JNDI 名を設定する。当該 DataSource のバインド先の JNDI パスを設定する。	文字列	Null
[プール名]	当該 DataSource に関連付けられる JDBC 接続プールの名前を設定する。DataSource は、関連付けられている接続プールから接続を取り出して返す。	文字列	Null

属性	説明	値の範囲	デフォルト値
［行のプリフェッチを有効化］	true に設定した場合には、ResultSet ごとにクライアントと WebLogic Server の間で行のプリフェッチが有効になる。外部クライアントが Weblogic Server を通じて JDBC アクセスを行う際には、行のプリフェッチが有効であれば、1 回のサーバ アクセスで複数の行がサーバからクライアントにフェッチされるため、パフォーマンスが向上する。なお、クライアントと WebLogic Server が同じ JVM 上で動作する場合には、WebLogic Server はこの設定を無視し、行のプリフェッチを用いない。	ブール 選択されている = true 選択されていない = false	選択されていない = false

属性	説明	値の範囲	デフォルト値
［Row Prefetch サイズ］	`ResultSet` ごとにクライアントと WebLogic Server の間でプリフェッチされる行の数。最適値はクエリの詳細によって大きく異なる。一般に、この値を大きくすると、一定の値に達するまでは、パフォーマンスは向上するが、その値以上に大きくしても、パフォーマンスが著しく向上することはない。100 行以上でパフォーマンスが向上することはめったにない。ほとんどの状況では、デフォルト値で十分なはずである。	設定可能な最小値 = 2 設定可能な最大値 = 65536	48

属性	説明	値の範囲	デフォルト値
［ストリーム チャンク サイズ］	ResultSet ごとにクライアントと WebLogic Server の間でプリフェッチされる行の数。 最適値はクエリの詳細によって大きく異なる。一般に、この値を大きくすると、一定の値に達するまでは、パフォーマンスは向上するが、その値以上に大きくしても、パフォーマンスが著しく向上することはない。 100 行以上でパフォーマンスが向上することはめったにない。ほとんどの状況では、デフォルト値で十分なはずである。	整数 最小値 = 1 最大値 = 65,536	256

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	当該データ ソースの対象として使用するサーバをユーザが選択できるようにする。	リスト	Null

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	当該データ ソースの対象として使用するクラスタをユーザが選択できるようにする。	リスト	Null

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域	英数字の文字列	Null

データ ソースの詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

35 JDBC マルチプール

以下の手順では、Administration Console を利用してマルチプールのコンフィグレーションおよび管理用の属性を設定する 方法について説明します。マルチプールのコンフィグレーションと管理の詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

JDBC マルチプールのコンフィグレーション

1. [JDBC] ノードをクリックして展開します。
2. [マルチプール] ノードをクリックします。右ペインに [マルチプール] テーブルが表示され、ドメインに定義されているすべてのマルチプールが示されます。
3. [新しい JDBC Multi Pool のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示されて、新しく作成するマルチプールのコンフィグレーションに関連するタブが示されます。
4. ［一般］タブで以下の作業を行います。
   - [名前] 属性フィールドに値を入力します。
   - そのあと、ドロップダウン リストから [Load-Balancing] か [High-Availability] のどちらかを選択して、マルチプールから接続プールを選択する方法を決めます。
5. [作成] をクリックして、[名前] フィールドに指定した名前のマルチプールインスタンスを作成します。新しいインスタンスが左ペインの [マルチプール] ノードの下に追加されます。
6. ［プール］タブで以下の作業を行います。

a. ［選択可］カラム内のプール リストから、マルチプールに割り当てたい接続プールを選択します。

b. 移動コントロールをクリックして、選択した接続プールを［選択済み］カラムに移動します。

7. ［適用］をクリックして割り当てを保存します。

コンフィグレーション プロセスで次に行うべき作業については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

JDBC マルチプールのクローンの作成

1. [JDBC] ノードをクリックして展開します。
2. [マルチプール] ノードをクリックします。右ペインに [マルチプール] テーブルが表示され、ドメインに定義されているすべてのマルチプールが示されます。
3. クローンを作成したいマルチプールの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示されて、マルチプールのクローン作成に関連するタブが示されます。
4. [名前] 属性フィールドに値を入力します。[Load-Balancing] または [High-Availability] チェックボックスをクリックして、マルチプールから接続プールを選択する方法を選びます。
5. [クローン] をクリックして、[名前] フィールドに指定した名前のマルチプール インスタンスを作成します。新しいインスタンスが左ペインの [マルチプール] ノードの下に追加されます。

JDBC マルチプールのすべてのインスタンスのモニタ

1. [JDBC] ノードをクリックして展開します。

2. [マルチプール] ノードをクリックします。右ペインに [JDBC マルチプール] テーブルが表示され、ドメインに定義されているすべてのマルチプールが示されます。
3. モニタしたいマルチプールの行の [すべてのインスタンスのモニタ] アイコンをクリックします。右ペインにダイアログが表示され、接続プール ドメイン全体でデプロイされているマルチプールのインスタンスがすべて示されます。

サーバ、クラスタへの JDBC マルチプールの割り当て

1. 左ペインで、[マルチプール] の下のインスタンス ノードをクリックして、割り当て対象のマルチプールを指定します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. ［サーバ］タブ、あるいは [クラスタ] タブで以下の手順を実行します。
   a. マルチプールの割り当て先の サーバを [選択可] カラムで選択します。
   b. 移動コントロールをクリックして、選択した対象を [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	JDBC マルチプールの名前を設定する。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL

属性	説明	値の範囲	デフォルト値
［アルゴリズム タイプ］	マルチプールから接続プールを選択する方法を選ぶ。		
	ロード バランス (Load-Balancing) 接続要求をメンバープールに均等に配分する（次に利用可能なプールからのラウンドロビン）。	ブール True = 選択されている False = 選択されていない	選択されていない
	高可用性 (Hight-Availability) 接続要求を順序付きリストから順次配分する。 すなわち、マルチプールに接続要求があるたびに、そのリストの先頭のプールから接続の取得を試みる。有効な接続を取得できない場合には、そのリストの次のプールを試す。有効な接続が取得されるか、あるいはリストの終わりに達するまで、この処理が繰り返される。なお、リストの終わりに達した場合には、例外が送出される。 データベースが停止していたり、プールが使用できないといった問題が実際に発生する場合にのみ、マルチプールがリスト内の次のプールに変わる点に注意。すべての接続がビジー状態にある場合には、マルチプールは単一のプールとして動作し、例外が送出される。	ブール True = 選択されている False = 選択されていない	選択されている

[プール]

属性	説明	値の範囲	デフォルト値
[プール リスト]	マルチプール内の接続プールのリストが取得されて格納される。	リスト	

[対象]

属性	説明	値の範囲	デフォルト値
［対象サーバ］			

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字の文字列	NULL

マルチプールのコンフィグレーションと管理の詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

36 JDBC トランザクション データ ソース

この表では、トランザクション データ ソースのコンフィグレーションと管理のために Administration Console で設定される属性について説明します。トランザクション データ ソースの詳細については、『WebLogic Server 管理者ガイド』の「JDBC 接続の管理」を参照してください。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	当該データ ソースの名前を返す。	文字列	MyJDBC Tx Data Source
[JNDI 名]	当該 TxDataSource のバインド先の JNDI パスを設定。この JNDI パスをルックアップするアプリケーションは、当該 TxDataSource に対応する `javax.sql.DataSource` インスタンスを取得することになる。なお、データソースをルックアップして接続を取得する方法が採用されたため、`DriverManager.getConnection()` や `Driver.Connect()` という旧式の使い方は廃止された点に注意。	有効な JNDI 名	Null
[プール名]	当該 TxDataSource に関連付けられる 接続プールの名前をユーザが設定できるようにする。当該 TxDataSource に対して `getConnection()` を呼び出すと、関連付けられている接続プール内の接続が返される。	有効な接続プール名	Null

属性	説明	値の範囲	デフォルト値
[2 フェーズ コミットを有効化]	トランザクションの 2 フェーズ コミットをユーザが有効 / 無効にできるようにする。この属性を指定すれば、XA 非対応の JDBC ドライバがトランザクションに参加していても JDBC 接続だけがそのトランザクションに参加しているように見せかけることができるが、複数のリソースがトランザクションに参加していて、そのうちの 1 つ（JDBC ドライバ）が XA リソースのように見せかける場合には、問題が起きることがある。すなわち、そうした状況では、ヒューリスティック障害が発生するおそれがある。この属性を true に設定するのは、そうせざるを得ない場合（たとえば、問題になっているデータベースに適した XA ドライバがない場合など）に留めるべきである。 なお、当該 TxDataSource が XA 接続プールに関連付けられている場合や、分散トランザクションに参加しているリソースが 1 つしかない場合には、この設定は無視される。	ブール 有効 = 選択されている 無効 = 選択されていない	選択されていない

属性	説明	値の範囲	デフォルト値
［行のプリフェッチを有効化］	True に設定した場合には、ResultSet ごとにクライアントと WebLogic Server の間で行のプリフェッチが有効になる。外部クライアントが Weblogic Server を通じて JDBC アクセスを行う際には、行のプリフェッチが有効であれば、1 回のサーバ アクセスで複数の行がサーバからクライアントにフェッチされるため、パフォーマンスが向上する。クライアントと WebLogic Server が同じ JVM 上で動作する場合には、WebLogic Server はこの設定を無視し、行のプリフェッチを用いない。	ブール 有効 = 選択されている 無効 = 選択されていない	選択されていない

属性	説明	値の範囲	デフォルト値
［Row Prefetch サイズ］	ResultSet ごとにクライアントと WebLogic Server の間でプリフェッチされる行の数。最適値はクエリの詳細によって大きく異なる。一般に、この値を大きくすると、一定の値に達するまでは、パフォーマンスは向上するが、その値以上に大きくしても、パフォーマンスが著しく向上することはない。100 行以上でパフォーマンスが向上することはめったにない。ほとんどの状況では、デフォルト値で十分なはずである。	設定可能な最小値 = 2 設定可能な最大値 = 65536	48
［ストリーム チャンク サイズ］	ストリーム データ型のデータ チャンク サイズを指定する。ストリーム データ型（たとえば、getBinarySTream() の呼び出しで得られるデータなど）は、必要に応じて、この属性で指定されたサイズのチャンク（かたまり）に分けて、WebLogic Server からクライアントへ送られる。	設定可能な最小値 = 1 設定可能な最大値 = 65,536 （単位 : バイト）	256

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	当該データソースの対象として使用するサーバをユーザが選択できるようにする。	リスト	Null

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	当該データソースの対象として使用するクラスタをユーザが選択できるようにする。	リスト	Null

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	必要に応じてユーザが情報を入力するための領域	英数字の文字列	Null

37 JMS 接続コンシューマ

JMS 接続コンシューマの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [セッション プール] ノードをクリックして展開します。
5. [セッション プール] の下のセッション プールをクリックして展開します。
6. [コンシューマ] ノードをクリックします。すべての接続コンシューマを示す [JMS 接続コンシューマ] テーブルが右ペインに表示されます。
7. [新しい JMS Connection Consumer のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しい接続コンシューマの設定に関連するタブが示されます。
8. 属性フィールドに値を入力します。
9. [作成] をクリックして、[名前] フィールドで指定した名前の接続コンシューマ インスタンスを作成します。新しいインスタンスが左ペインの [コンシューマ] ノードの下に追加されます。

JMS 接続コンシューマのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。

4. [セッション プール] ノードをクリックして展開します。
5. [セッション プール] の下のセッション プールをクリックして展開します。
6. [コンシューマ] ノードをクリックします。すべての接続コンシューマを示す [JMS 接続コンシューマ] テーブルが右ペインに表示されます。
7. クローンを作成する接続コンシューマの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、接続コンシューマのクローンの作成に関連するタブが示されます。
8. 属性フィールドに値を入力します。
9. [作成] をクリックして、[名前] フィールドで指定した名前の接続コンシューマ インスタンスを作成します。新しいインスタンスが左ペインの [コンシューマ] ノードの下に追加されます。

JMS 接続コンシューマの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [セッション プール] ノードをクリックして展開します。
5. [セッション プール] の下のセッション プール インスタンスをクリックして展開します。
6. [コンシューマ] ノードをクリックします。すべての接続コンシューマを示す [JMS 接続コンシューマ] テーブルが右ペインに表示されます。
7. 削除する接続コンシューマの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
8. [はい] をクリックして接続コンシューマを削除します。[JMS 接続コンシューマ] ノードの下の接続コンシューマ アイコンが削除されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	接続コンシューマの名前。 この属性は動的にコンフィグレーションできない。	JMS サーバ セッション プール内でユニークな有効 Java 識別子	MyJMS Connection Consumer[-*n*]
[最大メッセージ]	接続コンシューマによって蓄積されるメッセージの最大数。値 -1 は最大数が設定されていないことを示す。 この属性は動的にコンフィグレーションできるが、変更は接続コンシューマのセッション プールが再起動されるまで有効にならない。	-1、1 ～ 2^{63} ñ 1	-1
[セレクタ]	メッセージのフィルタリングに使用する JMS セレクタ式。セレクタの定義の詳細については、『*WebLogic JMS プログラマーズ ガイド*』を参照。 この属性は動的にコンフィグレーションできない。	有効な JMS メッセージ セレクタ式	なし

属性	説明	値の範囲	デフォルト値
[送り先]	接続コンシューマがリスニング対象とする送り先。 この属性は動的にコンフィグレーションできない。	有効な JNDI 名または NULL	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

38 JMS 接続ファクトリ

JMS 接続ファクトリの作成

1. [JMS] ノードをクリックして展開します。
2. [接続ファクトリ] ノードをクリックします。ドメインで定義されているすべての接続ファクトリを示す [JMS 接続ファクトリ] テーブルが右ペインに表示されます。
3. [新しい JMS Connection Factory のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する接続ファクトリの設定に関連するタブが示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドで指定した名前の接続ファクトリ インスタンスを作成します。新しいインスタンスが左ペインの [JMS 接続ファクトリ] ノードの下に追加されます。
6. WebLogic Sever に JMS 接続ファクトリを割り当てます。

詳細については、「JMS 接続ファクトリの割り当て」を参照してください。

JMS 接続ファクトリのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [接続ファクトリ] ノードをクリックします。ドメインで定義されているすべての接続ファクトリを示す [JMS 接続ファクトリ] テーブルが右ペインに表示されます。

3. クローンを作成する接続ファクトリの行の [クローン] アイコンをクリックします。接続ファクトリのクローンの作成に関連付けられているタブを示すダイアログが右ペインに表示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドで指定した名前の接続ファクトリ インスタンスを作成します。新しいインスタンスが左ペインの [JMS 接続ファクトリ] ノードの下に追加されます。
6. WebLogic Sever に JMS 接続ファクトリを割り当てます。

詳細については、「JMS 接続ファクトリの割り当て」を参照してください。

JMS 接続ファクトリの削除

1. [JMS] ノードをクリックして展開します。
2. [接続ファクトリ] ノードをクリックします。ドメインで定義されているすべての接続ファクトリを示す [JMS 接続ファクトリ] テーブルが右ペインに表示されます。
3. 削除する接続ファクトリの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
4. [はい] をクリックして接続ファクトリを削除します。[JMS 接続ファクトリ] ノードの下の接続ファクトリ アイコンが削除されます。

JMS 接続ファクトリの割り当て

1. 左ペインで、接続ファクトリに割り当てる [JMS 接続ファクトリ] の下のインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。

a. 接続ファクトリに割り当てる 1 つまたは複数の対象を [選択可] カラムで選択します。

b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。

c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	接続ファクトリの名前。JNDI 名は別途コンフィグレーションされる。 この属性は動的にコンフィグレーションできない。	特定のクラスタ内でユニークな Java 識別子	MyJMS Connection Factory[-*n*]
[JNDI 名]	接続ファクトリに割り当てられ、JNDI ネームスペース内で接続ファクトリの検索に使用される名前。接続ファクトリ名は別途コンフィグレーションされる。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペース内でユニークな Java 識別子	NULL
[クライアント ID]	永続的サブスクライバを持つクライアント用のクライアント ID。永続的サブスクライバの詳細については、WebLogic JMS プログラマーズ ガイドを参照。 この属性は動的にコンフィグレーションできない。	Java 識別子	NULL

属性	説明	値の範囲	デフォルト値
[デフォルト 優先順位]	優先順位が明示的に定義されていない場合にメッセージに適用するデフォルトの優先順位。 メッセージ送信時の優先順位の定義については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	0 ～ 9	4
[デフォルト 生存時間]	メッセージのデフォルトの最長存在期間（ミリ秒単位）。優先順位が明示的に定義されていない場合にメッセージに適用される。値 0 はメッセージが無限に存在することを示す。 メッセージ送信時の存在期間の定義については、WebLogic JMS プログラマーズ ガイドを参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	0 ～ 2^{63} ñ 1	0
[デフォルト 配信モード]	配信モードが明示的に定義されていない場合にメッセージに適用するデフォルトの配信モード。 メッセージ送信時の配信モードと定義については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	[永続] または [非永続]	永続

属性	説明	値の範囲	デフォルト値
[デフォルト 再送遅延]	ロールバックまたは回復されたメッセージが再配信されるまでのデフォルトの遅延時間 (ミリ秒単位)。 メッセージ送信時の配信モードと定義については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	-1 ～ 2^{63} ñ 1	0
[デフォルト 送信時間]	メッセージが生成されてから送り先で表示されるようになるまでのデフォルトの遅延時間 (ミリ秒単位)。 メッセージ送信時の配信モードと定義については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	-1 ～ 2^{63} ñ 1	0

属性	説明	値の範囲	デフォルト値
[最大メッセージ]	非同期セッション向けに存在し、メッセージ リスナに渡されていないメッセージの最大数。 値 -1 はメッセージ数を制限しないことを示す。ただし、その場合、使用可能な仮想メモリ量が制限される。 メッセージ数が [最大メッセージ] の値に達した場合の処理は次のとおり。 ♦ マルチキャスト セッションの場合、新しいメッセージは [超過時のポリシー] 属性によって指定されたポリシーに従って破棄され、DataOverrunException が発生する。 ♦ 非マルチキャスト セッションの場合、新しいメッセージはフロー制御されるか、アプリケーションがメッセージを受け入れることができるまでサーバ上に保持される。 マルチキャスト セッションの場合、接続が停止してもメッセージの配信は、[最大メッセージ] 値に達するまで続行される。この値に達すると、メッセージは超過時のポリシーに従って破棄される。 非同期メッセージおよびマルチキャストの詳細については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	-1、1 ～ 2^{31} ñ 1	10

属性	説明	値の範囲	デフォルト値
[超過時のポリシー]	マルチキャスト セッションの超過時のポリシー。未処理のメッセージ数が [最大メッセージ] 属性値に達すると、メッセージは指定されたポリシーに従って破棄される。 [新しいメッセージを保持] に設定した場合、最新のメッセージが最も古いメッセージに対して優先され、必要に応じて、古いメッセージは破棄される。 [古いメッセージを保持] に設定した場合、最も古いメッセージが最新のメッセージに対して優先され、必要に応じて、最新のメッセージは破棄される。 メッセージの存在期間は受け取り順序によって決まり、`JMSTimestamp` 値によって定義されるわけではない。 非同期メッセージおよびマルチキャストの詳細については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	[新しいメッセージを保持]、[古いメッセージを保持]	古いメッセージを保持

属性	説明	値の範囲	デフォルト値
[メッセージの短縮を許可]	close() メソッドを onMessage() メソッド呼び出し内で発行できるようにするメッセージ コンシューマを接続ファクトリが作成するかどうかを指定するフラグ。 このフラグを有効にした場合、永続的なブロッキングではなく、onMessage() メソッド呼び出し内からの close() メソッド呼び出しが成功する。 セッションの通知モードを AUTO_ACKNOWLEDGE に設定した場合、onMessage() が完了しても現在のメッセージは自動的に通知される。 onMessage() メソッドは、『WebLogic JMS プログラマーズ ガイド』に説明されている非同期メッセージ配信をサポートするために実装される。	ブール 有効 = 選択 無効 = 未選択	選択されていない

属性	説明	値の範囲	デフォルト値
[確認応答ポリシー]	接続ファクトリの確認応答ポリシー。この属性が適用されるのは、非トランザクション セッションで CLIENT_ACKNOWLEDGE 確認応答モードを使用する実装の場合のみ。 ♦ [All] - どのメッセージが確認応答メソッドを呼び出したかにかかわらず、セッションが受け取ったすべてのメッセージが確認応答される。 ♦ [Previous] - セッションが受け取ったすべてのメッセージのうち、確認応答を呼び出したメッセージまでが確認応答される。 確認応答モードについては、『WebLogic JMS プログラマーズ ガイド』を参照。 **注意 :** デフォルト値 [All] は、JMS 仕様の変更を反映している。WebLogic Server 6.1 のリリースより前のバージョンの JMS では、デフォルトは Previous だった。詳細については、『WebLogic JMS プログラマーズ ガイド』の「WebLogic JMS アプリケーションの移行」を参照。	[All] または [Previous]	All

[トランザクション]

属性	説明	値の範囲	デフォルト値
[トランザクション タイムアウト]	実行されたセッションのタイムアウト値（秒単位）。タイムアウト値で指定された時間を経過しても実行されたセッションがアクティブな場合、トランザクションはロール バックされる。値 0 は、デフォルト値が使用されることを示す。 長時間実行するトランザクションがある場合、トランザクションが完了するようにこの属性の値を調整する。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	0 ～ 2^{31} ñ 1	3600

属性	説明	値の範囲	デフォルト値
[ユーザ トランザクションを有効化]	接続ファクトリが JTA 対応のセッションを作成するかどうかを指定するためのフラグ。このフラグを設定した場合、関連するメッセージ プロデューサとメッセージ コンシューマは、トランザクション コンテキストについて実行中のスレッドを調べる。設定していない場合は、JTA トランザクションを無視する。 ただし、[XA Connection Factory Enabled] フラグが設定されている場合、[ユーザ トランザクションを有効化] は常に有効とみなされるため、無視される。 **注意：** トランザクション セッションでは、内部のトランザクションを優先するため、この設定に関わりなく現在のスレッドのトランザクション コンテキストを無視する。この設定は非トランザクション セッションのみに影響する。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	ブール 有効 = 選択 無効 = 未選択	選択されていない

属性	説明	値の範囲	デフォルト値
[XA コネクションファクトリを有効化]	キューまたはトピック接続ファクトリの代わりに、XA キューまたは XA トピック接続ファクトリが返されるかどうかを指定するフラグ。`getXAResource` メソッドを持つ XA キュー セッションまたは XA トピック セッションを返す場合にも使用できる。 この属性は動的にコンフィグレーションできるが、変更は新しい接続に対してのみ影響し、既存の接続は影響を受けない。	ブール 有効 = 選択 無効 = 未選択	選択されていない

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	選択可能な対象のリストをサーバに提供する。JMS クライアントが接続を作成する場合、このファクトリを使用していずれかの対象に接続する。	選択可能な対象および選択済み対象のリスト	対象が選択されていない状態

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	選択可能な対象のリストをクラスタに提供する。各 JMS サーバに割り当て可能な対象は 1 つのみ。	選択可能な対象および選択済み対象のリスト	対象が選択されていない状態

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

39 実行時 JMS 接続

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- クライアント ID
- アクティブなセッション
- 最大セッション数
- 総セッション数

40 実行時 JMS コンシューマ

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前
- 最大メッセージ数
- セレクタ
- 送り先

41 JMS 送り先

JMS キューの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS キューを示す [JMS 送り先] テーブルが右ペインに表示されます。
5. [新しい JMSQueue のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するキューの設定に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、キューのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [送り先] ノードの下に追加されます。
8. そのほかのタブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
9. [適用] をクリックして、変更を保存します。

JMS キューのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。

3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS キューを示す [JMS 送り先] テーブルが右ペインに表示されます。
5. クローンを作成するキューの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、新しいキューのクローンの作成に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. 右下隅の [作成] ボタンをクリックして、キューのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [送り先] ノードの下に追加されます。
8. そのほかのタブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
9. [適用] をクリックして、変更を保存します。

JMS キューの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS キューを示す [JMS 送り先] テーブルが右ペインに表示されます。
5. 削除するキューの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
6. [はい] をクリックしてキューを削除します。[送り先] ノードの下のキューアイコンが削除されます。

JMS トピックの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS トピックを示す [JMS 送り先] テーブルが右ペインに表示されます。
5. [新しい JMSTopic のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しいトピックの設定に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. 右下隅の [作成] ボタンをクリックして、トピックのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [送り先] ノードの下に追加されます。
8. そのほかのタブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
9. [適用] をクリックして、変更を保存します。

JMS トピックのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS トピックを示す [JMS 送り先] テーブルが右ペインに表示されます。

5. クローンを作成するトピックの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、新しいトピックのクローンの作成に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. 右下隅の [作成] ボタンをクリックして、トピックのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [送り先] ノードの下に追加されます。
8. そのほかのタブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
9. [適用] をクリックして、変更を保存します。

JMS トピックの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS トピックを示す [JMS 送り先] テーブルが右ペインに表示されます。
5. 削除するトピックの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
6. [はい] をクリックしてトピックを削除します。[送り先] ノードの下のトピック アイコンが削除されます。

すべてのアクティブな JMS の送り先のモニタ

詳細については、「JMS サーバ」の「すべてのアクティブな JMS の送り先のモニタ」を参照してください。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト
[名前]	送り先の名前。JNDI 名は別途コンフィグレーションされる。 この属性は動的にコンフィグレーションできない。	JMS サーバ内でユニークな Java 識別子	My JMS Destination[-*n*]
[JNDI 名]	JNDI ネームスペース内で送り先の検索に使用される名前。送り先名は別途コンフィグレーションされる。この属性を指定しない場合、JNDI ネームスペース内で送り先名がアドバタイズされず、ルックアップおよび使用することができない。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペースのスコープ内でユニークな Java 識別子	NULL

属性	説明	値の範囲	デフォルト
[ストアを有効化]	送り先が JMS サーバに指定された永続ストレージを使用するかどうかを指定するフラグ。 ■ このフラグを有効に設定したが、永続ストレージが定義されていない場合、設定は失敗し、WebLogic JMS は起動しない。 ■ このフラグを無効に設定した場合、送り先は永続的なメッセージをサポートしない。 ■ このフラグを [デフォルト] に設定した場合、永続ストレージが JMS サーバに定義されていれば、送り先は永続ストレージを使用し、永続的なメッセージをサポートする。 この属性は動的にコンフィグレーションできない。	無効、有効、デフォルト	デフォルト
[テンプレート]	送り先の派生元の JMS テンプレート。この属性が定義されていない場合、送り先の属性を送り先の一部として指定する必要がある。 送り先ごとに設定された [テンプレート] 属性は静的だが、動的に変更することができる。 **注意：** デフォルト値に設定されている属性は、JMS テンプレートから送り先の値を実行時に継承する。	既存の JMS テンプレート名またはなし	なし

属性	説明	値の範囲	デフォルト
[送り先キー]	送り先に到着するメッセージを並べ替える場合に、選択可能な送り先キーのリストを提供する。リストは、重要度の高いキーから低いキーへの順に並べられる。複数のキーが指定されると、JMSMessageID に基づくキーのみが、リストの最後のキーとして使用される。 **注意 :** キーで JMSMessageID が定義されていない場合は、暗黙的に最後のキーとみなされ、ソート順に「昇順」（先入れ、先出し）が設定される。	リスト	NULL

[しきい値と割当]

属性	説明	値の範囲	デフォルト
[最大バイト数]	送り先に保存可能な最大バイト数。値 -1 は送り先に保存可能なバイト数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、新しく配信されるメッセージにのみ適用され、すでに保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大バイトしきい値] 以上	-1

属性	説明	値の範囲	デフォルト
[最大バイトしきい値]	送り先に保存されているバイト数に基づく上限しきい値。 バイト数がこのしきい値を超えた場合、バイト ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのバイト ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。 **注意 :** [最大バイトしきい値] を -1 にリセットすることでバイト ページングを動的に無効にすることはできない。ページングを無効にするには、[最大バイトしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}-1、[最大バイト数] 以下、[最小バイトしきい値] より大	-1

属性	説明	値の範囲	デフォルト
[最小バイトしきい値]	送り先に保存されているバイト数に基づく下限しきい値。 バイト数がこのしきい値より少なくなると、送り先レベルのバイト ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大バイトしきい値] より小	-1

属性	説明	値の範囲	デフォルト
[Bytes Paging Enabled]	送り先でバイト ページングを有効にするかどうかを示すフラグ。 ■ この属性を [False] に設定すると、この送り先に対する送り先レベルのバイト ページングは明示的に無効になる。 ■ この属性を [True] に設定すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小バイトしきい値] 属性と [最大バイトしきい値] 属性の値がどちらも -1 より大きい場合は、この送り先に対する送り先レベルのバイト ページングが有効になる。 ■ この属性を [Default] に設定すると、JMS テンプレートが指定されている場合は、JMS テンプレートの値が継承される。送り先に対してテンプレートがコンフィグレーションされていない場合は、[Default] による結果は [False] と同じである。 **注意：** サーバ レベルのバイト ページングが有効である場合は、特定の送り先に対する送り先レベルのページングが無効であっても、サーバ レベルのページングが開始していると、送り先でメッセージをページングできる。ただし、特定の送り先に対して送り先レベルのページングが無効になっていると、送り先レベルの最大しきい値の設定によって、しきい値を超えたときにメッセージが強制的にページ アウトされることはない。	[String] [True] [False] [Default]	[Default]

属性	説明	値の範囲	デフォルト
[最大メッセージ数]	送り先に保存可能なメッセージの最大数。値 -1 は送り先に保存可能なメッセージ数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大メッセージしきい値] 以上	-1
[最大メッセージしきい値]	送り先に保存されるメッセージ数に基づく上限しきい値。 メッセージ数がこのしきい値を超えた場合、メッセージ ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。 **注意 :** [最大メッセージしきい値] を -1 にリセットすることでメッセージ ページングを動的に無効にすることはできない。ページングを無効にするには、[最大メッセージしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}-1、[最大メッセージ数] 以下、[最小メッセージしきい値] より大	-1

属性	説明	値の範囲	デフォルト
[最小メッセージしきい値]	送り先に保存されるメッセージ数に基づく下限しきい値。 メッセージ数がこのしきい値より少なくなると、送り先レベルのメッセージページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大メッセージしきい値] より小	-1

属性	説明	値の範囲	デフォルト
[Messages Paging Enabled]	送り先でメッセージ ページングを有効にするかどうかを示すフラグ。 ■ この属性を [False] に設定すると、この送り先に対する送り先レベルのメッセージ ページングが明示的に無効になる。 ■ この属性を [True] に設定すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小メッセージしきい値] 属性と [最大メッセージしきい値] 属性の値がどちらも -1 より大きい場合は、この送り先に対する送り先レベルのメッセージ ページングが有効になる。 ■ この属性を [Default] に設定すると、JMS テンプレートが指定されている場合は、JMS テンプレートの値が継承される。送り先に対してテンプレートがコンフィグレーションされていない場合は、[Default] による結果は [False] と同じである。 **注意：** サーバ レベルのメッセージ ページングが有効である場合は、特定の送り先に対する送り先レベルのページングが無効であっても、サーバ レベルのページングが開始していると、送り先でのメッセージをページングできる。ただし、特定の送り先に対して送り先レベルのページングが無効になっていると、送り先レベルの最大しきい値の設定によって、しきい値を超えたときにメッセージが強制的にページアウトされることはない。	[String] [True] [False] [Default]	[Default]

[オーバライド]

属性	説明	値の範囲	デフォルト
[優先順位オーバライド]	メッセージ プロデューサによって指定された優先度に関わりなく、送り先に届くすべてのメッセージに割り当てられる優先度の値。 デフォルト値 (-1) は送り先が優先度の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 9	-1
[生存時間オーバライド]	メッセージ プロデューサによって指定された生存時間に関わりなく、送り先に届くすべてのメッセージに割り当てられる生存時間の値。 デフォルト値 (-1) は送り先が生存時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1	-1

属性	説明	値の範囲	デフォルト
[送信時間オーバーライド]	メッセージが生成されてから送り先で表示できるようになるまでのデフォルトの遅延時間（ミリ秒単位）を、プロデューサまたは接続ファクトリによって指定された送信時間に関わらず定義する。 デフォルト値 (-1) は送り先が送信時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1 以上の長整数値、または配信スケジュールの文字列構文	-1
[配信モードのオーバライド]	メッセージ プロデューサによって指定された配信モードに関わりなく、送り先に届くすべてのメッセージに割り当てられる配信モード。 [配信しない] は配信モードがオーバーライドされないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	[永続]、[非永続]、または [配信しない]	[配信しない]

[再配信]

属性	説明	値の範囲	デフォルト
[再配信遅延のオーバライド]	コンシューマや接続ファクトリによって指定された再配信遅延に関わりなく、ロール バックまたは回復されたメッセージが再配信されるまでの遅延をミリ秒単位で定義する。 デフォルト値 (-1) は送り先が再配信遅延の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1	-1

属性	説明	値の範囲	デフォルト
[再配信の制限]	メッセージがエラー送り先に置かれるまでに試行できる再配信の回数。エラー送り先がコンフィグレーションされているかどうかにより、再配信の上限に達したときの対応は以下のようになる。 ■ エラー送り先がコンフィグレーションされていないか、エラー送信先の割り当てが超過している場合には、永続メッセージおよび非永続メッセージは単に削除される。 ■ エラー送り先がコンフィグレーションされていて、エラー送り先が割り当て内の場合には、エラーメッセージが記録され、そのメッセージは削除される。ただし、永続メッセージの場合は永続ストレージに残る。これにより、WebLogic Server が再起動されると永続メッセージが再配信される。 デフォルト値 (-1) は送り先が再配信制限の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1	-1
[エラー送り先]	再配信の上限に達したメッセージの送り先。エラー送り先が NULL の場合は、メッセージは単に削除される。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	既存の送り先またはなし	なし

[マルチキャスト]

属性	説明	値の範囲	デフォルト
[マルチキャスト アドレス]	マルチキャスト用の IP アドレス。このアドレスはマルチキャスト コンシューマにメッセージを送信するために使用される。 この属性はトピックに関してのみ有効で、動的にコンフィグレーションすることはできない。	正しい形式の IP アドレス（クラス D）	なし
[マルチキャスト 生存時間]	マルチキャストの生存時間の値。メッセージがコンシューマに到達するまでに通過するルータの数を指定する。値 0 はメッセージがルータを越えず、1 つのサブネット内に制限されることを示す。 この値は `JMSExpirationTime` の値とは無関係である。 この属性はトピックに関してのみ有効で、動的にコンフィグレーションすることはできない。	0 ～ 255	0
[マルチキャスト ポート]	マルチキャスト用の IP ポート。このポートはマルチキャスト コンシューマにメッセージを送信するために使用される。 この属性はトピックに関してのみ有効で、動的にコンフィグレーションすることはできない。	1024 ～ 65535	6001

[送り先のモニタ]

属性	説明	値の範囲	デフォルト
[送り先]	送り先名。		なし
[サーバ]	関連するサーバ名。		なし
[コンシューマ数]	現在の登録済みメッセージ コンシューマの数。		なし
[最大コンシューマ数]	任意の時点での登録済みメッセージ コンシューマの最大数。		なし
[コンシューマ総数]	登録済みメッセージ コンシューマの総数。		なし
[現在のバイト数]	現在保存されているバイト数。		なし
[保留バイト数]	保存されていて、確認応答およびコミットが行われていないトランザクション バイト数。		なし
[受信バイト数]	受信済みバイト数。		なし
[バイトしきい値の時間]	サーバの起動以降、送り先がバイトしきい値状態になっていた時間の合計。		なし
[メッセージ数]	この送り先に現在格納されているメッセージの数。		なし
[最大メッセージ数]	ある時点において格納されるメッセージの最大数。		なし

属性	説明	値の範囲	デフォルト
[保留メッセージ数]	保存されていて、確認応答およびコミットが行われていないトランザクション メッセージの数。		なし
[受信メッセージ数]	受信済みメッセージ数。		なし
[メッセージしきい値の時間]	送り先がメッセージしきい値状態になっていた時間の合計。		なし

[メモ]

属性	説明	値の範囲	デフォルト
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字の文字列	NULL

42 JMS 送り先キー

JMS 送り先キーの作成

1. [JMS] ノードをクリックして展開します。
2. [JMS 送り先キー] ノードをクリックします。すべての送り先キーを示す [JMS 送り先キー] テーブルが右ペインに表示されます。
3. [新しい JMS Destination Key のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する送り先キーの設定に関連するタブが示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、送り先キーのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [JMS 送り先キー] ノードの下に追加されます。

JMS 送り先キーのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [JMS 送り先キー] ノードをクリックします。すべての送り先キーを示す [JMS 送り先キー] テーブルが右ペインに表示されます。
3. クローンを作成する送り先キーの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、送り先キーのクローンの作成に関連するタブが示されます。
4. 属性フィールドに値を入力します。

5. [作成] をクリックして、送り先キーのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [JMS 送り先キー] ノードの下に追加されます。

JMS 送り先キーの削除

1. [JMS] ノードをクリックして展開します。
2. [JMS 送り先キー] ノードをクリックします。すべての送り先キーを示す [JMS 送り先キー] テーブルが右ペインに表示されます。
3. 削除する送り先キーの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
4. [はい] をクリックして送り先キーを削除します。[JMS 送り先キー] ノードの下の送り先キー アイコンが削除されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	キーの名前。 この属性は動的にコンフィグレーションできない。	特定のドメイン内でユニークな Java 識別子	MyJMS Destination Key[-*n*]

属性	説明	値の範囲	デフォルト値
[プロパティ]	並べ替えの基準となるプロパティ名。この値は、メッセージのプロパティ名、または並べ替えの基準となるメッセージ ヘッダ フィールドの名前を示す。 メッセージ ヘッダ フィールド キーは、メッセージ プロパティではなく、キー タイプおよび参照メッセージ ヘッダ フィールドを無視する。 **注意：** パフォーマンスを改善するには、メッセージ プロパティではなくメッセージ ヘッダ フィールドを並べ替えのキーとして使用すること。 この属性は動的にコンフィグレーションできない。	JMS プロパティ名 (ユーザ プロパティを含む)、または並べ替えの基準にできるメッセージ ヘッダ フィールドは以下の通り。 ♦ JMSMessageID ♦ JMSTimestamp ♦ JMSCorrelationID ♦ JMSPriority ♦ JMSExpiration ♦ JMSType ♦ JMSRedelivered ♦ JMSDeliveryTime	なし
[キー タイプ]	想定しているプロパティの型。 この属性は動的にコンフィグレーションできない。	[Boolean]、[Byte]、[Double]、[Float]、[Int]、[Long]、[Short]、[String]	String

属性	説明	値の範囲	デフォルト値
[ソート順]	送り先を並べ替える方向。 JMSMessageID に [昇順] を選択すると、ソート順は先入れ先出し（送り先のデフォルト）になる。ソート順を後入れ先出しにするには、値を [降順] に設定する。 この属性は動的にコンフィグレーションできない。	[昇順] または [降順]	昇順

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

43 実行時 JMS 送り先

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 送り先
- サーバ
- コンシューマ
- メッセージ数
- 受信メッセージ数
- 現在のバイト数
- 保留バイト数

44 実行時 JMS 恒久サブスクライバ

デフォルトでは、このペインを使用すると、ユーザが次の基準で JMS の恒久サブスクライバをモニタすることができます。

属性	説明	値の範囲	デフォルト値
[クライアント ID]	この恒久サブスクライバのクライアント ID。		なし
[Subscription Name]	この恒久サブスクライバのサブスクリプション名。		なし
[No Local]	この恒久サブスクライバの `noLocal` ブール値。		なし
[アクティブ]	恒久サブスクライバでこのサブスクリプションを使用するかどうかを指定する。		なし
[セレクタ]	恒久サブスクライバのセレクタ。		なし
[Messages Pending Count]	この恒久サブスクライバの保留メッセージ数（未処理および未確定のメッセージ数）。		なし
[Messages Current Count]	恒久サブスクライバで現在使用できるメッセージ数。		なし
[Bytes Pending Count]	恒久サブスクライバで保留中のバイト数。		なし
[Bytes Current Count]	この恒久サブスクライバの受信バイト数。		なし

45 JMS ファイル ストア

JMS ファイル ストアの作成

1. [JMS] ノードをクリックして展開します。
2. [ストア] ノードをクリックします。すべての JMS ファイル ストアを示す [JMS ストア] テーブルが右ペインに表示されます。
3. [新しい JMSFile Store のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するファイル ストアの設定に関連するタブが示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、ファイル ストアのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [ストア] ノードの下に追加されます。

JMS ファイル ストアのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [ストア] ノードをクリックします。すべての JMS ファイル ストアを示す [JMS ストア] テーブルが右ペインに表示されます。
3. クローンを作成するファイル ストアの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、ファイル ストアのクローンの作成に関連するタブが示されます。
4. 属性フィールドに値を入力します。

5. [作成] をクリックして、ファイル ストアのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [ストア] ノードの下に追加されます。

JMS ファイル ストアの削除

1. [JMS] ノードをクリックして展開します。
2. [ストア] ノードをクリックします。すべての JMS ファイル ストアを示す [JMS ストア] テーブルが右ペインに表示されます。
3. 削除するファイル ストアの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
4. [はい] をクリックしてファイル ストアを削除します。[ストア] ノードの下のファイル ストア アイコンが削除されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	バッキング ストア（またはストア）の名前。この名前はストア ファイルのプレフィックスとして使用される。 この属性は動的にコンフィグレーションできない。	JMS サーバ内でユニークな Java 識別子	MyJMSFile Store[-*n*]
[ディレクトリ]	ファイル バッキング ストアのディレクトリを指定する。 この属性は動的にコンフィグレーションできない。	バッキング ストアがインスタンス化されるシステム上の有効なディレクトリ	なし

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

46 JMS JDBC ストア

JMS JDBC ストアの作成

1. [JMS] ノードをクリックして展開します。
2. [ストア] ノードをクリックします。すべての JMS JDBC ストアを示す [JMS ストア] テーブルが右ペインに表示されます。
3. [新しい JMSJDBCStore のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する JDBC ストアの設定に関連するタブが示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、JDBC ストアのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [ストア] ノードの下に追加されます。

JMS JDBC ストアのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [ストア] ノードをクリックします。すべての JMS JDBC ストアを示す [JMS ストア] テーブルが右ペインに表示されます。
3. クローンを作成する JDBC ストアの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、JDBC ストアのクローンの作成に関連するタブが示されます。
4. 属性フィールドに値を入力します。

5. [作成] をクリックして、JDBC ストアのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [ストア] ノードの下に追加されます。

JMS JDBC ストアの削除

1. [JMS] ノードをクリックして展開します。
2. [ストア] ノードをクリックします。すべての JMS ファイル ストアを示す [JMS ストア] テーブルが右ペインに表示されます。
3. 削除する JDBC ストアの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
4. [はい] をクリックして JDBC ストアを削除します。[ストア] ノードの下の JDBC ストア アイコンが削除されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	JMS JDBC ストアの名前。 この属性は動的にコンフィグレーションできない。	JMS サーバ内でユニークな Java 識別子	MyJMSJDBC Store[-*n*]
[接続プール]	このバッキング ストア用の JDBC 接続プールの名前。 この属性は動的にコンフィグレーションできない。	JDBC 接続プール名またはなし	なし

属性	説明	値の範囲	デフォルト値
[プレフィックス名]	バッキングストア内の JMS テーブルのプレフィックス名。ユニークなプレフィックスを指定して、複数のストアを同じデータベース内に存在させることができる。 プレフィックスは、次の場合にテーブル名の先頭に付加される。 ♦ DBMS が完全修飾名を必要とする場合。 ♦ 2 つの WebLogic Server の JMS テーブルを区別して、複数のテーブルを 1 つの DBMS 上に格納できるようにする必要がある場合。 プレフィックスは次の形式で指定され、有効なテーブル名となるように JMS テーブル名の先頭に付加される。 `[[catalog.]schema.]prefix` **注意：** データが破壊されるため、2 つの JMS ストアで同じデータベースのテーブルを使用することはできない。	JMS テーブルに対して有効なテーブル プレフィックス	なし

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を指定するためのスペースを提供する。	文字列	NULL

47 実行時 JMS プロデューサ

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 送り先
- サーバ
- コンシューマ
- メッセージ数
- 受信メッセージ数
- 現在のバイト数
- 保留バイト数

48 JMS キュー

JMS キューの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS キューおよびトピックの送り先を示す [JMS 送り先] テーブルが右ペインに表示されます。
5. [新しい JMSQueue のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する送り先キューの設定に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、送り先キューのインスタンスを [名前] フィールドで指定した名前で作成します。左ペインの [送り先] ノードに、新しいインスタンスが追加されます。

JMS キューのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての送り先を示す [JMS 送り先] テーブルが右ペインに表示されます。

5. クローンを作成する送り先キューの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、送り先キューのクローンの作成に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、送り先キューのインスタンスを [名前] フィールドで指定した名前で作成します。左ペインの [送り先] ノードに、新しいインスタンスが追加されます。

JMS キューの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべてのセッション プールを示す [JMS 送り先] テーブルが右ペインに表示されます。
5. 削除する送り先キューの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
6. [はい] をクリックして送り先キューを削除します。[送り先] ノードの下のトピック アイコンが削除されます。

すべてのアクティブな JMS の送り先のモニタ

詳細については、「JMS サーバ」の「すべてのアクティブな JMS の送り先のモニタ」を参照してください。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	キューの名前。JNDI 名は別途コンフィグレーションされる。 この属性は動的にコンフィグレーションできない。	JMS サーバ内でユニークな Java 識別子	MyJMSQueue[-*n*]
[JNDI 名]	JNDI ネームスペース内で送り先のルックアップに使用される名前。送り先名は別途コンフィグレーションされる。この属性を指定しない場合、送り先名は JNDI ネームスペースを通して通知されない。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペースのスコープ内でユニークな Java 識別子	NULL

属性	説明	値の範囲	デフォルト値
[ストアを有効化]	キューが JMS サーバで指定されている永続ストアを使用するかどうかを指定するフラグ。 ■ このフラグを有効に設定しても、JMS サーバに対して永続ストアが定義されていない場合は、コンフィグレーションは失敗し、WebLogic JMS は起動しない。 ■ このフラグを無効に設定すると、キューは永続メッセージをサポートしない。 ■ このフラグをデフォルトに設定すると、JMS サーバに対して永続ストアが定義されている場合は、キューは永続ストアを使用し、永続メッセージをサポートする。 この属性は動的にコンフィグレーションできない。	無効、有効、デフォルト	デフォルト
[テンプレート]	キューの派生元の JMS テンプレート。この属性を定義しない場合は、キューの属性を送り先の一部として指定する必要がある。 キューごとの [テンプレート] 属性の設定は静的である。ただし、テンプレートの属性は動的に変更できる。	既存の JMS テンプレート名またはなし	なし
[送り先キー]	キューに到着するメッセージをソートする際のキーとなる、送り先キーのリストを設定する。優先順位の高いものから低いものの順に指定する。JMSMessageID に基づくキーは、最後にのみ指定できる。 **注意：** JMSMessageID がキーに指定されていない場合、暗黙で最後のキーとみなされ、ソート順は「Ascending」(FIFO) に設定される。	リスト	NULL

[しきい値と割当]

属性	説明	値の範囲	デフォルト値
[最大バイト数]	送り先に保存可能な最大バイト数。値 -1 は、送り先に保存可能なバイト数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1、[最大バイトしきい値] 以下	-1

属性	説明	値の範囲	デフォルト値
[最大バイトしきい値]	送り先に保存されているバイト数に基づく上限しきい値。 バイト数がこのしきい値を超えた場合、バイト ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのバイト ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。 **注意 :** [最大バイトしきい値] を -1 にリセットすることでバイト ページングを動的に無効にすることはできない。ページングを無効にするには、[最大バイトしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}ñ1、[最大バイト数] 以下、[最小バイトしきい値] より大	-1

属性	説明	値の範囲	デフォルト値
[最小バイトしきい値]	送り先に保存されているバイト数に基づく下限しきい値。 バイト数がこのしきい値より少なくなると、送り先レベルのバイト ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。	-1、0 ～ 2^{63}ñ1、[最大バイトしきい値] より小	-1

属性	説明	値の範囲	デフォルト値
[バイト ページングを有効化]	送り先でバイト ページングを有効にするかどうかを示すフラグ。 ■ この属性を [無効] に設定すると、この送り先に対する送り先レベルのバイト ページングは明示的に無効になる。 ■ この属性を [有効] に設定すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小バイトしきい値] 属性と [最大バイトしきい値] 属性の値がどちらも -1 より大きい場合は、この送り先に対する送り先レベルのバイト ページングが有効になる。 ■ この属性を [デフォルト] に設定すると、JMS テンプレートが指定されている場合は、JMS テンプレートの値が継承される。送り先に対してテンプレートがコンフィグレーションされていない場合は、[デフォルト] による結果は [無効] と同じである。 **注意：** サーバ レベルのバイト ページングが有効である場合は、特定の送り先に対する送り先レベルのページングが無効であっても、サーバ レベルのページングが開始していると、送り先でメッセージをページングできる。ただし、特定の送り先に対して送り先レベルのページングが無効になっていると、送り先レベルの最大しきい値の設定によって、しきい値を超えたときにメッセージが強制的にページ アウトされることはない。	[String] [有効] [無効] [デフォルト]	[デフォルト]

属性	説明	値の範囲	デフォルト値
[最大メッセージ数]	送り先に保存可能なメッセージの最大数。値 -1 は送り先に保存可能なメッセージ数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}ñ1、[最大メッセージしきい値] 以下	-1
[最大メッセージしきい値]	送り先に保存されるメッセージ数に基づく上限しきい値。 メッセージ数がこのしきい値を超えた場合、メッセージ ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。 **注意 :** [最大メッセージしきい値] を -1 にリセットすることでメッセージ ページングを動的に無効にすることはできない。ページングを無効にするには、[最大メッセージしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}ñ1、[最大メッセージ数] 以下、[最小メッセージしきい値] より大	-1

属性	説明	値の範囲	デフォルト値
[最小メッセージしきい値]	送り先に保存されるメッセージ数に基づく下限しきい値。 メッセージ数がこのしきい値より少なくなると、送り先レベルのメッセージ ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}ñ1、[最大メッセージしきい値] より小	-1

属性	説明	値の範囲	デフォルト値
[メッセージ ページングを有効化]	送り先でメッセージ ページングを有効にするかどうかを示すフラグ。 ■ この属性を [無効] に設定すると、この送り先に対する送り先レベルのメッセージ ページングは明示的に無効になる。 ■ この属性を [有効] に設定すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小メッセージしきい値] 属性と [最大メッセージしきい値] 属性の値がどちらも -1 より大きい場合は、この送り先に対する送り先レベルのメッセージ ページングが有効になる。 ■ この属性を [デフォルト] に設定すると、JMS テンプレートが指定されている場合は、JMS テンプレートの値が継承される。送り先に対してテンプレートがコンフィグレーションされていない場合は、[デフォルト] による結果は [無効] と同じである。 **注意 :** サーバ レベルのメッセージ ページングが有効である場合は、特定の送り先に対する送り先レベルのページングが無効であっても、サーバ レベルのページングが開始していると、送り先でのメッセージをページングできる。ただし、特定の送り先に対して送り先レベルのページングが無効になっていると、送り先レベルの最大しきい値の設定によって、しきい値を超えたときにメッセージが強制的にページアウトされることはない。	[String] [有効] [無効] [デフォルト]	[デフォルト]

[オーバライド]

属性	説明	値の範囲	デフォルト値
[優先順位オーバライド]	メッセージ プロデューサによって指定された優先度に関わりなく、送り先に届くすべてのメッセージに割り当てられる優先度。 デフォルト値 (-1) は送り先が優先順位の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 9	-1
[生存時間オーバライド]	メッセージ プロデューサによって指定された生存時間に関わりなく、送り先に届くすべてのメッセージに割り当てられる生存時間。 デフォルト値 (-1) は送り先が生存時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1	-1

属性	説明	値の範囲	デフォルト値
[送信時間オーバーライド]	メッセージが生成されてから送り先で表示できるようになるまでのデフォルトの遅延時間（ミリ秒単位）を、プロデューサまたは接続ファクトリによって指定された送信時間に関わらず定義する。 デフォルト値 (-1) は送り先が送信時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1、または配信スケジュールの文字列構文	-1
[配信モードのオーバライド]	メッセージ プロデューサによって指定された配信モードに関わりなく、送り先に届くすべてのメッセージに割り当てられる配信モード。 デフォルトの [配信しない] は配信モードがオーバーライドされないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	[永続]、[非永続]、または [配信しない]	[配信しない]

[モニタ]

[すべてのアクティブな JMS の送り先のモニタ]

属性	説明	値の範囲	デフォルト値
[送り先]	送り先名。		なし
[サーバ]	関連するサーバ名。		なし
[コンシューマ数]	現在の登録済みメッセージ コンシューマの数。		なし
[最大コンシューマ数]	任意の時点における登録済みメッセージ コンシューマの最大数。		なし
[コンシューマ総数]	登録済みメッセージ コンシューマの総数。		なし
[現在のバイト数]	現在保存されているバイト数。		なし
[保留バイト数]	保存されていて、確認応答およびコミットが行われていないトランザクション バイト数。		なし
[受信バイト数]	受信済みバイト数。		なし
[バイトしきい値の時間]	サーバの起動以降、送り先がバイトしきい値状態になっていた時間の合計。		なし

属性	説明	値の範囲	デフォルト値
[メッセージ数]	この送り先に現在格納されているメッセージの数。		なし
[最大メッセージ数]	ある時点において格納されるメッセージの最大数。		なし
[保留メッセージ数]	保存されていて、確認応答およびコミットが行われていないトランザクション メッセージの数。		なし
[受信メッセージ数]	受信済みメッセージ数。		なし
[メッセージしきい値の時間]	送り先がメッセージしきい値状態になっていた時間の合計。		なし

JMS キューのモニタの詳細については、「すべてのアクティブな JMS サービスのモニタ」を参照してください。

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字の文字列	NULL

49 実行時 JMS

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- 接続数
- 最大接続数
- 総接続数
- サーバ数
- 最大サーバ数
- 総サーバ数

50 JMS サーバ

JMS サーバの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。
3. [新しい JMS Server のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するサーバの設定に関連するタブが示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドに指定した名前でサーバ インスタンスを作成します。新しいインスタンスが左ペインの [サーバ] ノードの下に追加されます。[送り先] ノードと [セッション プール] ノードが新しいサーバ インスタンスの下にデフォルトで自動的に追加されます。
6. WebLogic Server に JMS サーバを割り当てます。

詳細については、「JMS サーバの割り当て」を参照してください。

JMS サーバのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。

3. クローンを作成するサーバの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、サーバのクローンの作成に関連するタブが示されます。
4. 属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドに指定した名前でサーバ インスタンスを作成します。新しいインスタンスが左ペインの [サーバ] ノードの下に追加されます。[送り先] ノードと [セッション プール] ノードが新しいサーバ インスタンスの下にデフォルトで自動的に追加されます。
6. WebLogic Server に JMS サーバを割り当てます。

詳細については、「JMS サーバの割り当て」を参照してください。

JMS サーバの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。
3. 削除するサーバの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
4. [はい] をクリックして、サーバを削除します。[サーバ] ノードの下にあるサーバアイコンが削除されます。

すべてのアクティブな JMS サービスのモニタ

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。

3. [すべてのアクティブ JMS サービスのモニタ] テキスト リンクをクリックします。アクティブな JMS サービスを示すダイアログが右ペインに表示されます。

JMS サーバのすべてのインスタンスのモニタ

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。
3. モニタするサーバの行の [すべてのインスタンスのモニタ] テキストリンクをクリックします。右ペインにダイアログが表示され、サーバ ドメイン全体でデプロイされているサーバのすべてのインスタンスが示されます。

すべてのアクティブな JMS の送り先のモニタ

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。
3. [すべてのアクティブな JMS の送り先のモニタ] テキスト リンクをクリックします。現在のドメインでアクティブな JMS の送り先を示すダイアログが右ペインに表示されます。

すべてのアクティブな JMS セッション プールのモニタ

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックします。ドメインで定義されているすべてのサーバを示す [JMS サーバ] テーブルが右ペインに表示されます。
3. [すべての JMSSession Pool Runtimes のモニタ] テキスト リンクをクリックします。現在のドメインでアクティブな JMS セッション プールを示すダイアログが右ペインに表示されます。

JMS サーバの割り当て

1. 左ペインで、サーバに割り当てる [サーバ] の下のインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] タブと [クラスタ] タブについて次の手順を実行します。
   a. データ ソースに割り当てる 1 つまたは複数のターゲットを [選択可] カラムで選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	JMS サーバの名前。 この属性は動的にコンフィグレーションできない。	特定のクラスタ内でユニークな Java 識別子	MyJMS Server[-*n*]
[ストア]	JMS サーバの既存の永続ストア。1 つの永続ストアは 1 つの JMS サーバでのみ使用できる。 値「なし」は永続メッセージがサポートされていないことを示す。永続ストアが指定されていない場合、このサーバの送り先は永続メッセージまたは恒久サブスクライバをサポートしない。 この属性は動的にコンフィグレーションできない。	既存の永続ストア名またはなし	なし

属性	説明	値の範囲	デフォルト値
[Paging Store]	JMS サーバのために非永続メッセージをページングする永続ストアの名前。ページング ストアは、永続メッセージまたは恒久サブスクライバに対して使用するものと同じストアであってはならない。1 つの永続ストアは 1 つの JMS サーバでのみ使用できる。 値「なし」はメッセージ ページングがサポートされていないことを示す。ページング ストアを指定しないと、サーバおよび送り先はメッセージ ページングをサポートしない。 **注意：** ページング用に JDBC ストアを使用することは推奨されない。トラフィックが多くパフォーマンスが低下することから、このようなコンフィグレーションは望ましくない。	既存の永続ストアまたはなし	なし
[一時的なテンプレート]	一時的キューおよび一時的トピックを含むすべての一時的送り先を作成するために使用する既存の JMS テンプレートの名前。一時的送り先の属性値は、この JMS テンプレートから派生する。 テンプレートの一部として指定した場合、一時的送り先は永続的なメッセージングをサポートしていないので、[ストア] 属性値は無視される。 この属性は動的にコンフィグレーションできる。 **注意：** この属性を「なし」に設定した場合、一時的送り先（キューまたはトピック）の作成は失敗する。	JMS テンプレート名またはなし	なし （送り先の属性のデフォルト値が使用される）

[しきい値と割当]

属性	説明	値の範囲	デフォルト値
[最大バイト数]	JMS サーバに保存可能な最大バイト数。値 -1 は JMS サーバに保存可能なバイト数が制限されていないことを示す。 この属性は動的にコンフィグレーションできる。	-1、0 ～ 2^{63}-1、[最大バイトしきい値] 以上	-1

属性	説明	値の範囲	デフォルト値
[最大バイトしきい値]	JMS サーバに保存されているバイト数に基づく上限しきい値。 バイト数がこのしきい値を超えた場合、バイト ページングが有効になっていて、ページング ストアがコンフィグレーションされていると、サーバ バイト ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。 値 -1 は、JMS サーバに対してサーバ バイトページングおよびしきい値ログ メッセージが無効であることを示す。 この属性は動的にコンフィグレーションできる。 **注意：** [最大バイトしきい値] を -1 にリセットすることでバイト ページングを動的に無効にすることはできない。ページングを無効にするには、[最大バイトしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}-1、[最大バイト数] 以下、[最小バイトしきい値] より大	-1

属性	説明	値の範囲	デフォルト値
[最小バイトしきい値]	JMS サーバに保存されているバイト数に基づく下限しきい値。 バイト数がこのしきい値より少なくなると、サーバ バイト ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。 値 -1 は、JMS サーバに対してサーバ ページングおよびしきい値ログ メッセージが無効であることを示す。 この属性は動的にコンフィグレーションできる。	-1、0 ～ 2^{63}-1、[最大バイトしきい値] より小	-1
[Bytes Paging Enabled]	JMS サーバでバイト ページングを有効にするかどうかを示すフラグ。 ■ このフラグを選択しないと、サーバ バイト ページングは明示的に無効になる。 ■ このフラグを選択すると、ページング ストアがコンフィグレーションされていて、[最小バイトしきい値] 属性と [最大バイトしきい値] 属性の値がどちらも -1 より大きい場合は、サーバ バイト ページングが有効になる。 ■ [最小バイトしきい値] 属性と [最大バイトしきい値] 属性のどちらかが定義されていない場合、または -1 と定義されている場合は、このフラグを選択しても、サーバ バイト ページングは暗黙で無効になる。	ブール値 有効 = 選択されている 無効 = 選択されていない	選択されていない

属性	説明	値の範囲	デフォルト値
[最大メッセージ数]	JMS サーバに保存可能なメッセージの最大数。値 -1 は JMS サーバに保存可能なメッセージ数が制限されていないことを示す。 この属性は動的にコンフィグレーションできる。	-1、0 ～ 2^{63}-1、[最大メッセージしきい値] 以上	-1
[最大メッセージしきい値]	JMS サーバに保存されるメッセージ数に基づく上限しきい値。 メッセージ数がこのしきい値を超えた場合、メッセージ ページングが有効になっていて、ページング ストアがコンフィグレーションされていると、サーバ メッセージ ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。 値 -1 は、JMS サーバに対してサーバ ページングおよびしきい値ログ メッセージが無効であることを示す。 この属性は動的にコンフィグレーションできる。 **注意：** [最大メッセージしきい値] を -1 にリセットすることでメッセージ ページングを動的に無効にすることはできない。ページングを無効にするには、[最大メッセージしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}-1、[最大メッセージ数] 以下、[最小メッセージしきい値] より大	-1

属性	説明	値の範囲	デフォルト値
[最小メッセージしきい値]	JMS サーバに保存されるメッセージ数に基づく下限しきい値。 メッセージ数がこのしきい値より少なくなると、サーバ メッセージ ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。 値 -1 は、JMS サーバに対してサーバ ページングおよびしきい値ログ メッセージが無効であることを示す。 この属性は動的にコンフィグレーションできる。 この属性は動的にコンフィグレーションできる。	-1、0 ～ 2^{63}-1、[最大メッセージしきい値] より小	-1
[Messages Paging Enabled]	JMS サーバでメッセージ ページングを有効にするかどうかを示すフラグ。 ■ このフラグを選択しないと、サーバ レベルのメッセージ ページングは明示的に無効になる。 ■ このフラグを選択すると、ページング ストアがコンフィグレーションされていて、[最小メッセージしきい値] 属性と [最大メッセージしきい値] 属性の値がどちらも -1 より大きい場合は、サーバ レベルのメッセージ ページングが有効になる。 ■ [最小メッセージしきい値] 属性と [最大メッセージしきい値] 属性のどちらかが定義されていない場合、または -1 と定義されている場合は、このフラグを選択しても、サーバ レベルのメッセージ ページングは暗黙で無効になる。	ブール値 有効 = 選択されている 無効 = 選択されていない	選択されていない

[モニタ]

[すべてのアクティブ JMS サービスのモニタ]

属性	説明	値の範囲	デフォルト値
[サーバ]	サービス名。	なし	なし
[接続]	現在の接続数。	なし	なし
[最大接続数]	特定の時点における接続の最大数。	なし	なし
[接続総数]	接続の総数。	なし	なし
[サーバ]	現在のサーバ数。	なし	なし
[最大サーバ数]	特定の時点におけるサーバの最大数。	なし	なし
[サーバ総数]	サーバの総数。	なし	なし

[すべての JMS サーバのインスタンスのモニタ]

属性	説明	値の範囲	デフォルト値
[名前]	JMS サーバの名前。		なし
[送り先]	送り先のリスト。		なし
[最大送り先数]	任意の時点におけるインスタンス化された送り先の最大数。		なし
[送り先総数]	インスタンス化された送り先の総数。		なし
[現在のバイト数]	JMS サーバのすべての送り先全体に保存されている現在のバイト数。		なし
[保留バイト数]	コミットされていないトランザクションの結果または確認応答の結果が保留中の送受信バイト数。		なし
[受信バイト数]	すべての送り先向けに受信されたバイト総数。		なし
[バイトしきい値の時間]	バイトしきい値状態になっていた時間。		なし
[現在のメッセージ数]	すべての送り先全体で現在保存されているメッセージの数。		なし

[最大メッセージ数]	任意の時点においてすべての送り先全体で保存されているメッセージの最大数。	なし
[受信メッセージ数]	すべての送り先向けに受信されたメッセージ総数。	なし
[メッセージしきい値の時間]	メッセージしきい値状態になっていた時間。	なし
[合計メッセージ数]	すべての送り先全体で保存されているメッセージの総数。	なし
[セッション プール数]	JMS サーバ用に定義されたセッション プールのリスト。	なし
[最大セッション プール数]	任意の時点におけるセッション プールの最大数。	なし
[セッション プール総数]	インスタンス化されたセッション プールの総数。	なし

[送り先のモニタ]

属性	説明	値の範囲	デフォルト値
[送り先]	送り先名。		なし
[サーバ]	関連するサーバ名。		なし
[コンシューマ数]	現在の登録済みメッセージ コンシューマの数。		なし
[最大コンシューマ数]	任意の時点における登録済みメッセージ コンシューマの最大数。		なし
[コンシューマ総数]	登録済みメッセージ コンシューマの総数。		なし
[現在のバイト数]	現在保存されているバイト数。		なし
[保留バイト数]	保存されていて、確認応答およびコミットが行われていないトランザクション バイト数。		なし
[受信バイト数]	受信済みバイト数。		なし
[バイトしきい値の時間]	サーバの起動以降、送り先がバイトしきい値状態になっていた時間の合計。		なし
[メッセージ数]	この送り先に現在格納されているメッセージの数。		なし

属性	説明	値の範囲	デフォルト値
[最大メッセージ数]	ある時点において格納されるメッセージの最大数。		なし
[保留メッセージ数]	保存されていて、確認応答およびコミットが行われていないトランザクション メッセージの数。		なし
[受信メッセージ数]	受信済みメッセージ数。		なし
[メッセージしきい値の時間]	送り先がメッセージしきい値状態になっていた時間の合計。		なし

[すべてのアクティブ JMS セッション プールのモニタ]

属性	説明	値の範囲	デフォルト値
[JMS Session Pool]	セッション プールの名前。		
[Server]	関連するサーバの名前。		
[コンシューマ]	コンシューマのリスト。		
[最大コンシューマ数]	任意の時点におけるセッション プールのコンシューマの最大数。		
[コンシューマ総数]	コンシューマの総数。		

[対象]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	選択可能な対象のリストを提供する。現在、JMS サーバでは最大で 1 つの対象を選択できる。	リスト	対象が選択されていない状態

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字の文字列	NULL

51 実行時 JMS サーバ

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- セッション プール数
- 送り先
- メッセージ数
- 合計メッセージ数
- 現在のバイト数
- 保留バイト数

52 JMS セッション プール

JMS セッション プールの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [セッション プール] ノードをクリックします。すべてのセッション プールを示す [JMS セッション プール] テーブルが右ペインに表示されます。
5. [新しい JMS Session Pool のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するセッション プールの設定に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、[名前] フィールドで指定した名前のセッション プール インスタンスを作成します。新しいインスタンスが左ペインの [セッション プール] ノードの下に追加されます。

JMS セッション プールのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [セッション プール] ノードをクリックします。すべてのセッション プールを示す [JMS セッション プール] テーブルが右ペインに表示されます。

5. クローンを作成するセッション プールの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、セッション プールのクローンの作成に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、[名前] フィールドで指定した名前のセッション プール インスタンスを作成します。新しいインスタンスが左ペインの [セッション プール] ノードの下に追加されます。

JMS セッション プールの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [セッション プール] ノードをクリックします。すべてのセッション プールを示す [JMS セッション プール] テーブルが右ペインに表示されます。
5. 削除するセッション プールの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
6. [はい] をクリックしてセッション プールを削除します。[セッション プール] ノードの下のセッション プール アイコンが削除されます。

すべてのアクティブな JMS セッション プールのモニタ

詳細については、「JMS サーバ」の「すべてのアクティブな JMS セッション プールのモニタ」を参照してください。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	セッション プールの名前。 この属性は動的にコンフィグレーションできない。	特定のクラスタ内でユニークな Java 識別子	MyJMS Session Pool[-*n*]
[接続ファクトリ]	セッション プールに関連する接続ファクトリ。	既存の接続プール名またはなし	なし
[リスナ クラス]	`onMessage()` メソッドを提供するメッセージ リスナ クラスの名前。このメソッドにより、メッセージが処理される。 この属性は動的にコンフィグレーションできない。	既存のメッセージ リスナ クラス名または CLASSPATH 内の Java クラス	なし

属性	説明	値の範囲	デフォルト値
[確認応答モード]	セッション プール内の非トランザクション セッションによって使用される通知モード。 トランザクション セッションの場合、メッセージはセッションがコミットされるときに自動的に通知されるので、このフィールドは無視される。 通知モードの詳細については、『WebLogic JMS プログラマーズ ガイド』を参照。 この属性は動的にコンフィグレーションできない。	[Auto]、[Client]、[Dups OK]、および [None]	Auto
[最大セッション]	セッション プール内の並行セッションの数。値 -1 は最大数が設定されていないことを示す。 この属性は動的にコンフィグレーションできるが、変更はセッション プールが再起動されるまで有効にならない。	-1、1 ～ 2^{63}ñ 1	-1
[処理済]	セッション プールがトランザクション セッションを作成するかどうかを指定するフラグ。 この属性は動的にコンフィグレーションできない。	有効 = 選択 無効 = 未選択	選択されていない

[モニタ]

[すべての JMSSession Pool Runtimes のモニタ]

属性	説明	値の範囲	デフォルト値
[JMS Session Pool]	セッション プールの名前。		なし
[Server]	関連するサーバの名前。		なし
[コンシューマ]	コンシューマのリスト。		なし
[最大コンシューマ数]	セッション プールのコンシューマの最大数。		なし
[コンシューマ総数]	コンシューマの総数。		なし

JMS セッション プールのモニタの詳細については、「すべてのアクティブな JMS セッション プールのモニタ 」を参照してください。

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

53 実行時 JMS セッション プール

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- セッション プール
- サーバ
- コンシューマ
- 最大コンシューマ数
- コンシューマ総数

54 JMS ストア

JMS ストアの情報は次のファイルに入っています。

- JMS ファイル ストア
- JMS JDBC ストア

55 JMS テンプレート

JMS テンプレートの作成

1. 左ペインの [JMS] ノードをクリックします。
2. [テンプレート] ノードをクリックします。右ペインに [JMS テンプレート] テーブルが表示され、ドメインで定義されているすべてのテンプレートが示されます。
3. [新しい JMS Template のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するテンプレートの設定に関連するタブが示されます。
4. [名前] 属性フィールドに値を入力します。
5. [作成] をクリックして、[名前] フィールドで指定した名前のテンプレートのインスタンスを作成します。新しいインスタンスが左ペインの [テンプレート] ノードの下に追加されます。
6. そのほかのタブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
7. [適用] をクリックして、変更を保存します。

JMS テンプレートのクローンの作成

1. 左ペインの [JMS] ノードをクリックします。
2. [テンプレート] ノードをクリックします。右ペインに [JMS テンプレート] テーブルが表示され、ドメインで定義されているすべてのテンプレートが示されます。

3. クローンを作成するテンプレートの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、新しいテンプレートのクローンの作成に関連するタブが示されます。
4. [名前] 属性フィールドに値を入力します。
5. 右下隅の [作成] ボタンをクリックして、テンプレートのインスタンスを [名前] フィールドで指定した名前で作成します。新しいインスタンスが左ペインの [テンプレート] ノードの下に追加されます。
6. そのほかのタブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
7. [適用] をクリックして、変更を保存します。

JMS テンプレートの削除

1. 左ペインの [JMS] ノードをクリックします。
2. [テンプレート] ノードをクリックします。右ペインに [JMS テンプレート] テーブルが表示され、ドメインで定義されているすべてのテンプレートが示されます。
3. 削除するサーバの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
4. [はい] をクリックして、テンプレートを削除します。[テンプレート] ノードの下のテンプレート アイコンが削除されます。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	JMS テンプレートの名前。 この属性は動的にコンフィグレーションできない。	特定のクラスタ内でユニークな Java 識別子	MyJMS Template[-*n*]
[送り先キー]	送り先に届いた時にメッセージを並べ替える場合に、選択可能な送り先キーのリストを提供する。リストは、重要度の高いキーから低いキーへの順に並べられる。 JMSMessageID に基づくキーは、最後のキーとしてのみ使用される。 **注意：** キーで JMSMessageID が定義されていない場合は、暗黙的に最後のキーとみなされ、ソート順に「昇順」（先入れ、先出し）が設定される。	リスト	NULL

[しきい値と割当]

属性	説明	値の範囲	デフォルト値
[最大バイト]	送り先に保存可能な最大バイト数。値 -1 は送り先に保存可能なバイト数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大バイトしきい値] 以上	-1

属性	説明	値の範囲	デフォルト値
[最大バイトしきい値]	送り先に保存されているバイト数に基づいく上限しきい値。 バイト数がこのしきい値を超えた場合、バイト ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのバイト ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。 **注意 :** 最大しきい値を -1 にリセットすることでバイト ページングを動的に無効にすることはできない。ページングを無効にするには、最大しきい値に非常に大きな値（最大 2^{63} -1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}-1、[最大バイト] 以下、[最小バイトしきい値] より大	-1

属性	説明	値の範囲	デフォルト値
[最小バイトしきい値]	送り先に保存されているバイト数に基づく下限しきい値。 バイト数がこのしきい値より少なくなると、送り先レベルのバイト ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大バイトしきい値] より小	-1

属性	説明	値の範囲	デフォルト値
[Bytes Paging Enabled]	送り先でバイト ページングを有効にするかどうかを示すフラグ。 ■ このフラグを選択しないと、送り先の設定でテンプレートがオーバライドされていない限り、テンプレートの送り先に対してバイト ページングは無効になる。 ■ このフラグを選択すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小バイトしきい値] 属性と [最大バイトしきい値] 属性の値がどちらも -1 より大きい場合は、送り先の設定でテンプレートがオーバライドされていない限り、テンプレートの送り先に対してバイト ページングが有効になる。 **注意 :** JMS Template MBean で値が定義されていない場合、デフォルトは False であり、送り先の設定でテンプレートをオーバライドしていない限り、テンプレートの送り先に対するバイト ページングは無効になる。	ブール値 有効 = 選択されている 無効 = 選択されていない	選択されていない
[最大メッセージ]	送り先に保存可能なメッセージの最大数。値 -1 は送り先に保存可能なメッセージ数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大メッセージしきい値] 以上	-1

属性	説明	値の範囲	デフォルト値
[最大メッセージしきい値]	送り先に保存されるメッセージ数に基づく上限しきい値。 メッセージ数がこのしきい値を超えた場合、メッセージ ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのメッセージ ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。 **注意 :** 最大しきい値を -1 にリセットすることでメッセージ ページングを動的に無効にすることはできない。ページングを無効にするには、最大しきい値に非常に大きな値（最大 2^{63} -1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}-1、[最大メッセージ] 以下、[最小メッセージしきい値] より大	-1

属性	説明	値の範囲	デフォルト値
[最小メッセージしきい値]	送り先に保存されるメッセージ数に基づく下限しきい値。 メッセージ数がこのしきい値より少なくなると、送り先レベルのメッセージ ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、[最大メッセージしきい値] より小	-1

属性	説明	値の範囲	デフォルト値
[Messages Paging Enabled]	送り先でメッセージ ページングを有効にするかどうかを示すフラグ。 ■ このフラグを選択しないと、送り先の設定でテンプレートがオーバライドされていない限り、テンプレートの送り先に対してメッセージ ページングは無効になる。 ■ このフラグを選択すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小メッセージしきい値] 属性と [最大メッセージしきい値] 属性の値がどちらも -1 より大きい場合は、送り先の設定でテンプレートがオーバライドされていない限り、テンプレートの送り先に対してメッセージ ページングが有効になる。 **注意：** JMS Template MBean で値が定義されていない場合、デフォルトは False であり、送り先の設定でテンプレートをオーバライドしていない限り、テンプレートの送り先に対するメッセージ ページングは無効になる。	ブール値 有効 = 選択されている 無効 = 選択されていない	選択されていない

[オーバライド]

属性	説明	値の範囲	デフォルト値
[優先順位オーバライド]	メッセージ プロデューサによって指定された優先度に関わりなく、送り先に届くすべてのメッセージに割り当てられる優先度。 デフォルト値 (-1) は送り先が優先度の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 9	-1
[生存時間オーバライド]	メッセージ プロデューサによって指定された生存時間に関わりなく、送り先に届くすべてのメッセージに割り当てられる生存時間の値。 デフォルト値 (-1) は送り先が生存時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1	-1

属性	説明	値の範囲	デフォルト値
[送信時間オーバーライド]	プロデューサまたは接続ファクトリの指定した送信時間に関わりなく、メッセージが生成されてから送り先で表示できるようになるまでのデフォルトの遅延時間（ミリ秒単位）を定義する。 デフォルト値 (-1) は送り先が送信時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1、または配信スケジュールの文字列構文	-1
[配信モードのオーバライド]	メッセージ プロデューサによって指定された配信モードに関わりなく、送り先に届くすべてのメッセージに割り当てられる配信モード。 [配信しない] は配信モードがオーバーライドされないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	[永続]、[非永続]、または [配信しない]	[配信しない]

[再配信]

属性	説明	値の範囲	デフォルト
[再配信遅延のオーバライド]	コンシューマや接続ファクトリによって指定された再配信遅延に関わりなく、ロール バックまたは回復されたメッセージが再配信されるまでの遅延をミリ秒単位で定義する。 デフォルト値 (-1) は送り先が再配信遅延の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1	-1

属性	説明	値の範囲	デフォルト
[再配信の制限]	メッセージがエラー送り先に置かれるまでに試行できる再配信の回数。この回数の上限に達した場合、エラー送り先が設定されているかどうかによって、以下の動作が発生する。 ■ エラー送り先がコンフィグレーションされていない場合、またはエラー送り先の割り当てを超過した場合は、永続メッセージも非永続メッセージも、そのまま削除される。 ■ エラー送り先がコンフィグレーションされていて、割り当てを超過していない場合は、メッセージのログが記録され、メッセージは削除される。ただし、永続メッセージは、永続ストアに残る。このため、WebLogic Server が再起動した時、永続メッセージは確実に再配信される。 デフォルト値 (-1) は送り先が再配信制限の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}-1	-1
[エラー送り先]	再配信の上限に達したメッセージの送り先。エラー送り先が NULL の場合は、このメッセージはそのまま削除される。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	既存の送り先またはなし	なし

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字の文字列	NULL

56 JMS トピック

JMS トピックの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての JMS トピックおよびキューの送り先を示す [JMS 送り先] テーブルが右ペインに表示されます。
5. [新しい JMSTopic のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しい送り先トピックの設定に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、送り先トピックのインスタンスを [名前] フィールドで指定した名前で作成します。左ペインの [送り先] ノードに、新しいインスタンスが追加されます。

JMS トピックのクローンの作成

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての送り先を示す [JMS 送り先] テーブルが右ペインに表示されます。

5. クローンを作成する送り先トピックの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、送り先トピックのクローンの作成に関連するタブが示されます。
6. 属性フィールドに値を入力します。
7. [作成] をクリックして、送り先トピックのインスタンスを [名前] フィールドで指定した名前で作成します。左ペインの [送り先] ノードに、新しいインスタンスが追加されます。

JMS トピックの削除

1. [JMS] ノードをクリックして展開します。
2. [サーバ] ノードをクリックして展開します。
3. [サーバ] の下のサーバ インスタンスをクリックして展開します。
4. [送り先] ノードをクリックします。すべての送り先を示す [JMS 送り先] テーブルが右ペインに表示されます。
5. 削除する送り先トピックの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
6. [はい] をクリックして送り先トピックを削除します。[送り先] ノードの下のトピック アイコンが削除されます。

すべてのアクティブな JMS の送り先のモニタ

詳細については、「JMS サーバ」の「すべてのアクティブな JMS の送り先のモニタ」を参照してください。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト
[名前]	トピックの名前。JNDI 名は別途コンフィグレーションされる。 この属性は動的にコンフィグレーションできない。	JMS サーバ内でユニークな Java 識別子	MyJMS Destination[-*n*]
[JNDI 名]	JNDI ネームスペース内で送り先のルックアップに使用される名前。送り先名は別途コンフィグレーションされる。この属性を指定しない場合、送り先名は JNDI ネームスペースを通して通知されない。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペースのスコープ内でユニークな Java 識別子	NULL
[送り先キー]	送り先に到着するメッセージをソートする際のキーとなる、送り先キーのリストを設定する。優先順位の高いものから低いものの順に指定する。JMSMessageID に基づくキーは、最後にのみ指定できる。 **注意：** JMSMessageID がキーに指定されていない場合、暗黙で最後のキーとみなされ、ソート順は「Ascending」(FIFO) に設定される。	リスト	NULL

属性	説明	値の範囲	デフォルト
[ストアを有効化]	送り先が JMS サーバで指定されている永続ストアを使用するかどうかを指定するフラグ。 ■ このフラグを有効に設定しても、JMS サーバに対して永続ストアが定義されていない場合は、コンフィグレーションは失敗し、WebLogic JMS は起動しない。 ■ このフラグを無効に設定すると、送り先は永続メッセージをサポートしない。 ■ このフラグをデフォルトに設定すると、JMS サーバに対して永続ストアが定義されている場合は、送り先は永続ストアを使用し、永続メッセージをサポートする。 この属性は動的にコンフィグレーションできない。	無効、有効、デフォルト	デフォルト
[テンプレート]	送り先の派生元の JMS テンプレート。この属性を定義しない場合は、キューの属性を送り先の一部として指定する必要がある。 送り先ごとの [テンプレート] 属性の設定は静的である。ただし、テンプレートの属性は動的に変更できる。	既存の JMS テンプレート名またはなし	なし

[しきい値と割当]

属性	説明	値の範囲	デフォルト
[最大バイト数]	送り先に保存可能な最大バイト数。値 -1 は、送り先に保存可能なバイト数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1、[最大バイトしきい値] 以上	-1

属性	説明	値の範囲	デフォルト
[最大バイトしきい値]	送り先に保存されているバイト数に基づく上限しきい値。 バイト数がこのしきい値を超えた場合、バイト ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのバイト ページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。 **注意 :** [最大バイトしきい値] を -1 にリセットすることでバイト ページングを動的に無効にすることはできない。ページングを無効にするには、[最大バイトしきい値] に非常に大きな値（最大 2^{63}ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2^{63}ñ1、[最大バイト数] 以下、[最小バイトしきい値] より大	-1

属性	説明	値の範囲	デフォルト
[最小バイトしきい値]	送り先に保存されているバイト数に基づく下限しきい値。 バイト数がこのしきい値より少なくなると、送り先レベルのバイト ページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先バイト ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、すでに保存されているメッセージには影響しない。	-1、0 ～ 2^{63}ñ1、[最大バイトしきい値] より小	-1

属性	説明	値の範囲	デフォルト
[バイト ページングを有効化]	送り先でバイト ページングを有効にするかどうかを示すフラグ。 ■ この属性を [無効] に設定すると、この送り先に対する送り先レベルのバイト ページングは明示的に無効になる。 ■ この属性を [有効] に設定すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小バイトしきい値] 属性と [最大バイトしきい値] 属性の値がどちらも -1 より大きい場合は、この送り先に対する送り先レベルのバイト ページングが有効になる。 ■ この属性を [デフォルト] に設定すると、JMS テンプレートが指定されている場合は、JMS テンプレートの値が継承される。送り先に対してテンプレートがコンフィグレーションされていない場合は、[デフォルト] による結果は [無効] と同じである。 **注意：** サーバ レベルのバイト ページングが有効である場合は、特定の送り先に対する送り先レベルのページングが無効であっても、サーバ レベルのページングが開始していると、送り先でメッセージをページングできる。ただし、特定の送り先に対して送り先レベルのページングが無効になっていると、送り先レベルの最大しきい値の設定によって、しきい値を超えたときにメッセージが強制的にページ アウトされることはない。	[String] [有効] [無効] [デフォルト]	[デフォルト]

属性	説明	値の範囲	デフォルト
[最大メッセージ数]	送り先に保存可能なメッセージの最大数。値 -1 は送り先に保存可能なメッセージ数が制限されていないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2⁶³ñ1、[最大メッセージしきい値] 以上	-1
[最大メッセージしきい値]	送り先に保存されるメッセージ数に基づく上限しきい値。 メッセージ数がこのしきい値を超えた場合、メッセージ ページングが有効になっていて、JMS サーバに対してページング ストアがコンフィグレーションされていると、送り先レベルのページングが開始され、最大しきい値状態を示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。 **注意：** [最大メッセージしきい値] を -1 にリセットすることでメッセージ ページングを動的に無効にすることはできない。ページングを無効にするには、[最大メッセージしきい値] に非常に大きな値（最大 2⁶³ñ1）を設定し、ページングが実行されないようにする。	-1、0 ～ 2⁶³ñ1、[最大メッセージ数] 以下、[最小メッセージしきい値] より大	-1

属性	説明	値の範囲	デフォルト
[最小メッセージしきい値]	送り先に保存されるメッセージ数に基づく下限しきい値。 メッセージ数がこのしきい値より少なくなると、送り先レベルのメッセージページングは停止し（ページングが行われている場合）、しきい値条件が解消されたことを示すログ メッセージがサーバに記録される。値 -1 は、送り先メッセージ ページングおよびしきい値ログ メッセージがその送り先に対しては無効であることを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63}ñ1、[最大メッセージしきい値] より小	-1

属性	説明	値の範囲	デフォルト
[メッセージ ページングを有効化]	送り先でメッセージ ページングを有効にするかどうかを示すフラグ。 ■ この属性を [無効] に設定すると、この送り先に対する送り先レベルのメッセージ ページングは明示的に無効になる。 ■ この属性を [有効] に設定すると、JMS サーバに対してページング ストアがコンフィグレーションされていて、[最小メッセージしきい値] 属性と [最大メッセージしきい値] 属性の値がどちらも -1 より大きい場合は、この送り先に対する送り先レベルのメッセージ ページングが有効になる。 ■ この属性を [デフォルト] に設定すると、JMS テンプレートが指定されている場合は、JMS テンプレートの値が継承される。送り先に対してテンプレートがコンフィグレーションされていない場合は、[デフォルト] による結果は [無効] と同じである。 **注意：** サーバ レベルのメッセージ ページングが有効である場合は、特定の送り先に対する送り先レベルのページングが無効であっても、サーバ レベルのページングが開始していると、送り先でのメッセージをページングできる。ただし、特定の送り先に対して送り先レベルのページングが無効になっていると、送り先レベルの最大しきい値の設定によって、しきい値を超えたときにメッセージが強制的にページアウトされることはない。	[String] [有効] [無効] [デフォルト]	[デフォルト]

[オーバライド]

属性	説明	値の範囲	デフォルト
[優先順位オーバライド]	メッセージ プロデューサによって指定された優先度に関わりなく、送り先に届くすべてのメッセージに割り当てられる優先度。 デフォルト値 (-1) は送り先が優先度の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 9	-1
[生存時間オーバライド]	メッセージ プロデューサによって指定された生存時間に関わりなく、送り先に届くすべてのメッセージに割り当てられる生存時間。 デフォルト値 (-1) は送り先が生存時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1	-1

属性	説明	値の範囲	デフォルト
[送信時間オーバーライド]	メッセージが生成されてから送り先で表示できるようになるまでのデフォルトの遅延時間（ミリ秒単位）を、プロデューサまたは接続ファクトリによって指定された送信時間に関わらず定義する。 デフォルト値 (-1) は送り先が送信時間の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1、または配信スケジュールの文字列構文	-1
[配信モードのオーバライド]	メッセージ プロデューサによって指定された配信モードに関わりなく、送り先に届くすべてのメッセージに割り当てられる配信モード。 デフォルトの [配信しない] は配信モードがオーバーライドされないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	[永続]、[非永続]、または [配信しない]	[配信しない]

[再配信]

属性	説明	値の範囲	デフォルト
[再配信遅延のオーバライド]	コンシューマや接続ファクトリによって指定された再配信遅延に関わりなく、ロール バックまたは回復されたメッセージが再配信されるまでの遅延をミリ秒単位で定義する。 デフォルト値 (-1) は送り先が再配信遅延の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1	-1

属性	説明	値の範囲	デフォルト
[再配信の制限]	メッセージがエラー送り先に置かれるまでに試行できる再配信の回数。この回数の上限に達した場合、エラー送り先がコンフィグレーションされているかどうかによって、以下の動作が発生する。 ■ エラー送り先がコンフィグレーションされていない場合、またはエラー送り先の割り当てを超過した場合は、永続メッセージも非永続メッセージも、そのまま削除される。 ■ エラー送り先がコンフィグレーションされていて、割り当てを超過していない場合は、メッセージのログが記録され、メッセージは削除される。ただし、永続メッセージは、永続ストアに残る。このため、WebLogic Server が再起動した時、永続メッセージは確実に再配信される。 デフォルト値 (-1) は送り先が再配信制限の設定をオーバーライドしないことを示す。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	-1、0 ～ 2^{63} ñ 1	-1
[エラー送り先]	再配信の上限に達したメッセージの送り先。エラー送り先が NULL の場合は、このメッセージはそのまま削除される。 この属性は動的にコンフィグレーションできるが、受信するメッセージにのみ適用され、保存されているメッセージには影響しない。	既存の送り先またはなし	なし

[マルチキャスト]

属性	説明	値の範囲	デフォルト
[マルチキャスト アドレス]	マルチキャスト用の IP アドレス。このアドレスはマルチキャスト コンシューマにメッセージを送信するために使用される。 この属性はトピックに関してのみ有効で、動的にコンフィグレーションすることはできない。	正しい形式の IP アドレス（クラス D）	N/A
[マルチキャスト生存時間]	マルチキャストの生存時間の値。メッセージがコンシューマに到達するまでに通過するルータの数を指定する。値 0 はメッセージがルータを越えず、1 つのサブネット内に制限されることを示す。 この値は `JMSExpirationTime` の値とは無関係である。 この属性はトピックに関してのみ有効で、動的にコンフィグレーションすることはできない。	0 ～ 255	0
[マルチキャスト ポート]	マルチキャスト用の IP ポート。このポートはマルチキャスト コンシューマにメッセージを送信するために使用される。 この属性はトピックに関してのみ有効で、動的にコンフィグレーションすることはできない。	1024 ～ 65535	6001

[モニタ]

[すべてのアクティブな JMS の送り先のモニタ]

属性	説明	値の範囲	デフォルト値
[送り先]	送り先名。		なし
[サーバ]	関連するサーバ名。		なし
[コンシューマ数]	現在の登録済みメッセージ コンシューマの数。		なし
[最大コンシューマ数]	任意の時点における登録済みメッセージ コンシューマの最大数。		なし
[コンシューマ総数]	登録済みメッセージ コンシューマの総数。		なし
[現在のバイト数]	現在保存されているバイト数。		なし
[保留バイト数]	保存されていて、確認応答およびコミットが行われていないトランザクション バイト数。		なし
[受信バイト数]	受信済みバイト数。		なし
[バイトしきい値の時間]	サーバの起動以降、送り先がバイトしきい値状態になっていた時間の合計。		なし

属性	説明	値の範囲	デフォルト値
[メッセージ数]	この送り先に現在格納されているメッセージの数。		なし
[最大メッセージ数]	ある時点において格納されるメッセージの最大数。		なし
[保留メッセージ数]	保存されていて、確認応答およびコミットが行われていないトランザクション メッセージの数。		なし
[受信メッセージ数]	受信済みメッセージ数。		なし
[メッセージしきい値の時間]	送り先がメッセージしきい値状態になっていた時間の合計。		なし

JMS キューのモニタの詳細については、「すべてのアクティブな JMS の送り先のモニタ」を参照してください。

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字の文字列	NULL

[メモ]

57 メッセージング ブリッジ

メッセージング ブリッジは、2 つの JMS プロバイダ間でメッセージの転送を行います。WebLogic メッセージング ブリッジの機能を使うと、WebLogic JMS の異なる実装を含む任意の 2 つの JMS プロバイダ間に、格納と転送のメカニズムをコンフィグレーションできます。

具体的には、JMS 送り先（キューまたはトピック）から別の JMS 送り先へのソース / ターゲット マッピング、およびその逆のマッピングをコンフィグレーションできます。ソース送り先に送られたメッセージは、次にターゲット送り先に自動転送されます。またメッセージング ブリッジを使うと、サービスの品質（QOS）、メッセージのフィルタ処理、トランザクション セマンティクス、および再接続ポリシーをコンフィグレーションできます。

マッピングされる各ソース / ターゲット JMS 送り先については、それが WebLogic JMS 送り先でもサード パーティの JMS プロバイダでも、[JMS ブリッジ送り先] インスタンスをコンフィグレーションする必要があります。ソース / ターゲット送り先が JMS に対応していないメッセージング製品である場合は、[一般ブリッジ送り先] インスタンスをコンフィグレーションする必要があります。

メッセージング ブリッジの作成

メッセージング ブリッジのコンフィグレーションは、以下の手順で行います。

1. [Messaging Bridge] ノードをクリックして展開します。
2. [ブリッジ] ノードをクリックして右ペインに [ブリッジ] タブを開きます。
3. 右ペインで [新しい Messaging Bridge のコンフィグレーション] リンクをクリックします。右ペインのコンフィグレーション ダイアログに、新しいメッセージング ブリッジのコンフィグレーションに関連するタブが表示されます。

4. [コンフィグレーション | 一般] タブで、[名前]、[ソース送り先]、[対象送り先]、[セレクタ]、[サービスの品質]、[QOS デグラデーション]、[最大待機時間]、[非同期モードを有効化]、[永続性を有効化]、および [起動する] という属性フィールドに値を入力します。メッセージング ブリッジの [一般] 属性の詳細については、「**[一般]**」を参照してください。
5. [作成] をクリックして、[名前] フィールドで指定した名前を持つ メッセージング ブリッジのインスタンスを作成します。新しいインスタンスが左ペインの [ブリッジ] ノード の下に追加されます。
6. [接続を再試行] タブをクリックし、属性フィールドを変更するか、割り当てられているデフォルト値を受け入れます。[接続を再試行] 属性の説明は、「[接続を再試行]」を参照してください。[適用] をクリックして変更を保存します。
7. [トランザクション] タブをクリックして、属性フィールドを変更するか、割り当てられているデフォルト値を受け入れます。[トランザクション] 属性の説明は、「[トランザクション]」を参照してください。[適用] をクリックして変更を保存します。
8. [対象] タブをクリックして、このメッセージング ブリッジ をサーバ、クラスタまたは移行できる対象のサーバに割り当てます。詳細については、「メッセージング ブリッジ のサーバ、クラスタまたは移行できる対象への割り当て」を参照してください。
9. [適用] をクリックして変更を保存します。

メッセージング ブリッジのクローンの作成

1. [ブリッジ] ノードをクリックして展開します。[Messaging Bridge] テーブルが右ペインに表示され、すべての メッセージング ブリッジのインスタンスを示します。

2. クローンを作成するメッセージング ブリッジの行で [クローン] アイコンをクリックします。右ペインのダイアログに、メッセージング ブリッジのクローン作成に関連するタブが表示されます。
3. 属性フィールドに値を入力します。
4. [クローン] をクリックして、[名前] フィールドで指定した名前を持つ メッセージング ブリッジを作成します。右ペインの [Messaging Bridge] テーブルに、メッセージング ブリッジ が追加されます。

メッセージング ブリッジの削除

1. [Messaging Bridge] ノードをクリックして展開します。[Messaging Bridge] テーブルが右ペインに表示され、すべてのメッセージング ブリッジのインスタンスを示します。
2. 削除するメッセージング ブリッジの行で [削除] アイコンをクリックします。右ペインに削除要求の確認を求めるダイアログが表示されます。
3. [はい] をクリックして、メッセージング ブリッジを削除した後、[続行] をクリックして右ペインに [Messaging Bridge] テーブルを再表示します。メッセージング ブリッジは、[Messaging Bridge] テーブルから削除されています。

メッセージング ブリッジ のサーバ、クラスタまたは移行できる対象への割り当て

1. 左ペインで、[ブリッジ] ノードをクリックして展開します。
2. [Messaging Bridge] ノードをクリックして展開し、ドメインで定義されているメッセージング ブリッジのリストを表示します。

3. 割り当てを行うメッセージング ブリッジをクリックします。右ペインのダイアログに、メッセージング ブリッジのインスタンスに関連するタブが表示されます。
4. [対象] タブをクリックすると、以下のような対象指定タブが表示されます。
   - [サーバ]ó メッセージング ブリッジ がデプロイされるサーバを 1 つまたは複数選択できます。メッセージング ブリッジ は、選択したすべてのサーバで使用可能になります。
   - [クラスタ]ó メッセージング ブリッジ がデプロイされるクラスタを 1 つ選択できます。メッセージング ブリッジ は、選択したクラスタ内のすべてのサーバで使用可能になります。
   - [移行できる対象]ó メッセージング ブリッジ がデプロイされるWebLogic Server の移行できる対象を 1 つ選択できます。WebLogic Server が初めて起動される際、このメッセージング ブリッジは、最初はユーザが指定したサーバ上でのみ 使用可能になります。その後、そのブリッジを移行できる対象としてリストされている別のサーバに移行できます。
5. [サーバ]、[クラスタ] または [移行できる対象] タブで、次のようにしてメッセージング ブリッジを割り当てます。
   a. [選択可] カラムで、メッセージング ブリッジに割り当てる対象を 1 つまたは複数選択します。
   b. 移動コントロールをクリックして、対象を [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

メッセージング ブリッジへのメモの追加

1. 左ペインで、[ブリッジ] ノードをクリックして展開します。
2. [Messaging Bridge] ノードをクリックして展開し、ドメインで定義されているメッセージング ブリッジのリストを表示します。

3. メモを追加するメッセージング ブリッジをクリックします。右ペインのダイアログに、メッセージング ブリッジのインスタンスに関連するタブが表示されます。
4. [メモ] タブをクリックします。[メモ] フィールドにメモを入力します。
5. [適用] をクリックして変更を保存します。

メッセージング ブリッジの停止と再起動

メッセージング ブリッジの一時的サスペンドは、以下の手順で行います。

1. [Messaging Bridge] ノードをクリックして展開します。
2. 停止するメッセージング ブリッジのインスタンスを選択します。
3. [コンフィグレーション | 一般] タブで、[起動する] チェック ボックスをオフにして、ブリッジを停止します。
4. ブリッジを再起動するには、[起動する] チェック ボックスをオンにします。

すべてのアクティブなメッセージング ブリッジのモニタ

1. 左ペインで [サーバ] ノードをクリックします。
2. 左ペインで特定のサーバを選択します。右ペインのダイアログに、そのサーバ インスタンスに関連するタブが表示されます。
3. [サービス] タブを選択します。
4. [ブリッジ] タブを選択します。

5. [すべての メッセージング ブリッジ ランタイム のモニタ] テキスト リンクをクリックします。テーブルが表示され、サーバのアクティブなメッセージング ブリッジがすべて示されます。

実行スレッド プール サイズのコンフィグレーション

メッセージング ブリッジの実行スレッド プールのデフォルト サイズをコンフィグレーションできます。たとえば、WebLogic Server のデフォルト スレッド プールから競合を減らすために、このデフォルト サイズを増減するとよいでしょう。入力する値を -1 にすると、このスレッド プールは無効となり、ブリッジは必ず WebLogic Server のデフォルト のスレッド プールを使用します。

1. 左ペインで [サーバ] ノードをクリックして展開します。
2. 所定のサーバ インスタンスを選択します。
3. 右ペインで [サービス] タブを選択します。
4. [ブリッジ] タブを選択します。
5. [メッセージング ブリッジ スレッドのプール サイズ] フィールドに新しい値を入力します。
6. [適用] をクリックして変更を保存します。

コンフィグレーション

[一般]

属性	説明	指定できる値	デフォルト値
[名前]	WebLogic Server ドメインでユニークなメッセージング ブリッジ 名。 この属性は動的にコンフィグレーションできない。	特定のドメイン内でユニークな Java 識別子	MyMessaging Bridge [-*n*]
[ソース送り先]	ソース送り先。メッセージング ブリッジはそこからメッセージを受信する。 この属性は動的にコンフィグレーションできない。	既存のソース ブリッジ 送り先名	Null
[対象送り先]	ターゲット送り先。メッセージング ブリッジはそこに向けてメッセージを送信する。 この属性は動的にコンフィグレーションできない。	既存のターゲット ブリッジ送り先名	Null

属性	説明	指定できる値	デフォルト値
[セレクタ]	メッセージング ブリッジで送信されるメッセージのフィルタ処理が可能になる。選択条件にかなうメッセージだけがメッセージング ブリッジで送信される。キューの場合、選択条件に合わないメッセージはそのまま残され、キューに累積される。トピックの場合、接続条件に合わないメッセージは捨てられる。 **注意：** セレクタを使用したメッセージのフィルタリングの詳細については、『WebLogic JMS プログラマーズ ガイド』の「WebLogic JMS アプリケーションの開発」を参照。 この属性は動的にコンフィグレーションできない。	英数字文字列	なし

属性	説明	指定できる値	デフォルト値
[サービスの品質]	メッセージング ブリッジでのメッセージ転送の保証を定義する。サービスの品質の有効値は、以下のとおり。 *[かならず 1 回]*ó 各メッセージは必ず 1 回送信される。サービスの品質は最高。 *[最大 1 回]*ó 各メッセージは最大 1 回送信される。メッセージによっては、対象送り先に配信されない場合がある。 *[重複可]*ó 各メッセージは、最低 1 回は送信される。対象送り先にメッセージを重複して送ることが可能。 **注意 :** [かならず 1 回] を使用するには、ソースとターゲットの接続ファクトリで `XAConnectionFactory` を使用するようコンフィグレーションされていることが必要。また、`jms-xa-adp.rar` アダプタがデプロイされ、ソース / ターゲット の送り先に対して [JNDI アダプタ名] 属性で識別されることが必要。 この属性は動的にコンフィグレーションできない。	[かならず 1 回]、[最大 1 回]、[重複可]	[かならず 1 回]

属性	説明	指定できる値	デフォルト値
[QOS デグラデーション]	選択すると、コンフィグレーションした QOS が有効ではない場合に、メッセージング ブリッジが自動的に要求した QOS のレベルを下げる。このような場合、WebLogic メッセージが 配信される。WebLogic スタートアップ ウィンドウ（または、ログ ファイル）にメッセージが配信される。このオプションが未選択 (false) の場合、メッセージング ブリッジは要求した QOS を満たすことができず、エラーとなって メッセージング ブリッジは起動しない。 この属性は動的にコンフィグレーションできない。	ブール値 有効 = 選択 無効 = 未選択	未選択
[最大待機時間]（ミリ秒）	非同期モードで実行されているブリッジの場合、これは接続の状態をチェックするまでのメッセージング ブリッジがアイドル状態でいる最長時間 (秒単位)。同期モードで実行されているブリッジの場合、これは、トランザクションが呼び出されていなければ、メッセージング ブリッジが受信呼び出しをブロックすることができる時間を示す。	0 ～ 2^{31}-1	60

属性	説明	指定できる値	デフォルト値
[非同期モードを有効化]	メッセージング ブリッジが非同期モードで動作するかどうかを定義する。非同期モードを有効 (true) にしたメッセージング ブリッジは、ソース送り先によって制御される。メッセージング ブリッジ はメッセージをリスンし、受信するとそのまま転送する。値が無効（false）の場合、たとえソース側が非同期受信をサポートしていても、ブリッジは強制的に同期モードで動作する。 **注意：** QOS を *[かならず1 回]* としたメッセージング ブリッジを非同期モードで動作させるには、`weblogic.jms.extensions` Javadoc で説明されているように、ソース送り先が `MDBTransaction` インタフェースをサポートすることが必要。そうでない場合、ソース送り先で `MDBTransactions` をサポートしていないことを検知すると、ブリッジは自動的に同期モードに切り替わる。`MDBTransactions` の詳細については、『WebLogic エンタープライズ JavaBeans プログラマーズ ガイド』の「メッセージ駆動型 Bean の使い方」を参照。 この属性は動的にコンフィグレーションできない。	ブール値 有効 = 選択 無効 = 未選択	選択

属性	説明	指定できる値	デフォルト値
[永続性を有効化]	この属性が有効（*true*）になっているメッセージング ブリッジは、ソース送り先タイプが「トピック」の場合には、ソース送り先に対する恒久サブスクリプションを作成する。こうすると、ブリッジが実行されていない場合に、ソース JMS 実装は、送られてきたメッセージを保存できる。ブリッジが再起動すると、保存していたメッセージをターゲット送り先に転送する。この値を無効（*false*）に設定すると、ブリッジがダウンしている間にソース JMS トピック宛てに送られてきたメッセージは、ターゲット送り先に転送できない。 **注意：** この属性は、JMS トピックに対して、または送り先が同様の特性を持つ場合に限り使用する。 **注意：** ブリッジを永続的にオフラインにする必要がある場合は、そのブリッジを使用する恒久サブスクリプションがあればすべて削除しなければならない。恒久サブスクリプション削除の詳細については、『*WebLogic JMS プログラマーズ ガイド*』の「恒久サブスクリプションの削除」を参照。 この属性は動的にコンフィグレーションできない。	ブール値 有効 = 選択 無効 = 未選択	未選択

属性	説明	指定できる値	デフォルト値
[起動する]	実行時のメッセージング ブリッジの初期状態（つまり、起動時の状態）を定義する。デフォルト値 は選択 (*true*)。このチェック ボックスを未選択 (*false*) にすることにより、メッセージング ブリッジの停止が可能。逆に、チェック ボックスを再度選択すると、ブリッジが再起動する。 **注意 :** これは、実行時のブリッジの状態を示すものではない。ブリッジの状態のモニタリングについては、57-5 ページの「すべてのアクティブなメッセージング ブリッジのモニタ」を参照。	ブール値 有効 = 選択 無効 = 未選択	選択

[接続を再試行]

属性	説明	指定できる値	デフォルト値
[最小遅延（秒）]	再接続試行の間隔の最少遅延（秒）。メッセージング ブリッジが起動し、送り先に接続できなかったり、接続が失われて、メッセージング ブリッジが最初の再接続を試みると、ここで指定した秒数のうちに再接続を試みる。 この属性は動的にコンフィグレーションできない。	0 ～ 2^{31}-1	15
[増加遅延（秒）]	再接続試行の間隔の遅延の増分（秒）。ブリッジが接続に失敗するたびに、ここで指定した秒数を遅延に加算してから、次の再接続を行う。 この属性は動的コンフィグレーションが可能。	0 ～ 2^{31}-1	5
[最大遅延（秒）]	再接続試行の間隔の最大遅延（秒）。再接続の試行は毎回 [増加遅延（秒）] で指定した秒数ずつ遅延していくが、ここで指定した値を超えて遅延することはない。 この属性は動的コンフィグレーションが可能。	0 ～ 2^{31}-1	60

[トランザクション]

属性	説明	指定できる値	デフォルト値
[トランザクション タイムアウト]	トランザクション マネージャが、各トランザクションをタイムアウトにするまでに、待つ秒数を定義する。トランザクション タイムアウトは、ブリッジのサービスの品質でトランザクションを要求する場合に使われる。 この属性は動的にコンフィグレーションできない。	0 ～ 2^{31}-1	30
[バッチ サイズ]	メッセージング ブリッジが 1 トランザクション内で転送するメッセージ数を定義する。同期モードで動作し、サービスの品質で 2 フェーズ トランザクションを要求するブリッジにのみこの属性が適用される。 この属性は動的コンフィグレーションが可能。	0 ～ 2^{31}-1	10
[バッチ間隔（ミリ秒）]	[バッチ サイズ] の量に到達したかどうかに関わらず、1 トランザクションでメッセージのバッチを送信するまでにブリッジが待つ最大時間をミリ秒で定義する。同期モードで動作し、サービスの品質で 2 フェーズ トランザクションを要求するブリッジにのみこの属性が適用される。 この属性は動的コンフィグレーションが可能。	0 ～ 2^{31}-1	-1

[対象]

属性	説明	指定できる値	デフォルト値
[移行できる対象]	メッセージング ブリッジ がデプロイされる WebLogic Server の移行できる対象を定義する。WebLogic Server が初めて起動される際、メッセージング ブリッジは、最初はユーザが指定したサーバ上でのみ使用可能となる。その後、Administration Console やコマンド ライン ツールを使って、移行できる対象に含まれている別のサーバへの移行が可能。	使用可能で選択済みの対象の一覧	選択された対象はない
[クラスタ]	メッセージング ブリッジがデプロイされる WebLogic Server クラスタを定義する。メッセージング ブリッジは、選択したクラスタ内のすべてのサーバで使用可能となる。	使用可能で選択済みの対象の一覧	選択された対象はない

属性	説明	指定できる値	デフォルト値
[サーバ]	メッセージング ブリッジがデプロイされる WebLogic Server を定義する。メッセージング ブリッジは、選択したすべての WebLogic Server で使用可能となる。	使用可能で選択済みの対象の一覧	選択された対象はない

[メモ]

属性	説明	指定できる値	デフォルト値
[メモ]	ユーザが情報を入力するための領域を提供する。	英数字文字列	Null

58 JMS ブリッジ送り先

WebLogic による JMS の実装かサード パーティの JMS プロバイダかにかかわらず、マッピングされる各ソース / ターゲット JMS 送り先に対して、[JMS ブリッジ送り先] インスタンスをコンフィグレーションする必要があります。

JMS ブリッジ送り先の作成

1. [Messaging Bridge] ノードをクリックして展開します。
2. [JMS ブリッジ送り先] ノードをクリックして、右ペインに [JMS ブリッジ送り先] タブを開きます。
3. 右ペインで [新しい JMS Bridge Destination のコンフィグレーション] リンクをクリックします。右ペインのコンフィグレーション ダイアログに、新しい JMS ブリッジ送り先のコンフィグレーションに関連するタブが表示されます。
4. [コンフィグレーション] タブで、[名前]、[JNDI アダプタ名]、[アダプタ クラスパス]、[接続 URL]、[初期コンテキスト ファクトリ]、[接続ファクトリ JNDI 名]、[送り先 JNDI 名]、および [送り先タイプ] という属性フィールドに値を入力します。[一般] 属性の詳細については、「[コンフィグレーション]」を参照してください。
5. オプションとして、[ユーザ名] 属性および [ユーザ パスワード] 属性を入力します。これらの属性の詳細については、「[コンフィグレーション]」を参照してください。

   注意 : 指定した送り先に対する操作はすべて、このユーザ名とパスワードを使って行われます。そのため、メッセージング ブリッジを機能させるには、ソース / ターゲットの送り先のユーザ名 / パスワードには、基になる送り先へのアクセス権が必要です。

6. [作成] をクリックして、[名前] フィールドで指定した名前を持つブリッジ送り先のインスタンスを作成します。新しいインスタンスが左ペインの [JMS ブリッジ送り先] ノードの下に追加されます。
7. ソース JMS ブリッジ送り先の属性の定義が完了したら、ターゲット JMS ブリッジ送り先も、この手順を繰り返してコンフィグレーションします。コンフィグレーションの順序は逆でもかまいません。

メッセージング ブリッジのコンフィグレーションで次に行うべき作業については、57-1 ページの「メッセージング ブリッジ」を参照してください。

JMS ブリッジ送り先のクローンの作成

1. [Messaging Bridge] ノードをクリックして展開します。
2. [JMS ブリッジ送り先] ノードをクリックして展開します。[JMS ブリッジ送り先] テーブルが右ペインに表示され、すべての JMS ブリッジ送り先のインスタンスを示します。
3. クローンを作成する JMS ブリッジ送り先のインスタンスの行で [クローン] アイコンをクリックします。右ペインのダイアログに、JMS ブリッジ送り先のクローン作成に関連するタブが表示されます。
4. 属性フィールドに値を入力します。
5. [クローン] をクリックして、[名前] フィールドで指定した名前を持つ JMS ブリッジ送り先を作成します。右ペインの [Messaging Bridge] テーブルに、JMS ブリッジ送り先が追加されます。

JMS ブリッジ送り先の削除

1. [Messaging Bridge] ノードをクリックして展開します。
2. [JMS ブリッジ送り先] ノードをクリックして展開します。[JMS ブリッジ送り先] テーブルが右ペインに表示され、すべての JMS ブリッジ送り先のインスタンスを示します。

3. 削除する JMS ブリッジ送り先の行で [削除] アイコンをクリックします。右ペインに削除要求の確認を求めるダイアログが表示されます。
4. [はい] をクリックして、JMS ブリッジ送り先を削除した後、[続行] をクリックして右ペインに [JMS ブリッジ送り先] テーブルを再表示します。ブリッジ送り先は、[JMS ブリッジ送り先] テーブルから削除されています。

JMS ブリッジ送り先へのメモの追加

1. 左ペインで、[Messaging Bridge] ノードをクリックして展開します。
2. [JMS ブリッジ送り先] ノードをクリックして展開し、ドメインで定義されている JMS ブリッジ送り先のリストを表示します。
3. メモを追加する JMS ブリッジ送り先をクリックします。右ペインのダイアログに、JMS ブリッジ送り先のインスタンスに関連するタブが表示されます。
4. [メモ] タブをクリックします。[メモ] フィールドにメモを入力します。
5. [適用] をクリックして変更を保存します。

コンフィグレーション

[コンフィグレーション]

属性	説明	指定できる値	デフォルト値
[名前]	WebLogic Server ドメイン内でユニークなブリッジ送り先名。 この属性は動的にコンフィグレーションできない。	特定のドメイン内でユニークな Java 識別子	MyJMS Bridge Destination [-*n*]
[JNDI アダプタ名]	アダプタの JNDI 名。これは、アダプタの記述子ファイルの中で定義されている文字列で、これを使って WebLogic Server JNDI にアダプタをバインドする。 ブリッジ送り先は、指定した送り先との通信に使用するアダプタの JNDI 名を指定する必要がある。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペース内でユニークな Java 識別子	eis.jms.WLS ConnectionFactoryJNDIXA

属性	説明	指定できる値	デフォルト値
[アダプタ クラスパス]	ブリッジ送り先の CLASSPATH。主に、WebLogic JMS の各種リリースと接続するために使われる。 WebLogic Server 6.0 以前の環境で実行されている送り先に接続する際、ブリッジ送り先は、旧版の WebLogic Server 実装用のクラスの置かれている位置を示す CLASSPATH を指定する必要がある。 この属性は動的にコンフィグレーションできない。	値は英数字文字列でなければならない。	Null
[接続 URL]	接続ファクトリと送り先のルックアップに使われる JNDI プロバイダの URL。 この属性は動的にコンフィグレーションできない。	値は英数字文字列でなければならない。	Null
[初期コンテキスト ファクトリ]	JNDI コンテキストの取得に使われるファクトリ。 この属性は動的にコンフィグレーションできない。	値は英数字文字列でなければならない。	weblogic.jndi.WLInitialContextFactory
[接続ファクトリ JNDI 名]	接続の作成に使われる JMS 接続ファクトリ。 **注意：** *[かならず 1 回]* という QOS を使用するには、この接続ファクトリは XAConnectionFactory でなければならない。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペース内でユニークな Java 識別子	Null

属性	説明	指定できる値	デフォルト値
[送り先 JNDI 名]	JMS 送り先の JNDI 名。 この属性は動的にコンフィグレーションできない。	JNDI ネームスペース内でユニークな Java 識別子	Null
[送り先タイプ]	[キュー] または [トピック] のどちらか。 この属性は動的にコンフィグレーションできない。	[キュー] または [トピック]	[キュー]
[ユーザ名]	メッセージング ブリッジがブリッジ アダプタに与えるオプションのユーザ名。 **注意：** 指定した送り先に対する操作はすべて、この [ユーザ名] と [ユーザ パスワード] を使って行われます。そのため、メッセージング ブリッジを機能させるには、ソース / ターゲットの送り先のユーザ名 / パスワードには、基になるソース / ターゲットの送り先へのアクセス権が必要です。 この属性は動的にコンフィグレーションできない。	値は英数字文字列でなければならない。	Null
[ユーザ パスワード]	メッセージング ブリッジがブリッジ アダプタに与えるパスワード。 この属性は動的にコンフィグレーションできない。	値は英数字文字列でなければならない。	Null

[メモ]

属性	説明	指定できる値	デフォルト値
[メモ]	この属性は、ユーザが情報を入力するための領域を提供する。	値は英数字文字列でなければならない。	Null

59 一般ブリッジ送り先

非 JMS メッセージング製品の場合は、メッセージング ブリッジによってマッピングされるソース送り先と対象送り先ごとに、一般的な BridgeDestination インスタンスをコンフィグレーションする必要があります。

注意： WebLogic JMS には非 JMS メッセージング製品へのアクセス用に仮の General Bridge Destination フレームワークが含まれていますが、WebLogic Server ではこのような製品に対するアダプタはサポートされていません。したがって、サードパーティの OEM ベンダからカスタム アダプタを入手し、そのマニュアルに従ってコンフィグレーションを行う必要があります。非 JMS カスタム アダプタの入手方法の詳細については、BEA プロフェッショナル サービスにお問い合わせいただいてもかまいません。

一般ブリッジ送り先の作成

1. [Messaging Bridge] ノードを展開します。
2. [一般ブリッジ送り先] ノードをクリックして右ペインに [一般ブリッジ送り先] タブを開きます。
3. 右ペインで [新しい General Bridge Destination のコンフィグレーション] リンクをクリックします。[コンフィグレーション] ダイアログに、新しい一般ブリッジ送り先のコンフィグレーションに関連するタブが表示されます。
4. [コンフィグレーション] タブで、[名前]、[JNDI アダプタ名]、[アダプタ クラスパス]、[プロパティ] という属性フィールドに値を入力します。[一般] 属性の詳細については、「[コンフィグレーション]」を参照してください。
5. オプションとして、[ユーザ名] 属性および [ユーザ パスワード] 属性を入力します。これらの属性の詳細については、「[コンフィグレーション]」を参照してください。

注意： 指定した送り先に対する操作はすべて、このユーザ名とパスワードを使って行われます。そのため、メッセージング ブリッジを機能させるには、ソース / ターゲットの送り先のユーザ名 / パスワードには、基になる送り先へのアクセス権が必要です。

6. [作成] をクリックして、[名前] フィールドで指定した名前を持つ 一般ブリッジ送り先のインスタンスを作成します。新しいインスタンスが左ペインの [一般ブリッジ送り先] ノードの下に追加されます。
7. ソース一般ブリッジ送り先の属性の定義が完了したら、ターゲット一般ブリッジ送り先も、この手順を繰り返してコンフィグレーションします。コンフィグレーションの順序は逆でもかまいません。

メッセージング ブリッジのコンフィグレーションで次に行うべき作業について 70-1 ページの「メッセージング ブリッジ」を参照してください。

一般ブリッジ送り先のクローンの作成

1. [Messaging Bridge] ノードおよび [一般ブリッジ送り先] ノードを展開します。
2. [一般ブリッジ送り先] テーブルが右ペインに表示され、すべての 一般ブリッジ送り先 のインスタンスを示します。
3. クローンを作成する一般ブリッジ送り先のインスタンスの行で [クローン] アイコンをクリックします。一般ブリッジ送り先のクローン作成に関連するタブがダイアログに表示されます。
4. 属性フィールドに値を入力します。
5. [クローン] をクリックして、[名前] フィールドで指定した名前を持つ 一般ブリッジ送り先を作成します。右ペインの [一般ブリッジ送り先] テーブルに、一般ブリッジ送り先が追加されます。

一般ブリッジ送り先の削除

1. [Messaging Bridge] ノードおよび [一般ブリッジ送り先] ノードを展開します。
2. [一般ブリッジ送り先] テーブルが右ペインに表示され、すべての一般ブリッジ送り先のインスタンスを示します。
3. 削除する一般ブリッジ送り先ののインスタンスの行で [削除] アイコンをクリックします。削除要求の確認を求めるプロンプトがダイアログに表示されます。
4. [はい] をクリックして、一般ブリッジ送り先を削除した後、[続行] をクリックして右ペインに [一般ブリッジ送り先] テーブルを再表示します。一般ブリッジ送り先 は、[一般ブリッジ送り先] テーブルから削除されています。

一般ブリッジ送り先へのメモの追加

1. [Messaging Bridge] ノードを展開します。
2. [一般ブリッジ送り先] ノードをクリックして展開し、ドメインで定義されている一般ブリッジ送り先のリストを表示します。
3. メモを追加する一般ブリッジ送り先をクリックします。一般ブリッジ送り先のインスタンスに関連するタブがダイアログに表示されます。
4. [メモ] タブをクリックします。[メモ] フィールドにメモを入力します。
5. [適用] をクリックして変更を保存します。

コンフィグレーション

[コンフィグレーション]

属性	説明	指定できる値	デフォルト値
[名前]	WebLogic ドメイン内でユニークなブリッジ 送り先名。ソース送り先の場合は、デフォルトの名前を「Source Bridge Destination」のような名前に変更する。ターゲットの場合は、「Target Bridge Destination」のような名前を使用する。コンフィグレーションした名前は、[ブリッジ] の [一般] タブの [ソース送り先] 属性と [対象送り先] 属性のオプションとして表示される。 この属性は動的にコンフィグレーションできない。	特定のドメイン内でユニークな Java 識別子	MyGeneral Bridge Destination [-*n*]
[JNDI アダプタ名]	ブリッジ 送り先は、指定した送り先との通信に使用するアダプタの JNDI 名を指定する必要がある。 WebLogic Server では、非 JMS メッセージング製品に対するアダプタは提供されていない。したがって、サードパーティ OEM ベンダの専用アダプタを使用するか、BEA プロフェッショナル サービスに問い合わせる必要がある。 この属性は、動的コンフィグレーションが可能。	JNDI ネームスペース内でユニークな Java 識別子	eis.jms.WLS ConnectionFactoryJNDIXA

属性	説明	指定できる値	デフォルト値
[アダプタ クラスパス]	ブリッジ 送り先の CLASSPATH 。主に、WebLogic JMS の各種リリースと接続するために使われる。 WebLogic Server 6.0 以前の環境で実行されている送り先に接続する際、ブリッジ 送り先は、旧版の WebLogic Server 実装用のクラスの置かれている位置を示す CLASSPATH を指定する必要がある。 サードパーティのメッセージング製品に接続するときは、WebLogic Server の CLASSPATH で製品の CLASSPATH を指定する必要がある。 この属性は、動的コンフィグレーションが可能。	値は英数字文字列でなければならない。	Null

属性	説明	指定できる値	デフォルト値

[プロパティ (key=value)]	ブリッジ送り先用に定義されたプロパティをすべて指定する。各プロパティは、セミコロンで区切ること（たとえば、`DestinationJNDIName=myTopic;DestinationType=topic;`）。
	サードパーティの OEM ベンダから提供されるアダプタを使用する非 JMS メッセージング製品の場合は、適切なコンフィグレーション方法についてはベンダのマニュアルを参照する必要がある。
	JMS のすべての実装では、以下のプロパティが必要である。
	`ConnectionURL` ñ 送り先との接続を確立するために使われる URL。
	`InitialContextFactory` ñ JNDI コンテキストの取得に使われるファクトリ。
	`ConnectionFactoryJNDIName` ñ 接続の作成に使われる JMS 接続ファクトリ。
	`DestinationJNDIName` ñ JMS 送り先の JNDI 名。
	`DestinationType` ñ Queue または Topic。
	この属性は、動的コンフィグレーションが可能。
		値は、以下のプロパティ ID の後ろに続く、英数字文字列でなければならない。
		`ConnectionURL=`
		`InitialContextFactory=`
		`ConnectionFactoryJNDIName=`
		`DestinationJNDIName=`
		`DestinationType=`
			Null

属性	説明	指定できる値	デフォルト値
[ユーザ名]	メッセージング ブリッジ がブリッジ アダプタに与えるオプションのユーザ名。 **注意 :** 指定した送り先に対する操作はすべて、この [ユーザ名] と [ユーザ パスワード] を使って行われます。そのため、メッセージング ブリッジを機能させるには、ソース / ターゲットのブリッジ送り先のユーザ名 / パスワードには、基になるソース / ターゲットの送り先へのアクセス権が必要です。 この属性は、動的コンフィグレーションが可能。	値は英数字文字列でなければならない。	Null
[ユーザ パスワード]	メッセージング ブリッジ がブリッジ アダプタに与えるパスワード。 この属性は、動的コンフィグレーションが可能。	値は英数字文字列でなければならない。	Null

[メモ]

属性	説明	指定できる値	デフォルト値
[メモ]	この属性は、ユーザが情報を入力するための領域を提供する。	値は英数字文字列でなければならない。	Null

60 JNDIBinding

WebLogic Server の EJB、RMI、JMS、JDBC、およびメール オブジェクトは、JNDI ツリーにバインドすることができます。

JMS 接続ファクトリまたは JDBC データ ソース オブジェクトを JNDI ツリーにバインドするには、それらオブジェクト作成時に、Administration Console を使用します。

JNDI の詳細については、『WebLogic Server 管理者ガイド』の「JNDI の管理」および『WebLogic JNDI プログラマーズ ガイド』を参照してください。

61 JNDI コンテキスト

JNDI ツリーは Administration Console から見ることができます。

1. 左ペインで、ユーザの WebLogic Server のノードを右クリックします。ポップアップ メニューが表示されます。
2. ［JNDI ツリーを見る］をクリックします。新しく開かれたウィンドウの右ペインに情報が表示されます。

JNDI の詳細については、『WebLogic Server 管理者ガイド』の「 JNDI の管理」および『WebLogic JNDI プログラマーズ ガイド』を参照してください。

62 Jolt

以下に、Administration Console を使用して、Jolt 接続プールのコンフィグレーションおよび管理に必要な属性を設定する手順を説明します。

Jolt 接続プールのコンフィグレーション

1. 左ペインの [Jolt] ノードをクリックします。ドメインで定義されているすべての Jolt 接続プールを示す [Jolt 接続プール] テーブルが右ペインに表示されます。
2. [新しい Jolt Connection Pool プールのコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する接続プールの設定に関連するタブが示されます。
3. [名前]、[最小プール サイズ]、[最大プール サイズ]、および [タイムアウト] 属性フィールドに値を入力します。[セキュリティ コンテキストを有効化] チェックボックスをクリックして、セキュリティ コンテキストの有効、無効を設定します。
4. [作成] をクリックして、[名前] フィールドで指定した名前の接続プール インスタンスを作成します。新しいインスタンスが左ペインの [Jolt] ノードの下に追加されます。
5. [アドレス] および [ユーザ] タブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

Jolt 接続プールのクローンの作成

1. 左ペインの [Jolt] ノードをクリックします。ドメインで定義されているすべての Jolt 接続プールを示す [Jolt 接続プール] テーブルが右ペインに表示されます。
2. クローンを作成する接続プールの行の [クローン] アイコンをクリックします。右ペインにダイアログが表示され、新しい接続プールのクローンの作成に関連するタブが示されます。
3. [名前]、[最小プール サイズ]、[最大プール サイズ]、および [タイムアウト] 属性フィールドに値を入力します。[セキュリティ コンテキストを有効化] チェックボックスをクリックして、セキュリティ コンテキストの有効、無効を設定します。
4. [作成] をクリックして、[名前] フィールドで指定した名前の接続プール インスタンスを作成します。新しいインスタンスが左ペインの [Jolt] ノードの下に追加されます。
5. [アドレス] および [ユーザ] タブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

Jolt 接続プールの削除

1. 左ペインの [Jolt] ノードをクリックします。ドメインで定義されているすべての Jolt 接続プールを示す [Jolt 接続プール] テーブルが右ペインに表示されます。
2. 削除する接続プールの行の [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして接続プールを削除します。[Jolt] ノードの下の接続プール アイコンが削除されます。

Jolt 接続プールの割り当て

1. 左ペインの [Jolt] の下のインスタンス ノードをクリックして、割り当て対象の接続プールを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. Jolt 接続ファクトリに割り当てる 1 つまたは複数のターゲットを [選択可] カラムで選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

Jolt 接続プールのすべてのインスタンスのモニタ

1. 左ペインの [Jolt] の下のインスタンス ノードをクリックして、モニタする接続プールを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [すべてのアクティブなプールのモニタ] テキスト リンクをクリックします。右ペインに [Jolt 接続プール] テーブルが表示され、サーバにデプロイされたすべての接続プール インスタンスが示されます。

63 Jolt 接続プール

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト
[名前]	この接続プールの名前を返す。	文字列	MyJolt Connection Pool
[最小プール サイズ]	ユーザがこの接続プールの最小プール サイズを設定できる。	整数	0
[最大プール サイズ]	ユーザがこの接続プールの最大プール サイズを設定できる。	整数	1
[タイムアウト]	クライアントが応答受信タイムアウトするまでの時間をユーザが設定できる。	整数	0

属性	説明	値の範囲	デフォルト
[セキュリティ コンテキストを有効化]	この接続プールに対してセキュリティ コンテキストを有効または無効にする。	ブール 有効 = 選択 無効 = 未選択	選択されていない

[アドレス]

属性	説明	値の範囲	デフォルト
[プライマリ アドレス]	ユーザがプライマリ アドレスを設定できる。	有効なアドレス	NULL
[フェイルオーバ アドレス]	ユーザが障害発生時に使用するアドレスを設定できる。	有効なアドレス	NULL

[ユーザ]

属性	説明	値の範囲	デフォルト
[ユーザ名]	この接続プールのユーザ名を設定する。	管理者が設定した有効なユーザ名	NULL
[ユーザ ロール]	この接続プールのユーザ ロールを設定する。	管理者が設定した有効なユーザ ロール	NULL
[ユーザ パスワード]	この接続プールのユーザ パスワードを設定する。	管理者が設定した有効なユーザ パスワード	NULL
[アプリケーション パスワード]	ユーザがこの接続プールのアプリケーション パスワードを設定できる。	有効なパスワード	NULL

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト
[対象サーバ]	ユーザがこの接続プールの対象サーバを設定できる。	有効なサーバ名	NULL

[クラスタ]

属性	説明	値の範囲	デフォルト
[対象クラスタ]	ユーザがこの接続プールの対象クラスタを設定できる。	有効なサーバ名	NULL

[メモ]

属性	説明	値の範囲	デフォルト
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

64 実行時 Jolt 接続プール

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- マシン
- 名前
- プール状態
- 接続数
- 最大容量

65 JTA

以下に、Administration Console を使用して JTA のコンフィグレーションに必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「トランザクションの管理」と「JDBC 接続の管理」、および『WebLogic JTA プログラマーズ ガイド』(開発者ガイド) を参照してください。

JTA のコンフィグレーション

1. [ドメイン] ノードをクリックします。右ペインにダイアログが表示され、ドメインの コンフィグレーションに関連するタブが示されます。
2. [JTA] タブをクリックします。
3. [タイムアウト秒数]、[トランザクションを保持する最長時間]、[beforeCompletion の反復上限]、[最大トランザクション数]、および [ユニーク名の最大数] 属性フィールドに値を入力するか、デフォルト値をそのまま使用します。
4. 必要に応じて、[ヒューリスティックを無視] 属性を有効または無効にします。
5. [適用] をクリックして、変更を保存します。

66 JTA トランザクション

デフォルトでは、このペインを使用すると、ユーザが少なくとも 5 秒間アクティブになっていたトランザクションを次の基準で並べ替えて表示させることができます。

- 名前
- ステータス
- サーバ数
- リソース
- プロパティ

表示されているトランザクションの存続期間の上限を変更するには、秒数を編集して [表示] をクリックします。

表示するカラムや並べ替えの順序を変更するには、[このビューをカスタマイズ] テキスト リンクをクリックします。

67 LDAP レルム（廃止）

以下に、Administration Console を使用して、LDAP レルムの作成や管理に必要な属性を設定する手順を説明します。LDAP レルムの詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

LDAP レルムの作成

1. 左ペインの [レルム] ノードをクリックします。ドメインで定義されているすべての LDAP レルムを示す [レルム] テーブルが右ペインに表示されます。
2. [新しい LDAP Realm V1 (Deprecated) のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するレルムの設定に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. 右下隅の [作成] ボタンをクリックして、レルムのインスタンスを [名前] フィールドで指定した名前で作成します。左ペインの [レルム] ノードの下に、新しいインスタンスが追加されます。
5. [LDAP レルム V1 (廃止)]、[ユーザ]、および [グループ] タブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

LDAP レルムのクローンの作成

1. 左ペインの [レルム] ノードをクリックします。ドメインで定義されているすべての LDAP レルムを示す [レルム] テーブルが右ペインに表示されます。

2. クローンを作成するレルムの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、レルムのクローンの作成に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前でレルム インスタンスを作成します。左ペインの [レルム] ノードの下に、新しいインスタンスが追加されます。
5. [LDAP レルム V1 (廃止)]、[ユーザ]、および [グループ] タブを個々にクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

LDAP レルムの削除

1. 左ペインの [レルム] ノードをクリックします。ドメインで定義されているすべての LDAP レルムを示す [レルム] テーブルが右ペインに表示されます。
2. 削除するレルムの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、レルムを削除します。[レルム] ノードの下にあるレルム アイコンが削除されます。

Windows NT セキュリティ レルムを使用するには、キャッシング レルムを有効にし、[基本レルム] フィールドに Windows NT セキュリティ レルムのクラス名を入力する必要があります。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	LDAP セキュリティ レルムの名前を指定する。たとえば、AccountingRealm など。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[レルム クラス名]	LDAP セキュリティ レルムを含む Java クラスの名前を指定する。Java クラスは、WebLogic Server の CLASSPATH に含まれている必要がある。	この属性は変更不可	

[LDAP レルム V1 (廃止)]

属性	説明	値の範囲	デフォルト値
[LDAP URL]	LDAP サーバの場所。URL を、LDAP サーバが稼動しているコンピュータ名とリスンしているポート番号に変更する。WebLogic Server で、SSL プロトコルを使用する LDAP サーバに接続するには、URL にある LDAP サーバの SSL ポートを使う。		ldap://ldapserver:389
[プリンシパル]	WebLogic Server で LDAP サーバへの接続に使用される LDAP ユーザの識別名 (DN)。このユーザは、LDAP ユーザやグループをリストできなくてはならない。		
[証明]	[プリンシパル] フィールドで定義された LDAP ユーザを認証するパスワード。	有効なパスワード	NULL

属性	説明	値の範囲	デフォルト値
[SSL の有効化]	LDAP サーバと WebLogic Server の間の通信を保護する SSL プロトコルを有効にするオプション。次のガイドラインに留意すること。 ♦ LDAP サーバが SSL プロトコルを使用するようコンフィグレーションされていない場合には、このフィールドを無効にする。 ♦ [ユーザ認証] フィールドを [external] に設定した場合は、このフィールドを有効化しなければならない。	ブール True = 選択されている False = 選択されていない	NULL
[認証プロトコル]	LDAP サーバを認証する、認証タイプ。Netscape Directory Server は CRAM-MD5 をサポートしている。Microsoft Site Server および Novell NDS は、Simple をサポートしている。	♦ [none] = 認証しない ♦ [simple] = パスワード認証 ♦ [CRAM-MD5] = 証明書認証	なし

[ユーザ]

属性	説明	値の範囲	デフォルト値
[ユーザ認証]	ユーザの認証方法を決定する。	次のいずれかの属性を設定する ♦ [local] に設定すると、LDAP セキュリティ レルムは、LDAP ディレクトリ サーバのパスワードを含むユーザ データを検索し、WebLogic Server のパスワードをチェックする。[local] 設定は、Netscape Directory Server と Microsoft Site Server に適している ♦ [external] に設定すると、LDAP セキュリティ レルムは、LDAP ディレクトリ サーバを WebLogic Server クライアントから提供されるユーザ名およびパスワードと結びつけることで、ユーザを認証する。[external] 設定では、SSL プロトコルを使用する必要がある。[external] 設定は Novell NDS に適している ♦ [bind] に設定すると、LDAP セキュリティ レルムはユーザを認証する。	bind
[ユーザ パスワード属性]	LDAP ユーザのパスワードを設定する。	LDAP ユーザのパスワード	NULL

属性	説明	値の範囲	デフォルト値
[ユーザ DN]	[ユーザ名 アトリビュート] の属性と組み合わせたときにユーザをユニークに識別できるように、この属性を属性リストに設定する。	文字列	NULL
[ユーザ名属性]	LDAP ユーザのログイン名を設定する。	このフィールドの値には LDAP ユーザの普通の名前を使用できるが、一般にはユーザ ID などの短縮した文字列を使用する。	NULL

[グループ]

属性	説明	値の範囲	デフォルト値
[グループ DN]	[グループ名属性] 属性と組み合わせて LDAP サーバ内のグループをユニークに識別する属性リストを入力する。	文字列	NULL
[グループ名属性]	LDAP サーバ内のグループの名前を入力する。通常は普通の名前。	文字列	NULL
[グループはコンテキスト]	グループ メンバーシップを LDAP サーバに記録する方法を指定する。	ブール 有効 = 選択 無効 = 未選択 各グループ エントリが 1 ユーザを含む場合には、この属性を有効にする。デフォルトでは、この属性は有効。 各グループ メンバーの属性を含む 1 つのグループ エントリがある場合には、この属性を無効にする	選択
[グループ ユーザ名属性]	グループ エントリ内のグループ メンバーを含む LDAP 属性の名前に設定する。	有効なグループ メンバー名	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

68 マシン

以下に、Administration Console を使用して、マシンのコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」および「WebLogic Server とクラスタのコンフィグレーション」を参照してください。

マシンのコンフィグレーション

1. [マシン] ノードをクリックします。右ペインに [マシン] テーブルが表示され、ドメインで定義されているすべてのマシンが示されます。
2. [新しい Machine のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するマシンの設定に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前でマシン インスタンスを作成します。左ペインの [マシン] ノードの下に、新しいインスタンスが追加されます。
5. 必要に応じて、[ノード マネージャ] タブをクリックして属性値を変更します。
6. [適用] をクリックして変更を保存します。

マシンのクローンの作成

1. [マシン] ノードをクリックします。右ペインに [マシン] テーブルが表示され、ドメインで定義されているすべてのマシンが示されます。

2. クローンを作成するマシンの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、マシンのクローンの作成に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. [クローン] をクリックして、[名前] フィールドに指定した名前でマシン インスタンスを作成します。左ペインの [マシン] ノードの下に、新しいインスタンスが追加されます。
5. 必要に応じて、[ノード マネージャ] タブをクリックして属性値を変更します。
6. [適用] をクリックして変更を保存します。

マシンの削除

1. [マシン] ノードをクリックします。右ペインに [マシン] テーブルが表示され、ドメインで定義されているすべてのマシンが示されます。
2. 削除するマシンの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、マシンを削除します。[マシン] ノードの下にあるマシン アイコンが削除されます。

マシンの割り当て

1. 左ペインの [マシン] の下で、割り当てるマシンのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サーバ] タブをクリックします。
3. [選択可] カラムで、マシンに割り当てる 1 つまたは複数のターゲットを選択します。

4. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
5. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この属性は、マシン名のリストを返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL

[サーバ]

属性	説明	値の範囲	デフォルト値
[サーバ]	このマシンで動作しているサーバのリストを返す。	リスト	NULL

[ノード マネージャ]

属性	説明	値の範囲	デフォルト値
[リスン アドレス]	マシンのリスン アドレスを割り当てる。		Localhost
[リスン ポート]	マシンのリスン ポートを割り当てる		7002
[認証]	サーバの稼動するセキュリティ認証を設定する。		NULL
[認証パスワード]	セキュリティ認証のパスワードを設定する。	認証に対する有効なパスワード	NULL
[確実に信頼されたファイル]	信頼された認証のあるファイルを示す		NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

69 メール セッション

メール セッションの作成

1. [メール] ノードをクリックします。右ペインに [メール セッション] テーブルが表示され、ドメインで定義されているすべてのメール セッションが示されます。
2. [新しい Mail Session のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するメール セッションの設定に関連するタブが示されます。
3. [名前]、[JNDI 名]、および [プロパティ] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前でメール セッションのインスタンスを作成します。左ペインの [メール] ノードの下に、新しいインスタンスが追加されます。

メール セッションのクローンの作成

1. [メール] ノードをクリックします。右ペインに [メール セッション] テーブルが表示され、ドメインで定義されているすべてのメール セッションが示されます。
2. クローンを作成するメール セッションの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、メール セッションのクローンの作成に関連するタブが示されます。
3. [名前]、[JNDI 名]、および [プロパティ] 属性フィールドに値を入力します。

4. [作成] をクリックして、[名前] フィールドに指定した名前でメール セッションのインスタンスを作成します。左ペインの [メール] ノードの下に、新しいインスタンスが追加されます。

メール セッションの削除

1. [メール] ノードをクリックします。右ペインに [メール セッション] テーブルが表示され、ドメインで定義されているすべてのメール セッションが示されます。
2. 削除するメール セッションの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、メール セッションを削除します。[メール] ノードの下にあるメール セッション アイコンが削除されます。

メール セッションのすべてのインスタンスのモニタ

1. [メール] ノードをクリックします。右ペインに [メール セッション] テーブルが表示され、ドメインで定義されているすべてのメール セッションが示されます。
2. モニタするメール セッションの行で [すべてのインスタンスのモニタ] アイコンをクリックします。右ペインにダイアログが表示され、メール セッションのすべてのインスタンスが示されます。

メール セッションの割り当て

1. 左ペインの [メール] の下で、割り当てるメール セッションのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] タブと [クラスタ] タブで次の手順を実行します。
   a. [選択可] カラムで、メール セッションに割り当てる 1 つまたは複数のターゲットを選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	このメール セッションの名前を返す。	文字列	NULL
[JNDI 名]	このメール セッションの JNDI 名を設定する。	文字列	NULL
[プロパティ]	このメール セッションに必要なプロパティを設定する。	mail.transport.protocol=SMTP mail.store.protocol=pop3	NULL

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	このメール セッションがデプロイされるサーバを選択できる。	リスト	未選択

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	このメール セッションがデプロイされるクラスタを選択できる。	リスト	未選択

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザがコメントを入力できる。	文字列	NULL

70 MBean

管理サーバでは、Management Bean（MBean）と呼ばれる JavaBean に似たオブジェクトが使用されます。MBean は、Sun の Java Management Extension（JMX）規格に基づいています。このオブジェクトを使用することで、ドメインのリソースに管理を目的としてアクセスできます。管理サーバには、コンフィグレーション MBean と実行時 MBean があります。コンフィグレーション MBean により、コンフィグレーション属性に対して SET（書き込み）アクセスと GET（読み込み）アクセスが両方とも可能になります。

詳細については、『管理者ガイド』を参照してください。

71 実行時メッセージ駆動型 EJB

以下の統計によって、メッセージ駆動型 EJB をモニタすることができます。

統計	説明
Idle Beans Count	現在アイドル中の Bean 数
Beans In Use Count	現在使用中の Bean 数
Waiter Total Count	待機中のトランザクション数
Timeout Total Count	タイムアウトになったトランザクション数
JMS Connection Alive	JMS 接続のステータス

72 NT レルム

以下に、Administration Console を使用して、レルムの作成や管理に必要な属性を設定する手順を説明します。レルムの詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

NT レルムのコンフィグレーション

1. 左ペインの [レルム] ノードをクリックします。右ペインに [レルム] テーブルが表示され、ドメインで定義されているすべての NT レルムが示されます。
2. [新しい NTRealm のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するレルムの設定に関連するタブが示されます。
3. [名前] および [プライマリ ドメイン] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前でレルム インスタンスを作成します。左ペインの [レルム] ノードの下に、新しいインスタンスが追加されます。

NT レルムのクローンの作成

1. 左ペインの [レルム] ノードをクリックします。右ペインに [レルム] テーブルが表示され、ドメインで定義されているすべての NT レルムが示されます。
2. クローンを作成するレルムの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、レルムのクローンの作成に関連するタブが示されます。
3. [名前] および [プライマリ ドメイン] 属性フィールドに値を入力します。

4. [作成] をクリックして、[名前] フィールドに指定した名前でレルム インスタンスを作成します。左ペインの [レルム] ノードの下に、新しいインスタンスが追加されます。

NT レルムの削除

1. 左ペインの [レルム] ノードをクリックします。右ペインに [レルム] テーブルが表示され、ドメインで定義されているすべての NT レルムが示されます。
2. 削除するレルムの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、レルムを削除します。[レルム] ノードの下にあるレルム アイコンが削除されます。

Windows NT セキュリティ レルムを使用するには、キャッシング レルムを有効にし、[基本レルム] フィールドに Windows NT セキュリティ レルムのクラス名を入力する必要があります。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	AccountingRealm など、Windows NT セキュリティ レルムの名前を指定する。	最大 256 文字の英数字	MyNTRealm[-n]
[レルム クラス名]	Windows NT セキュリティ レルムを実装する Java クラスの名前を指定する。Java クラスは、WebLogic Server の CLASSPATH に含まれている必要がある。	この属性は変更不可	weblogic.security.acl.ntrealm.NTRealm
[プライマリ ドメイン]	この属性を使用すると、Windows NT ドメインのユーザとグループが定義されているコンピュータのホストとポート番号を入力できる。		NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

73 プリファレンス

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[言語]	コンソールおよびヘルプ システムの言語を選択できる。	英語、日本語	英語
[自動更新間隔]	コンソールの自動更新時間を設定する。	整数（秒数）	10
[最後に選択したタブの位置を保存]	最後に選択したタブの位置に戻る機能を有効または無効にする。	ブール 有効 = 選択 無効 = 未選択	選択
[ナビゲーション ツリーを使用]	コンソールでナビゲーション ツリーを使用する機能を有効または無効にする。	ブール 有効 = 選択 無効 = 未選択	選択

[グラフ]

属性	説明	値の範囲	デフォルト値
[グラフデータの収集間隔]	グラフ データのポーリング間隔を設定する。	整数（ミリ秒単位）	500

74 RDBMS レルム

以下の表に、RDBMS レルムの作成や管理に必要な属性を示します。レルムの詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	AccountingRealm など、セキュリティ レルムの名前を指定する。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[レルム クラス名]	セキュリティ レルムが所属する Java クラスの名前を返す。Java クラスは、WebLogic Server の CLASSPATH に含まれている必要がある。	この属性は変更不可	

[データベース]

属性	説明	値の範囲	デフォルト値
[ドライバ]	JDBC ドライバの完全なクラス名を設定する。このクラス名は、WebLogic Server の CLASSPATH に含まれている必要がある。		NULL
[URL]	RDBMS レルムで使用しているデータベースの URL（JDBC ドライバのマニュアル参照）を設定する。		NULL
[ユーザ名]	データベースへのログインに使用するユーザ名を指定する。		NULL
[パスワード]	管理者によって指定された、データベースのデフォルト ユーザのパスワードを設定する。		NULL

[スキーマ]

属性	説明	値の範囲	デフォルト値
[スキーマ プロパティ]	使用中のデータベースからユーザ、グループ、ACL、およびパーミッションを取得するためのプロパティを指定する。	key=value のリスト	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

75 RDBMS デプロイメント記述子

weblogic-cmp-rdbms-jar.xml デプロイメント記述子

この章では、バージョン 6.1 の WebLogic Server に固有の XML 要素について説明します。これらの要素は weblogic600-cmp-rdbms-jar.xml ファイルを構成し、WebLogic Server EJB コンテナ内のデプロイメント記述子を定義するために使われるものです。これらの要素は、WebLogic Server Administration Console のほぼ同じ名前のフィールドに対応します。これらのデプロイメント記述子はバージョン 2.0 の EJB に対して使用します。

automatic-key-generation

値の範囲	なし
デフォルト値	なし
要件	省略可能
親要素	weblogic-rdbms-bean
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

automatic-key-generation 要素では、シーケンス / キー生成機能を使用するかどうかを指定します。

cmp-field

値の範囲	有効な名前
デフォルト値	なし
要件	フィールドの値は大文字 / 小文字が区別され、Bean 内のフィールド名と一致しなければならない。また、このフィールドに対応する `cmp-entry` エントリが `ejb-jar.xml` に必要である。

親要素	`weblogic-rdbms-bean`
	`field-map`
	`weblogic-rdbms-relation`
	`field-group`

デプロイメントファイル	`weblogic-cmp-rdbms-jar.xml`

機能

この名前は、データベースから取得された情報が入る、Bean インスタンス内のマッピング対象フィールドを指定します。

cmr-field

値の範囲	有効な名前
デフォルト値	なし
要件	cmr-field で参照されるフィールドに対して、値が一致する cmr-field エントリが ejb-jar.xml に必要である。
親要素	weblogic-rdbms-relation field-group
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

cmr-field 要素は、cmr-field の名前を指定します。

column-map

値の範囲	なし
デフォルト値	なし
要件	foreign-key-column がリモート Bean を参照する場合、key-column 要素は指定されない。
親要素	weblogic-rdbms-bean 　　weblogic-relationship-role
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

この要素は、データベースの 1 つのテーブル内の外部キー列から、対応する主キーへのマッピングを表現します。2 つの列は同じテーブルに存在していても、していなくてもかまいません。列が属するテーブルは、column-map 要素がデプロイメント記述子に出現するコンテキストから暗黙のうちに自明です。

create-default-dbms-table

値の範囲	True | False
デフォルト値	False
要件	この要素は、開発およびプロトタイピング フェーズの間にのみ便宜的に使用する。これは、使用される DBMS CREATE 文におけるテーブル スキーマが、コンテナの最適な定義概算となるからである。ほとんどの場合、プロダクション環境ではより正確なスキーマ定義が必要となる。
親要素	`weblogic-rdbms-jar`
デプロイメント ファイル	`weblogic-cmp-rdbms-jar.xml`

機能

`create-default-dbms-table` 要素では、デプロイメント ファイルの記述および Bean クラスに基づいてデフォルト テーブルを自動的に作成する機能を有効または無効にします。False に設定すると、この機能は無効になり、テーブルは自動的に生成されません。`True` に設定すると、この機能が有効になりテーブルが自動的に作成されます。`TABLE CREATION` が失敗すると、`Table Not Found` エラーが送出されます。その場合、テーブルを手動で作成しなければなりません。

data-source-name

値の範囲	この Bean のすべてのデータベース接続に使われるデータ ソースの有効な名前
デフォルト値	なし
要件	データベース接続用の、標準の WebLogic Server JDBC データ ソースとして定義しなければならない。
親要素	`weblogic-rdbms-bean`
デプロイメント ファイル	`weblogic-cmp-rdbms-jar.xml`

機能

`data-source-name` 要素では、この Bean のすべてのデータベース接続で使われる JDBC データ ソース名を指定します。

db-cascade-delete

値の範囲	
デフォルト値	なし
要件	Oracle データベースだけに対応する要素。1 対 1 または 1 対多の関係に対してのみ指定可能。
親要素	`weblogic-rdbms-bean` `weblogic-relationship-role`
デプロイメント ファイル	`weblogic-cmp-rdbms-jar.xml`

機能

`db-cascade-delete` 要素では、データベース カスケード機能を有効にするかどうかを指定します。

dbms-column

値の範囲	有効な名前
デフォルト値	なし
要件	dbms-column では大文字 / 小文字が保持されるが、すべてのデータベースで大文字 / 小文字が区別されるとは限らない。
親要素	weblogic-rdbms-bean field-map
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

フィールドがマッピングされるデータベース列の名前です。

dbms-column-type

値の範囲	有効な名前
デフォルト値	なし
要件	Oracle データベースとの組み合わせでのみ使用可能。
親要素	weblogic-rdbms-bean field-map
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

dbms-column-type 要素は、現在のフィールドを Oracle データベースの BLOB または CLOB にマッピングします。この要素には OracleBlob または OracleClob のどちらかを使用できます。

delay-database-insert-until

値の範囲	
デフォルト値	ejbPostCreate
要件	NULL 値を許容しない foreign-key column に cmr-field がマッピングされるとき、データベースへの挿入操作は ejbPostCreate が完了するまで遅延される。この場合、Bean がデータベースに挿入される前に、ejbPostCreate において cmr-field を NULL 以外の値に設定しなければならない。 Bean の主キーが判明するまでは、ejbCreate の実行中に cmr-field が設定されない場合がある。
親要素	weblogic-rdbms-bean
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

delay-database-insert-until 要素では、RDBMS CMP を使用する新しい Bean がデータベースに挿入されるときの正確な時間を指定します。

ejbPostCreate メソッドが Bean の永続フィールドを変更し終えるまでの間、データベースへの挿入操作を遅延することが推奨されます。これにより、不必要なストア操作が回避され、パフォーマンスが向上する可能性があります。

柔軟性を最大限に高めるためには、互いに関連する Bean を ejbPostCreate メソッドで作成しないようにすることをお勧めします。このような処理を行うと、まだ作成されていない Bean を関連する Bean が参照することを禁じるデータベース制約がある場合に、データベース挿入の遅延を実施できなくなる場合があります。

ejb-name

値の範囲	EJB の有効な名前
デフォルト値	なし
要件	ejb-jar.xml で定義される、CMP エンティティ Bean の ejb-name と一致しなければならない。
親要素	weblogic-rdbms-bean
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

ejb-cmp-rdbms.xml で定義される EJB を指定する名前。この名前は ejb-jar.xml における、CMP エンティティ Bean の ejb-name と一致しなければなりません。

field-group

値の範囲	有効な名前
デフォルト値	グループが指定されていないファインダおよび関係に対しては、`default` という名前の特殊なグループが使われる。
要件	デフォルト グループには Bean の cmp-field がすべて収められるが、cmr-field は収められない。
親要素	`weblogic-rdbms-relation`
デプロイメント ファイル	`weblogic-cmp-rdbms-jar.xml`

機能

`field-group` 要素は、Bean の cmp-field および cmr-field のサブセットを表現します。Bean 内の互いに関連するフィールドは、1 つの単位としてメモリに展開されるグループにまとめることができます。グループをファインダまたは関係と関連付けることによって、ファインダを実行した、または関係を追跡した結果として Bean がロードされるとき、グループに指定したフィールドだけがロードされるようにすることができます。

フィールドは複数のグループに属することができます。この場合、フィールドに対しての `getXXX` メソッドでは、そのフィールドを含む最初のグループが対象となります。

field-map

値の範囲	有効な名前
デフォルト値	なし
要件	データベースの列にマッピングされるフィールドは、Bean の CMP フィールドに対応しなければならない。
親要素	`weblogic-rdbms-bean`
デプロイメントファイル	`weblogic-cmp-rdbms-jar.xml`

機能

Bean インスタンスの CMP フィールドに対応し、データベースの特定の列にマッピングされるフィールドの名前です。

foreign-key-column

値の範囲	有効な名前
デフォルト値	なし
要件	外部キーの列に対応しなければならない。
親要素	`weblogic-rdbms-bean` `column-map`
デプロイメントファイル	`weblogic-cmp-rdbms-jar.xml`

機能

`foreign-key-column` 要素は、データベースの外部キーの列を表します。

generator-name

値の範囲	なし
デフォルト値	なし
要件	省略可能
親要素	weblogic-rdbms-bean automatic-key-generation
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

generator-name 要素は、ジェネレータの名前を指定するために使われます。

次に例を示します。

- generator-type 要素の値が ORACLE の場合、generator-name 要素の値は使用される ORACLE_SEQUENCE の名前になります。
- generator-type 要素の値が NAMED_SEQUENCE_TABLE の場合、generator-name 要素の値は使用される SEQUENCE_TABLE の名前になります。

generator-type

値の範囲	なし
デフォルト値	なし
要件	省略可能
親要素	weblogic-rdbms-bean automatic-key-generation
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

generator-type 要素では、使用するキー生成メソッドを指定します。指定できる値には、次のものがあります。

- ORACLE
- SQL_SERVER
- NAMED_SEQUENCE_TABLE

group-name

値の範囲	有効な名前
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-relation field-group weblogic-rdbms-bean finder finder-query
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

group-name 要素では、フィールド グループの名前を指定します。

include-updates

値の範囲	True | False
デフォルト値	False
要件	パフォーマンスを優先する場合、デフォルト値の False を使用する。
親要素	weblogic-rdbms-bean weblogic-query
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

include-updates 要素では、現在のトランザクションの間に行われる更新を、クエリの結果に反映する必要があるかどうかを指定します。この要素を True に設定すると、コンテナはクエリを実行する前に、現在のトランザクションによって行われるすべての変更をディスクに書き出します。

key-cache-size

値の範囲	なし
デフォルト値	なし
要件	省略可能
親要素	weblogic-rdbms-bean automatic-key-generation
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

key-cache-size 要素は、主キー自動生成機能で利用できる主キー キャッシュのサイズを指定します。この要素は省略可能です。

key-column

値の範囲	有効な名前
デフォルト値	なし
要件	主キーの列に対応しなければならない。
親要素	weblogic-rdbms-bean column-map
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

key-column 要素は、データベースの主キーの列を表します。

max-elements

値の範囲	なし
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-bean weblogic-query
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

max-elements 要素では、多値クエリによって返される要素数の上限を指定します。この要素は JDBC の maxRows 機能に似た働きをします。

method-name

値の範囲	なし
デフォルト値	なし
要件	ワイルドカードとしての「*」文字は使用できない。
親要素	weblogic-rdbms-bean query-method
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

method-name 要素では、ファインダまたは ejbSelect メソッドの名前を指定します。

method-param

値の範囲	有効な名前
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-bean method-params
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

method-param 要素では、メソッド パラメータの Java 型名を指定します。型名は完全修飾形式で指定します。

method-params

値の範囲	有効な名前のリスト
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-bean query-method
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

method-params 要素では、メソッド パラメータの Java 型名の順序付きリストを指定します。型名は完全修飾形式で指定します。

query-method

値の範囲	なし
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-bean
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

query-method 要素では、weblogic-query と関連付けられるメソッドを指定します。この要素の指定には、デプロイメント記述子 ejb-jar.xml と同じ形式を使用します。

relation-name

値の範囲	有効な名前
デフォルト値	なし
要件	関連付けられた記述子ファイル ejb-jar.xml 内の ejb-relation 要素の子である ejb-relation-name と一致しなければならない。
親要素	weblogic-rdbms-relation
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

relation-name 要素では、関係の名前を指定します。

relationship-role-name

値の範囲	有効な名前
デフォルト値	なし
要件	関連付けられた記述子ファイル ejb-jar.xml 内の ejb-relationship-role 要素の子である ejb-relationship-role-name と一致しなければならない。
親要素	weblogic-rdbms-relation weblogic-relationship-role
デプロイメントファイル	weblogic-cmp-rdbms-jar.xml

機能

relationship-role-name 要素では、関係ロールの名前を指定します。

table-name

値の範囲	データベース内のソース テーブルの有効な SQL 名（完全修飾形式）
デフォルト値	なし
要件	table-name は常に設定しなければならない。
親要素	weblogic-rdbms-bean weblogic-rdbms-relation
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

テーブルの完全修飾 SQL 名。この Bean の data-source に対して定義されるユーザは、このテーブルに対して読み取りおよび書き込みの権限を持つ必要がありますが、スキーマ変更の権限は必ずしも必要ではありません。

weblogic-ql

値の範囲	なし
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-bean weblogic-query
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

weblogic-ql 要素には、EJB QL 言語に対する WebLogic 固有の機能拡張を含むクエリを指定します。EJB QL 言語の標準機能だけを使用するクエリについては、デプロイメント記述子 ejb-jar.xml に指定することを推奨します。

weblogic-query

値の範囲	なし
デフォルト値	なし
要件	なし
親要素	weblogic-rdbms-bean
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

weblogic-query 要素では、WebLogic 固有の属性を必要に応じてクエリと関連付けることができます。たとえば、weblogic-query 要素を使って、EJB QL に対する WebLogic 固有の機能拡張を含むクエリを指定することができます。EJB QL に対する WebLogic の機能拡張を利用しないクエリについては、デプロイメント記述子 ejb-jar.xml に指定することを推奨します。

また、weblogic-query 要素には、クエリによって事前にキャッシュにロードしておくことが望ましいエンティティ Bean をクエリが取得する場合に、フィールド グループ（field-group 要素で指定）をクエリと関連付ける用途もあります。

weblogic-relationship-role

値の範囲	有効な名前
デフォルト値	なし
要件	テーブルへのロールのマッピングは、関連する weblogic-rdbms-bean および ejb-relation 要素で指定される。
親要素	weblogic-rdbms-relation
デプロイメント ファイル	weblogic-cmp-rdbms-jar.xml

機能

weblogic-relationship-role 要素は、外部キーから主キーへのマッピングを表現するために使われます。関係がローカルであるとき、1 対 1 関係に対する 1 つのマッピングだけが指定されます。多対多の関係については、2 つのマッピングを指定しなければなりません。

キーが複雑な場合、1 つのロールに対して複数の列マッピングが指定されます。ロールが単にグループ名を指定しているだけの場合、column-map は指定されません。

この節では、バージョン 6.1 の WebLogic Server に固有の XML 要素について説明します。これらの要素は weblogic510-cmp-rdbms-jar.xml ファイルを構成し、WebLogic Server EJB コンテナ内のデプロイメント記述子を定義するために使われるものです。これらの要素は、WebLogic Server Administration Console のほぼ同じ名前のフィールドに対応します。これらのデプロイメント記述子はバージョン 1.1 の EJB に対して使用します。

RDBMS 定義要素

ここでは、RDBMS 定義要素について説明します。

pool-name

pool-name 要素では、この EJB のデータベース接続で使われる WebLogic Server 接続プールの名前を指定します。

schema-name

schema-name 要素では、ソース テーブルがデータベース内に位置するスキーマを指定します。この要素は、EJB の接続プールに定義されたユーザにとってのデフォルト スキーマではないスキーマを使用する場合にのみ必要です。

注意 : このフィールドでは大文字 / 小文字が区別されますが、多くの SQL 実装では大文字 / 小文字の区別は無視されます。

table-name

table-name 要素では、データベース内のソース テーブルを指定します。この要素は常に必要です。

注意 : EJB の接続プールに定義されるユーザは、指定されたテーブルに対して読み取りおよび書き込みの権限を持つ必要がありますが、スキーマ変更の権限は必ずしも必要ではありません。このフィールドでは大文字 / 小文字が区別されますが、多くの SQL 実装では大文字 / 小文字の区別は無視されます。

EJB フィールド マッピング要素

ここでは、EJB フィールド マッピング要素について説明します。

attribute-map

attribute-map 属性は、EJB インスタンスの 1 つのフィールドをデータベース テーブルの特定の列とリンクします。attribute-map では、WebLogic Server の RDBMS ベースの永続性を利用する EJB の各フィールドに対して、厳密に 1 つのエントリを指定しなければなりません。

object-link

個々の attribute-map エントリは、データベースの列および EJB インスタンスのフィールド間のリンクを表現する object-link 属性で構成されます。

bean-field

bean-field 属性では、データベースから取得されたデータが入る EJB インスタンスのフィールドを指定します。この要素では大文字 / 小文字が区別され、要素の値は Bean インスタンスのフィールドの名前と厳密に一致しなければなりません。

また、この要素で参照されるフィールドに対応する cmp-field 要素が、Bean の ejb-jar.xml ファイルで定義されている必要があります。

dbms-column

dbms-column 要素では、EJB のフィールドがマッピングされるデータベースの列を指定します。この要素では大文字 / 小文字が区別されますが、多くのデータベースでは大文字 / 小文字の区別は無視されます。

注意： WebLogic Server では、dbms-column 要素の値として、引用符で囲んだ RDBMS キーワードを使用できません。たとえば、"create" や "select" のような列名を使った属性マップは、使用するデータストアでそれらの語が予約済みである場合には作成できません。

ファインダ要素

ここでは、ファインダ要素について説明します。

finder-list

finder-list 要素は、Bean 集合の位置を特定するために生成される全ファインダの集合を定義します。

finder-list には、findByPrimarykey メソッドを除き、ホーム インターフェースに定義される個々のファインダ メソッドに対して必ず 1 つのエントリを定義しなければなりません。findByPrimaryKey に対応するエントリが定義されない場合、コンパイル時にエントリが生成されます。

注意： findByPrimaryKey に対応するエントリを指定しない場合、WebLogic Server はコンパイル時に生成されるエントリを、妥当性を検証することなく使用します。ほとんどの場合、findByPrimaryKey に対応するエントリの定義を省略して、デフォルトで生成されるメソッドを使うことが推奨されます。

finder

finder 要素は、ホーム インターフェースに定義されるファインダ メソッドを定義します。WebLogic Server は finder 要素の子要素を参照することによって、ホーム インターフェースのどのメソッドが記述対象となっているかを識別し、必要なデータベース操作を実行できます。

method-name

method-name 要素では、ホーム インターフェースのファインダ メソッドの名前を定義します。この要素にはメソッドの正確な名前を指定しなければなりません。

method-params

method-params 要素では、method-name に指定されているファインダ メソッドに渡されるパラメータの集合を定義します。

注意： WebLogic Server はこのリストを、EJB のホーム インターフェースに定義されたファインダ メソッドのパラメータ型と照合します。パラメータ リストの順序および型は、ホーム インターフェースに定義された順序および型と厳密に一致しなければなりません。

method-param

method-param 要素では、パラメータの型の名前を完全修飾形式で定義します。型名は java.lang.Class オブジェクトに評価され、評価結果のオブジェクトは、EJB のファインダ メソッドの各パラメータと厳密に一致しなければなりません。

パラメータのプリミティブ名（double や int など）を使って、プリミティブ型のパラメータを指定することができます。プリミティブでないデータ型を method-param 要素で使用する場合は、完全修飾名を指定しなければなりません。たとえば、Timestamp ではなく java.sql.Timestamp と指定します。修飾名を使用しないと、デプロイメント ユニットのコンパイル時に ejbc によってエラー メッセージが返されます。

finder-query

finder-query 要素は、このファインダでデータベースから値を取得するために使われる WebLogic Query Language（WLQL）文字列を指定します。

注意： finder-query 要素の値であるテキストは、常に XML の CDATA 属性を使って定義します。CDATA を使うことにより、ファインダのコンパイル時に、WLQL 文字列内の特殊文字が原因でエラーが発生することがなくなります。

finder-expression

finder-expression 要素では、このファインダでのデータベース クエリで変数として使用する Java 言語拡張を指定します。

注意： WebLogic Server EJB コンテナの将来のバージョンでは、（EJB 2.0 仕様の要件に従って）EJB QL クエリ言語が使われる予定です。EJB QL では、埋め込み形式の Java 式を使用できません。そのため、将来の EJB コンテ

ナへのアップグレードを容易にするために、エンティティ EJB のファインダを作成するときは、WLQL に Java 式を埋め込まないようにしてください。

76 レルム

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前

77 セキュリティ

以下に、Administration Console を使用して、セキュリティの作成や管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[システム ユーザ]	サーバの起動と停止、リソースのロックとロックの解除など、WebLogic Server のシステム レベルの操作を管理する管理ユーザを返す。system ユーザは、WebLogic Server のインストール手順の中で定義する。	最大 256 文字の英数字で、カンマまたはスペースは使用不可	system
[キャッシング レルム]	セキュリティを設定または確認するキャッシング レルムを選択できる。	リスト	未選択
[接続フィルタ]	セキュリティ レルムの接続フィルタを選択できる。	リスト	未選択
[ゲスト不可]	guest ログインを使用して WebLogic Server のリソースおよびサービスにアクセスできるかどうかを指定する。	ブール True = 選択されている False = 選択されていない	選択

[ファイル レルム]

属性	説明	値の範囲	デフォルト値
[最大ユーザ数]	ファイル レルムと共に使用するユーザの最大数を指定する。ファイル レルムは、ユーザ数 10,000 人以下での使用を想定されている。	1 ～ 10000	1000
[最大グループ数]	ファイル レルムと共に使用するグループの最大数を指定する。	1 ～ 10000	1000
[最大 ACL]	ファイル レルムと共に使用する ACL の最大数を指定する。	1 ～ 10000	1000

[詳細設定]

属性	説明	値の範囲	デフォルト値
[すべてのログをチェック]	このボックスを選択すると、リソース上のすべての checkPermission が WebLogic Server のログ ファイルに通知メッセージとして記録される。この属性を有効にすると、無効にする場合よりも多くのメッセージが WebLogic Server のログ ファイルに含まれる。無効にすると、保護されたリソースへのアクセスがログ メッセージになる。		選択されていない
[結果バッチサイズ]	ユーザ、グループ、ACL を表示するときに、1 回の要求で返されるユーザ、グループ、ACL、およびパーミッションの数を指定する。	0 より大きい値	200

[監査]

属性	説明	値の範囲	デフォルト値
[監査プロバイダ クラス]	監査プロバイダを実装する Java クラスの名前を指定できる。監査プロバイダは `weblogic.security.audit.AuditProvider` インタフェースの実装。	有効な監査プロバイダ クラスの名前	NULL

[パスワード]

属性	説明	値の範囲	デフォルト値
[最小パスワード文字数]	パスワードに必要な文字数を設定する。パスワードは、8 文字以上でなければならない。	整数	8
[ロックアウト有効化]	無効なログインが行われたときにユーザ アカウントがロックされる。	ブール 有効 = 選択 無効 = 未選択	選択
[ロックアウトしきい値]	ユーザ アカウントがロックされるまで許されるユーザ名とパスワードの無効な組み合わせの数を指定する。それ以降にユーザ アカウントへのアクセスが試行された場合、入力されたユーザ名とパスワードの組み合わせが正しくても、セキュリティ例外が発生する。システム管理者によって明示的にロック解除されるか、[ロックアウト遅延] の時間の経過後に他のログインが試行されるまで、アカウントはロックされる。	整数	5 分

属性	説明	値の範囲	デフォルト値
[ロックアウト遅延]	[ロックアウト リセット遅延] 属性で指定された時間内に無効なログインが試行された後、ユーザのアカウントがロックされる分数を指定する。	整数	30 分
[ロックアウト リセット遅延]	無効なログインの試行によってユーザ アカウントがロックされるまでの分数を指定する。このフィールドで指定した時間内に、[ロックアウトしきい値] フィールドで指定した数の無効なログインが試行されると、アカウントがロックされる。	整数	5 分
[ロックアウト キャッシュ サイズ]	試行しなかったログインと試行した無効なログインのキャッシュ サイズを設定する。	0 ～ 10	5

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

78 サーバ

以下に、Administration Console を使用して、WebLogic Server のコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「WebLogic Server とクラスタのコンフィグレーション」および「WebLogic Server の起動と停止」を参照してください。

サーバのコンフィグレーション

1. 左ペインの [サーバ] ノードをクリックします。右ペインに [サーバ] テーブルが表示され、ドメインで定義されているすべてのサーバが示されます。
2. [新しい Server のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するサーバの設定に関連するタブが示されます。
3. [名前]、[マシン]、[リスン ポート]、[管理ポート]、[リスン アドレス]、および [外部 DNS 名] 属性フィールドに値を入力します。
4. 右下隅の [作成] ボタンをクリックして、[名前] フィールドに指定した名前でサーバ インスタンスを作成します。左ペインの [サーバ] ノードに、新しいインスタンスが追加されます。
5. その他のタブをクリックして、属性フィールドを変更するか、割り当てられているデフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

サーバのクローンの作成

1. 左ペインの [サーバ] ノードをクリックします。右ペインに [サーバ] テーブルが表示され、ドメインで定義されているすべてのサーバが示されます。

2. クローンを作成するサーバの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、新しいサーバのクローンの作成に関連するタブが示されます。
3. [名前]、[マシン]、[リスン ポート]、[管理ポート]、[リスン アドレス]、および [外部 DNS 名] 属性フィールドに値を入力します。
4. 右下隅の [クローン] ボタンをクリックして、[名前] フィールドに指定した名前でサーバのインスタンスを作成します。左ペインの [サーバ] ノードに、新しいインスタンスが追加されます。
5. その他のタブをクリックして、属性フィールドを変更するか、割り当てられているデフォルト値をそのまま使用します。
6. [適用] をクリックして、変更を保存します。

サーバの削除

1. 左ペインの [サーバ] ノードをクリックします。右ペインに [サーバ] テーブルが表示され、ドメインで定義されているすべてのサーバが示されます。
2. 削除するサーバの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、サーバを削除します。[サーバ] ノードの下にあるサーバ アイコンが削除されます。

サーバのログの表示

1. 左ペインの [サーバ] の下で、ログを表示するサーバのインスタンス ノードを右クリックします。ポップ アップ メニューが開きます。
2. [サーバ ログを見る] をクリックします。新しいブラウザ ウィンドウが表示され、ログが示されます。

サーバの JNDI ツリーの表示

1. 左ペインの [サーバ] の下で、JNDI ツリーを表示するサーバのインスタンス ノードを右クリックします。ポップアップ メニューが開きます。
2. [JNDI ツリーを見る] をクリックします。新しいブラウザ ウィンドウが表示され、ネーミング コンテキスト データが示されます。

サーバの実行キューの表示

1. 左ペインの [サーバ] の下で、実行キューを表示するサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [すべてのアクティブなキューのモニタ ...] テキスト リンクをクリックします。右ペインに [実行キュー] テーブルが表示され、このサーバ用に定義されているすべての実行キューが示されます。

サーバの実行スレッドの表示

1. 左ペインの [サーバ] の下で、実行スレッド データを表示するサーバのインスタンス ノードを右クリックします。ポップ アップ メニューが開きます。
2. [実行スレッドを見る] をクリックします。情報が右ペインに表示されます。

サーバのソケットの表示

1. 左ペインの [サーバ] の下で、ソケットを表示するサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [すべてのアクティブなソケットのモニタ] リンクをクリックします。右ペインに [ソケット] テーブルが表示され、このサーバ用に定義されているすべてのソケットが示されます。

サーバの接続の表示

1. 左ペインの [サーバ] の下で、接続を表示するサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [すべての接続のモニタ] リンクをクリックします。右ペインに [接続] テーブルが表示され、このサーバ用に定義されているすべての接続が示されます。

サーバ上でのガベージ コレクションの実行

1. 左ペインの [サーバ] の下で、メモリ使用状況を表示するサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [パフォーマンス] タブをクリックします。

4. [メモリ使用状況] グラフが高い使用率を示しているかどうかをチェックします。[メモリ使用状況] グラフは、選択したサーバが現在動作している場合にだけ表示されます。
5. [ガベージ コレクションを強制する] ボタンをクリックして、ガベージ コレクションを実行します。コレクションが正常に実行されたことを示すメッセージが表示されます。

JTA のモニタ

1. 左ペインの [サーバ] の下で、JTA の使用状況を表示するサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [JTA] タブをクリックします。JTA の読み込み専用の統計が右ペインに表示されます。

サーバ セキュリティのモニタ

1. 左ペインの [サーバ] の下で、セキュリティをモニタするサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [セキュリティ] タブをクリックします。このインスタンスのセキュリティ データが表示されます。

サーバのバージョンの表示

1. 左ペインの [サーバ] の下で、バージョンを表示するサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [バージョン] タブをクリックします。このインスタンスのバージョン データが表示されます。

サーバのクラスタのモニタ

1. 左ペインの [サーバ] の下で、クラスタをモニタするサーバのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [モニタ] タブをクリックします。
3. [クラスタ] タブをクリックします。このインスタンスのクラスタ データが表示されます。

サーバ上での EJB のデプロイ

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、EJB デプロイメント用のサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [デプロイメント] タブをクリックします。[EJB] タブが表示されます。
3. [選択可] カラムで、サーバにデプロイする 1 つまたは複数の EJB を選択します。

4. 移動コントロールをクリックして、選択した EJB を [選択済み] カラムに移動します。
5. [適用] をクリックして割り当てを保存します。

サーバにデプロイされたすべての EJB のモニタ

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、EJB モニタの対象となるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [デプロイメント] タブをクリックします。[EJB] タブが表示されます。
3. [すべてのアクティブな EJB のモニタ] テキスト リンクをクリックします。右ペインに [アクティブな EJB] テーブルが表示され、このサーバにデプロイされたすべての EJB が示されます。

サーバへの Web アプリケーション コンポーネントのデプロイ

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、Web アプリケーション デプロイメントの対象となるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [デプロイメント] タブをクリックします。
3. [Web アプリケーション] タブをクリックします。
4. [選択可] カラムで、サーバにデプロイする 1 つまたは複数の Web アプリケーションを選択します。
5. 移動コントロールをクリックして、選択した Web アプリケーションを [選択済み] カラムに移動します。

6. [適用] をクリックして割り当てを保存します。

サーバ上のすべての Web アプリケーション コンポーネントのモニタ

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、Web アプリケーション モニタの対象となるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [デプロイメント] タブをクリックします。
3. [Web アプリケーション] タブをクリックします。
4. [すべてのアクティブな Web アプリケーションのモニタ] テキスト リンクをクリックします。右ペインに [アクティブな Web アプリケーション] テーブルが表示され、このサーバにデプロイされたすべての Web アプリケーションが示されます。

サーバへの起動クラスおよび停止クラスのデプロイ

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、起動 / 停止クラスをデプロイするサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [デプロイメント] タブをクリックします。
3. [起動 / 停止] タブをクリックします。
4. [選択可] カラムで、サーバにデプロイする 1 つまたは複数の起動クラスを選択します。
5. 移動コントロールをクリックして、選択した起動クラスを [選択済み] カラムに移動します。

6. [適用] をクリックして割り当てを保存します。
7. [停止クラス] コントロールを使用して手順 4、5、および 6 を繰り返して、サーバに停止クラスをデプロイします。

サーバへの JDBC 接続プールの割り当て

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、JDBC 接続プールを割り当てるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サービス] タブをクリックします。
3. [JDBC] タブをクリックします。
4. [選択可] カラムで、サーバに割り当てる 1 つまたは複数の JDBC 接続プールを選択します。
5. 移動コントロールをクリックして、選択した JDBC 接続プールを [選択済み] カラムに移動します。
6. [適用] をクリックして割り当てを保存します。

サーバへの WLEC 接続プールの割り当て

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、WLEC 接続プールを割り当てるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サービス] タブをクリックします。
3. [WLEC] タブをクリックします。
4. [選択可] カラムで、サーバに割り当てる 1 つまたは複数の WLEC 接続プールを選択します。

5. 移動コントロールをクリックして、選択した WLEC 接続プールを [選択済み] カラムに移動します。
6. [適用] をクリックして割り当てを保存します。

サーバ上のすべての WLEC 接続プールのモニタ

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、WLEC 接続プールのモニタの対象となるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サービス] タブをクリックします。
3. [WLEC] タブをクリックします。
4. [すべてのアクティブ プールのモニタ] テキスト リンクをクリックします。右ペインに [WLEC 接続プール] テーブルが表示され、このサーバに割り当てられているすべての接続プールが示されます。

サーバへの XML レジストリの割り当て

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、XML レジストリを割り当てるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サービス] タブをクリックします。
3. [XML] タブをクリックします。
4. [XML レジストリ] ドロップダウン リスト ボックスからレジストリを選択します。
5. 残りのフィールドに新しい値を入力するか、デフォルト値をそのまま使用します。

6. [適用] をクリックして割り当てを保存します。

サーバへのメール セッションの割り当て

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、メール セッションを割り当てるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サービス] タブをクリックします。
3. [Mail] タブをクリックします。
4. [選択可] カラムで、サーバに割り当てる 1 つまたは複数のメール セッションを選択します。
5. 移動コントロールをクリックして、選択したメール セッションを [選択済み] カラムに移動します。
6. [適用] をクリックして割り当てを保存します。

サーバへの FileT3 の割り当て

1. 左ペインの [サーバ] の下のインスタンス ノードをクリックして、FileT3 を割り当てるサーバを選択します。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サービス] タブをクリックします。
3. [FileT3] タブをクリックします。
4. [選択可] カラムで、サーバに割り当てる 1 つまたは複数の FileT3 を選択します。
5. 移動コントロールをクリックして、選択した FileT3 を [選択済み] カラムに移動します。
6. [適用] をクリックして割り当てを保存します。

WebLogic Server Console 内のコンパイラの変更

1. WebLogic Server Console を起動します。
2. ナビゲーション ツリーの [サーバ] フォルダを開きます。
3. [サーバ] フォルダでサーバを選択します（インストール時のデフォルトでは myserver）。
4. [コンフィグレーション] タブをクリックします。
5. [コンパイラ] タブをクリックして、[Java コンパイラ] テキスト ボックスに sj.exe コンパイラのフル パスを入力します。
   例：`c:\visualcafe31\bin\sj.exe`
6. JRE rt.jar ライブラリへのフル パスを [クラスパスの後ろに追加] テキスト ボックスに入力します。
   例： `\%WL_HOME%\jdk130\jre\lib\rt.jar`
7. [適用] をクリックします。
8. サーバを再起動して、新しい Java コンパイラと [クラスパスの後ろに追加] の値を有効にします。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	サーバ名を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	myserver
[マシン]	マシン名を返す。	カンマまたはスペースを含まない英数字の文字列	NULL
[リスン ポート]	このサーバが SSL メッセージをリスンするために使用するポートを指定する。	有効なリスン ポート	7001

属性	説明	値の範囲	デフォルト値
[管理ポート]	管理サーバが管理関連のすべての通信を受信するために使用する SSL ポートを指定する。 管理サーバ上で値を指定する場合、すべての管理対象サーバは管理サーバへの情報の送信にポート番号（および SSL）を使用する。また、ユーザはすべての管理コマンドでこのポート番号と SSL を使用しなければならない。たとえば、管理サーバ上の Administration Console にアクセスするためにポート番号を使用しなければならない。 値を指定しない場合、管理サーバは管理関連の通信を受信するためにデフォルトのリスン ポートを使用する。	1 ～ 65534 の範囲内で未使用の任意のポート番号	NULL
[リスン アドレス]	ドメインのリスン アドレスを設定できる。	有効なリスン アドレス	NULL

属性	説明	値の範囲	デフォルト値
[外部 DNS 名]	システムにアドレス変換ファイアフォールが含まれていて、クラスタ化された WebLogic Server と フロントエンド Web サーバへのプラグイン（Netscape（プロキシ）プラグインなど）との間に位置する場合、このサーバとの通信でプラグインが使用するアドレスをこの属性に設定する。		
[インターフェイス アドレス]			NULL

[クラスタ]

属性	説明	値の範囲	デフォルト値
[クラスタ]	このサーバが所属するクラスタを指定する。	有効なクラスタ名	なし
[レプリケーション グループ]	このクラスタが所属するレプリケーション グループを指定する。	文字列	なし
[セカンダリ プリファレンス グループ]	このクラスタのセカンダリ プリファレンス グループを指定する。	文字列	なし
[クラスタの重み]	このサーバで実行される作業の割合を指定する。	整数	100

[HTTP]

属性	説明	値の範囲	デフォルト値
[デフォルト Web アプリケーション]	このサーバのデフォルトの Web アプリケーションの名前を返す。	文字列	DefaultWebApp_myserver
[POST タイムアウト秒]	WebLogic Server が HTTP POST データ中の連続データを受信する間隔のタイムアウトを秒単位で設定する。POST データを使用してサーバを過負荷状態にしようとするサービス拒否攻撃を防ぐために使用する。	整数	0
[デフォルト サーバ名]	WebLogic Server では、リクエストをリダイレクトするときに、[デフォルト サーバ名] で指定した文字列を使用して、HTTP 応答ヘッダとして返されるホスト名を設定する。 ファイアウォールまたはロード バランサを使用し、ブラウザからリクエストをリダイレクトして、元のリクエストで送られたものと同じホスト名を参照する場合に便利。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[Send Server Header を有効化]	false に設定すると、サーバ名は HTTP 応答と共に送られない。ヘッダの領域が限られる無線アプリケーションの場合に便利。	ブール True = 有効 False = 無効	True
[WAP 有効化]	この属性を選択すると、セッション ID は JVM 情報に含まれなくなる。これは、URL のサイズが 128 文字に制限される WAP デバイスで URL 書き換えを使用する場合に必要になる。[WAP 有効化] を選択すると、クラスタでのレプリケートされたセッションの使用に影響することがある。	有効 無効	無効
[最大 POST 時間]	WebLogic Server が HTTP POST データ中の連続データを待機する時間を秒単位で設定する。	整数 0 より小さい値は、時間が無制限であることを示す。	-1
[最大 POST サイズ]	HTTP POST データ中の連続データの最大サイズを設定する。	整数 0 より小さい値は、サイズが無制限であることを示す。	-1
[Keep Alive を有効化]	HTTP キープアライブを有効にするかどうかを設定する。	ブール True = 有効 False = 無効	選択

属性	説明	値の範囲	デフォルト値
[持続時間]	非アクティブな HTTP 接続を閉じるまで WebLogic Server が待機する秒数。	整数	30
[HTTPS 持続時間]	非アクティブな HTTPS 接続を閉じるまで WebLogic Server が待機する秒数。	整数	60

[SSL]

属性	説明	値の範囲	デフォルト値
[有効化]	サーバ間 SSL 接続を有効にする。サーバ間の自動 SSL をオーバライドするには、この属性を無効にする。	ブール True = 選択されている False = 選択されていない	選択
[リスン ポート]	WebLogic Server が SSL 接続をリスンするために使用する専用ポートを指定する。	有効なリスン ポート	7002
[サーバ キー ファイル名]	WebLogic Server のプライベートなキー ファイル名。	文字列	config\mydomain\demokey.pem
[サーバ認証ファイル名]	WebLogic Server のプライベートなキー ファイル名を指定する。	文字列	config\mydomain\democert.pem
[サーバ認証チェーン ファイル]	WebLogic Server のデジタル証明書の完全なディレクトリ位置を指定する。	文字列	config\mydomain\ca.pem
[クライアント認証を強制する]	この属性を true に設定すると、クライアント証明書が使用される。	ブール True = 選択されている False = 選択されていない	選択されていない

属性	説明	値の範囲	デフォルト値
[信頼性のある CA ファイル名]	WebLogic Server によって信頼されている証明局のデジタル証明書が格納されているファイルの名前を指定する。	証明機関の 1 つのデジタル証明書またはデジタル証明書チェーンが収められているファイル。	trusted-ca.pem
[認可済み認証機関]	証明書の有効性を調べるために使用する証明書認証者を指定する。	有効な認証者	NULL
[暗号化キーを使用]	WebLogic Sever のプライベート キーをパスワードを使用して暗号化するかどうかを指定する。 ♦ true に設定する場合、プライベートキーを使用するにはパスワードが必要になる。 ♦ false に設定する場合、プライベートキーは暗号化されず、パスワードを提供せずに使用できる。	選択されている = True 選択されていない = False	選択されていない

属性	説明	値の範囲	デフォルト値
[Java を使用]	ネイティブ Java ライブラリを使えるようにする。WebLogic Server では SSL プロトコルの pure-Java 実装を提供する。ネイティブ ライブラリを使用すると、Solaris、Windows NT、および IBM AIX プラットフォーム上で SSL 処理のパフォーマンスが向上する。	選択されている = True 選択されていない = False	選択

属性	説明	値の範囲	デフォルト値
[ハンドラを有効化]	WebLogic Server が、以下のいずれかの理由によりクライアント認証に失敗した SSL 接続を拒否するかどうかを指定する。 ♦ 要求したクライアント デジタル証明書が提供されなかった。 ♦ クライアントがデジタル証明書を提示しなかった。 ♦ クライアントのデジタル証明書は、[信頼性のある CA ファイル名] 属性によって指定された証明局から発行されたものではなかった。 デフォルトでは、ある WebLogic Server から別の WebLogic Server への SSL 接続の開始が SSL ハンドラによって実現する。たとえば、WebLogic Server の EJB により、別の Web サーバの HTTP ストリームを開くことができる。[ハンドラを有効化] 属性が有効であれば、WebLogic Server は SSL 接続におけるクライアントとして動作する。	選択されている = True 選択されていない = False	選択

属性	説明	値の範囲	デフォルト値
[キーの有効期間をエクスポート]	WebLogic Server がドメスティック サーバとエクスポータブル クライアントとの間で、新規のキーを生成する前に、エクスポータブル キーを使用できる回数を指定する。新規のキーを生成する前のキーの使用回数が少ないほど、WebLogic Server のセキュリティは向上する。	最大値は `java.lang.Integer.MAX_VALUE` として指定される。最小値は 1。	500
[ログイン タイムアウト ミリ秒]	SSL 接続のタイム アウトまで WebLogic Server が待機する時間をミリ秒で設定する。SSL 接続は通常の接続よりもネゴシエーションに時間がかかる。クライアントがインターネット経由で接続する場合は、ネットワーク レイテンシの増加に対応するためにデフォルト値を大きくする。値 0 を指定するとこの属性は無効になる。	最大値は `java.lang.Integer.MAX_VALUE` として指定される。最小値は 1。	25000
[認可キャッシュ サイズ]	トークンによって使用されていないデジタル証明書の数を設定する。	最大値は `java.lang.Integer.MAX_VALUE` として指定される。最小値は 1。	3

属性	説明	値の範囲	デフォルト値
[ホスト名検証を無視]	WebLogic Server が別の WebLogic Server のクライアントとして動作する場合、インストールされたホスト名検証を無効にする。		選択されていない
[ホスト名の検証]	ホスト名検証インタフェースを実装する Java クラスの名前を設定する。	文字列	NULL
[Two-way SSL Enabled 相互 SSL を有効化]	相互認証を選択できるようにするために、相互 SSL を設定できる。クライアント証明書が提示された場合は、相互認証が行われる。提示されない場合は、クライアント証明書を要求しないで接続を受け入れる。	ブール 選択されている = 相互 SSL は有効 選択されていない = 相互 SSL は無効	選択されていない

[チューニング]

属性	説明	値の範囲	デフォルト値
[ソケット リーダー]	ソケット リーダーとして使用可能なスレッドの割合を設定する。	整数	33%
[ログイン タイムアウト]	このサーバのログイン タイムアウトを設定する。	整数	1000 ms
[バックログを受け入れ]	バックログに使用できる接続数を返す。処理される接続数を増やすには、この属性の数を増やす。	整数	50
[許可されたリバース DNS]	このサーバで DNS 逆引き参照を行うかどうかを指定する。	ブール True = 選択されている False = 選択されていない	選択されていない
[ネイティブ IO を有効化]	このサーバ用にネイティブ I/O を有効にするかどうかを指定する。	ブール 有効 = 選択 無効 = 未選択	選択
[ワークスペースにユーザのキーのみを表示]	ユーザ キーを表示するかどうかを指定する。	ブール True = 選択されている False = 選択されていない	選択されていない

属性	説明	値の範囲	デフォルト値
[デフォルト JMS 接続ファクトリを有効化]	このサーバ用にデフォルト JMS 接続ファクトリを有効にするかどうかを指定する。	ブール 有効 = 選択 無効 = 未選択	選択
[トンネリングを有効化]	このサーバ用にトンネリングを有効にするかどうかを指定する。	ブール 有効 = 選択 無効 = 未選択	選択されていない
[トンネリング クライアント Ping]	サーバがクライアントを ping するまでの待ち時間を秒単位で設定する。	整数	45
[トンネリング クライアント タイムアウト]	サーバがタイムアウトするまでの待ち時間を秒単位で設定する。	整数	40

[プロトコル]

属性	説明	値の範囲	デフォルト値
[デフォルト プロトコル]	このサーバが使用するデフォルトのプロトコルを設定する。	リスト	t3
[デフォルト セキュア プロトコル]	このサーバが使用するデフォルトのセキュリティ プロトコルを設定する。	リスト	t3s
[最大 T3 メッセージ サイズ]	メッセージの最大サイズをバイト単位で設定する。	整数	10000000
[T3 メッセージ タイムアウト]	メッセージがタイムアウトするまでの時間を秒単位で設定する。	整数	480
[最大 HTTP メッセージ サイズ]	メッセージの最大サイズをバイト単位で設定する。	整数	10000000
[HTTP メッセージ タイムアウト]	メッセージがタイムアウトするまでの時間を秒単位で設定する。	整数	480

属性	説明	値の範囲	デフォルト値
[IIOP の有効化]	このサーバ用に IIOP を有効にするかどうかを指定する。	ブール 有効 = 選択 無効 = 未選択	選択
[最大 IIOP メッセージ サイズ]	メッセージの最大サイズをバイト単位で設定する。	整数	10000000
[IIOP メッセージ タイムアウト]	メッセージがタイムアウトするまでの時間を秒単位で設定する。	整数	480
[デフォルト IIOP パスワード]	このサーバのデフォルトの IIOP パスワードを設定する。	有効なパスワード	*****
[デフォルト IIOP ユーザ]	このサーバのデフォルトの IIOP ユーザを設定する。	管理者が設定したユーザ名	guest

[コンパイラ]

属性	説明	値の範囲	デフォルト値
[Java コンパイラ]	このサーバが使用する Java コンパイラを設定する。	有効な Java コンパイラ	javac
[クラスパスの前に追加]	この属性がクラスパスの先頭に追加される。	クラス パス情報	NULL
[クラスパスの後ろに追加]	この属性がクラスパスの最後に追加される。	クラス パス情報	NULL
[追加 Rmic オプション]	このサーバの rmic オプションを設定する。	rmic オプション	NULL

[モニタ]

[一般]

属性	説明	値の範囲	デフォルト値
[状態]	サーバの状態を示す。		Running
[アクティブ化時刻]	最後のアクティブ化の日付を返す。	日付	最後にアクティブ化した日付

[パフォーマンス]

属性	説明	値の範囲	デフォルト値
[要求スループット]	このサーバによる要求の処理速度を返す。		なし
[要求待ち時間]	サービス待ちのリクエスト数を返す。		なし
[Request Wait Time]	最も古いリクエストがサービスを待機している時間を返す。		
[Total Requests Serviced]	このサーバによって処理された総要求数を返す。		
[メモリ使用状況]	サーバが使用しているメモリの量を示す。		なし

[メモリ]

属性	説明	値の範囲	デフォルト値
[メモリ使用状況]	サーバが使用しているメモリの量を返す。	整数	なし
[割り当て済みメモリ]	このサーバに割り当てるメモリの量を設定する。	整数	66846720

[クラスタ]

属性	説明	値の範囲	デフォルト値
[生存サーバ数]	現在稼働しているサーバの数を返す。		なし
[再送リクエスト数]	情報の再送信の要求数を返す。		なし
[送信したフラグメント数]	送信された情報の断片の数を返す。		なし
[受信したフラグメント数]	受信された情報の断片の数を返す。		なし
[失われたマルチキャスト メッセージ数]	失われたマルチキャスト メッセージの数を返す。		なし
[サーバ名]	現在稼働しているサーバの名前を返す。		なし
[プライマリ数]	サーバの一次配布名を返す。		なし
[二次的送り先名]	サーバの二次配布名を返す。		なし

[セキュリティ]

属性	説明	値の範囲	デフォルト値
[ロックアウトされているユーザの総数]	サーバからロックアウトされたユーザの数を返す。	整数	0
[不正なログイン総数]	無効なログインの試行回数を返す。	整数	0
[ロック中に試行されたログイン数]	サーバがロックされている間に試行されたログインの数を返す。	整数	0
[ロック解除されているユーザの総数]	ログインのロックが解除されたユーザの数を返す。	整数	0
[不正なログインの最大回数]	無効なログインの数を返す。	整数	0
[ロックされているユーザ数]	ログインがロックされたユーザの数を返す。	整数	0

[JMS]

属性	説明	値の範囲	デフォルト値
[接続数]	現在の接続数をレポートする。	整数	0
[最大接続数]	最大接続数をレポートする。	整数	0
[接続総数]	接続の総数をレポートする。	整数	0
[JMS サーバ数]	現在稼働している JMS サーバの数を返す。	整数	0
[最大 JMS サーバ数]	JMS サーバの最大数をレポートする。	整数	0
[JMS サーバ総数]	JMS サーバの総数をレポートする。	整数	0

[JTA]

属性	説明	値の範囲	デフォルト値
[トランザクション総数]	処理されたトランザクションの総数。この総数には、コミットされたトランザクション、ロールバックされたトランザクション、およびヒューリスティクなトランザクションがすべて含まれる。	整数	0
[コミット総数]	コミットされたトランザクションの数	整数	0
[ロールバック総数]	ロールバックされたトランザクションの数。	整数	0
[タイムアウト ロールバック数]	タイムアウトのためにロールバックされたトランザクションの数。	整数	0
[リソース ロールバック数]	リソース エラーのためにロールバックされたトランザクションの数。	整数	0

属性	説明	値の範囲	デフォルト値
[アプリケーション ロールバック数]	アプリケーション エラーのためにロールバックされたトランザクションの数。	整数	0
[システム ロールバック数]	システム エラーのためにロールバックされたトランザクションの数。	整数	0
[ヒューリスティック総数]	ヒューリスティックなステータスで完了したトランザクションの数。	整数	0
[トランザクション中止総数]	コミットされる前に中止されたトランザクションの数。	整数	0
[平均コミット時間]	コミットされたすべてのトランザクションに費やされた総ミリ秒数。	整数	0

[バージョン]

属性	説明	値の範囲	デフォルト値
[WebLogic のバージョン]	現在使用している WebLogic Server のバージョンを返す。		現在使用しているバージョン
[JDK ベンダ]	サーバで使用されている JDK のベンダの名前を返す。		NULL
[JDK バージョン]	サーバで使用されている JDK のバージョンを返す。		NULL
[オペレーティング システム]	サーバのオペレーティング システムを返す。		NULL
[OS のバージョン]	サーバのオペレーティング システムのバージョンを返す。		NULL
[J2EE12Only モードを有効化]			選択されていない
[J2EE13Warning を有効化]			選択されていない

[Entity]

属性	説明	値の範囲	デフォルト値
[Total Current Entries]	サーバの現在のエントリの総数を示す。		0
[Total Current Persistent Entries]	サーバの現在の永続エントリの総数を示す。		0
[Average Percentage Persistent]	サーバの永続エントリの平均を示す。		0
[Total Transient Current Entries]			0
[Average Percentage Transient]	サーバの一時エントリの平均パーセンテージを示す。		0
[Minimum Entry Timeout]			0
[Maximum Entry Timeout]			0
[Average Timeout]			0
[Total Current Session Entries]			0
[Total Persistent Current Session Entries]			0
[Average Percentage Session Persistent]			0
[Total Transient Current Session Entries]			0

属性	説明	値の範囲	デフォルト値
[Average Percentage Session Transient]			0
[Minimum Entry Session Timeout]			0
[Maximum Entry Session Timeout]			0
[Average Session Timeout]			0
[Maximum Entry Memory Size]			0
[Minimum Entry Memory Size]			0
[Average Per Entry Memory Size]			
[Average Per Entry Disk Size]			
[Total Number of Rejections]			
[Total Size of Rejections]			
[Percent Rejected]			
[Total Number of Renewals]			
[Total Historical Current Entries]			
[Total Historical Persistent Current Entries]			

属性	説明	値の範囲	デフォルト値
[Historical Average Percentage Persistent]			
[Historical Total Transient Current Entries]			
[Historical Average Percent Transient]			
[Historical Minimum Entry Timeout]			
[Historical Maximum Entry Timeout]			
[Historical Average Entry Timeout]			
[Historical Maximum Entry Memory Size]			
[Historical Minimum Entry Memory Size]			
[Historical Average Per Entry Memory Size]			
[Historical Average Per Entry Dixk Size]			
[Historical Total Number of Rejections]			
[Historical Total Size of Rejections]			
[Historical Percent Rejected]			
[Historical Total Number of Renewals]			

属性	説明	値の範囲	デフォルト値
[Current Maximum Entry Memory Size]			
[Current Minimum Entry Memory Size]			
[Current Average Per Entry Memory Size]			
[Current Average Per Entry Disk Size]			
[Current Memory Usage]			
[Current Disk Usage]			

[ログ]

[一般]

属性	説明	値の範囲	デフォルト値
[File Name ファイル名]	ログ メッセージをディスクに書き込むために使用するファイルの名前を指定する。	ファイル名には、最大 256 文字の英数字を使用できる。	*serverName*.log
[Log to StdoutStdout へログ出力]	サーバが、ログ ファイル以外にも、標準出力にメッセージを送信できるようにする。[Debug to stdoutStdout へデバッグ情報出力] および [Stdout severity thresholdStdout 重大度しきい値] を使用して、サーバから標準出力に送信するメッセージのタイプを指定する。	ブール True = 選択 False = 未選択	選択
[Debug to stdoutStdout へデバッグ情報出力]	サーバが、ログ ファイル以外にも、重要度 [Debug デバッグ] のメッセージを標準出力に送信するかどうかを指定する。このプロパティを有効にするには、[Log to StdoutStdout へログ出力] を有効にする必要がある。	ブール True = 選択 False = 未選択	未選択

属性	説明	値の範囲	デフォルト値
「Stdout severity thresholdStdout 重大度しきい値]	サーバが標準出力に送信するメッセージの重要度の最低レベル。このプロパティを有効にするには、[Log to StdoutStdout へログ出力] を有効にする必要がある。	♦ [InfoInfo]。通常の操作の報告に使用。 ♦ [WarningWarning]。疑わしい操作やコンフィグレーションが発生したが、通常の操作には影響しないと思われる。 ♦ [ErrorError]。ユーザ エラーが発生した。システムまたはアプリケーションは、割り込みやサービスの停止をせずにエラーに対処できる。 ♦ [NoticeNotice]。[InfoInfo] または [WarningWarning] レベルのメッセージで、サーバの監視に特に重要なもの。 ♦ [CriticalCritical]。システム エラーまたはサービス エラーが発生した。システムは回復できるが、サービスが一時的に停止するか、永続的に停止する恐れがある。 ♦ [AlertAlert]。システムの他の部分は機能しているが、あるサービスが使用不能な状態であることを示す。自動回復できないので、管理者が直ちに問題を解決する必要がある。 ♦ [EmergencyEmergency]。サーバが使用不能な状態であることを示す。深刻なシステム障害または危機的状態を示す。	Error

[ローテーション]

属性	説明	値の範囲	デフォルト値
[Rotation Type ローテーション タイプ]	古いログ メッセージを別のファイルに移行するための条件。 サーバがファイル名を変更した後、その後のメッセージは、新しいファイルに [Logging ログ |General 一般] タブの [File Name ファイル名] で指定したファイル名を付けて蓄積される。	♦ [None なし]。メッセージは、単一のファイルに蓄積される サイズが大きくなりすぎたら、ファイルの内容を消去する必要がある。 ♦ [by Size サイズ]。ログ ファイルが [File Min Size 最小ファイルサイズ] で指定したサイズに達すると、サーバは、ファイル名を FileName.*n* に変更する。 ♦ [by Time 時間]。[File Time Span ファイルローテーション間隔] で指定した時間間隔ごとに、サーバは、ファイル名を FileName.*n* に変更する。	なし

属性	説明	値の範囲	デフォルト値
[File Min Size 最小ファイルサイズ]	新しいログ ファイルが作成されるしきい値を設定する。 [Rotation Type ローテーション タイプ] が [by Size サイズ] の場合にのみ有効。	整数	500
[Rotation Time ローテーション開始時刻]	時間ベースのローテーションの流れについて、開始日時を指定する。この値で指定した日時に、サーバは現在のログ ファイルの名前を FileName.*n* に変更する。その後、サーバは、[File Time Span ファイルローテーション間隔] で指定した間隔でログ ファイル名を変更する。「毎月曜の 09:00」のように反復的な開始日時や、「2002 年 1 月 9 日の 09:00 に」のような非反復的な開始日時を作成することも可能。 [Rotation Type ローテーション タイプ] が [by Time 時間] の場合にのみ有効。	MM-dd-yyyy-k:mm:ss という java.text.SimpleDateFormat 形式を使用して、日時を指定する。この形式の詳細については、『J2EE Javadoc』を参照のこと。 指定した日時をすでに過ぎてしまった場合、その曜日 (E)、時 (H)、分 (m)、および秒 (s) と、現在の日時から開始日時を再計算する。	00:00 デフォルトでは、ローテーション サイクルは毎日の始まりの時刻 (00:00 AM) に始まるが、ローテーション サイクルは特定の曜日または特定の日付で開始するようコンフィグレーションできる。
[File Time Span ファイルローテーション間隔]	新しいログ ファイルが作成されるしきい値を設定する。 [Rotation Type ローテーション タイプ] が [by Time 時間] の場合にのみ有効。	整数	24

属性	説明	値の範囲	デフォルト値
[Number of Files Limited ファイル数の制限]	ログ ファイルをローテーションさせる場合に、サーバが作成するファイル数を制限するかどうかを指定する。制限は、[File Count ファイル数] の値に基づく。	ブール True = 選択 False = 未選択	未選択
[File Count ファイル数]	ログをローテーションさせるとき、サーバが作成するログ ファイルの最大数。[Number of Files Limited ファイル数の制限] が [true]、かつ、[Rotation Type ローテーション タイプ] が [by Size サイズ] または [by Time 時間] の場合のみ有効。	整数	7

[ドメイン]

属性	説明	値の範囲	デフォルト値
[Log to Domain Logfile ドメイン ログファイルにログを書き込む]	このサーバが、メッセージを（自分自身のログを採る他に）ドメイン ログに送るかどうかを指定する。	ブール 有効 = 選択 無効 = 未選択	選択
[Domain Log Filter ドメイン ログ フィルタ]	このサーバが、ドメイン ログにどのメッセージを送るかを指定する。[none なし] を指定すると、サーバは重要度が [Error] およびそれ以上のメッセージをすべて送る。 このプロパティは、[Log To Domain Log ドメイン ログファイルにログを書き込む] ファイルが有効な場合にのみ有効。	そのドメイン用に定義された すべての [Domain Log Filters ドメイン ログ フィルタ] を含む。サーバが使用できる [Domain Log Filters ドメイン ログ フィルタ] は 1 つだけ。	なし

[HTTP]

属性	説明	値の範囲	デフォルト値
[Enable Logging ログを有効化]	サーバが HTTP リクエストをログするかどうかを指定する。リクエストは、独立した HTTP ログ ファイルに保存される。	ブール 有効 = 選択 無効 = 未選択	選択
[Logfile Name ログファイル名]	HTTP ログ ファイルの完全なログファイル名を設定する。	完全修飾ログファイル名	./config/mydomain/logs/access.log
[Format フォーマット]	HTTP ログ ファイルで使用するフォーマットを設定する。どちらのフォーマットも、W3C によって定義されたもの。拡張ログ フォーマットでは、ログ ファイル内にサーバ ディレクティブを使用して、サーバが記録する情報をカスタマイズする。	[Commoncommon]、[Extendedextended]	[Commoncommon]

属性	説明	値の範囲	デフォルト値
[Log Buffer Size ログ バッファ サイズ]	HTTP リクエストを格納するバッファの最大サイズ（キロバイト）。バッファがこのサイズに達すると、サーバはそのデータを HTTP ログ ファイルに書き出す。[Flush Every 更新間隔] プロパティを使用して、サーバがバッファ サイズをチェックする頻度を設定する。	整数（KB） 0-1024	8
[Max Log File SizeK Bytes最大ログファイル サイズ]	ローテーションされるログ ファイルの最大サイズを設定する。	整数（KB） 0 に設定すると、ファイルの最大サイズは無制限になる。	0
[Rotation Type ローテーション タイプ]	古い HTTP リクエストを独立したログ ファイルに移行するための条件。 サーバがファイル名を変更した後、その後のメッセージは、新しいファイルに [Logfile Name ログ ファイル名] で指定したファイル名を付けて蓄積される。	♦ [size サイズ]。ログ ファイルが [Max Log File SizeK Bytes 最大ログファイル サイズ] に達すると、サーバは、ファイル名を *LogfileName.n* に変更する。 ♦ [date 時間]。[Rotation Period ローテーション間隔] で指定した時間間隔ごとに、サーバは、ファイル名を *LogFileName.n* に変更する。	サイズ

属性	説明	値の範囲	デフォルト値
[Rotation Period ローテーション間隔]	サーバが古い HTTP リクエストを別のファイルに保存する間隔(分)。 この値は、日付ベースのローテーション タイプを使用する場合にのみ有効。	整数（分）	2147483647
[Flush Every 更新間隔]	サーバが HTTP リクエストを格納するバッファのサイズをチェックする間隔（秒)。バッファが [Logfile BufferK Bytes ログ バッファ サイズ] プロパティに指定されているサイズを超えると、サーバはデータを HTTP リクエスト ログ ファイルに書き出す。	整数（秒数）	60

属性	説明	値の範囲	デフォルト値
[Rotation Time ローテーション時間]	時間ベースのローテーションの流れについて、開始日時を指定する。この値で指定した日時に、サーバは現在のログ ファイルの名前を *LogFileName.n* に変更する。その後、サーバは、[LogRotationPeriodMins ローテーション間隔] で指定した間隔でログ ファイル名を変更する。「毎月曜の 09:00」のように反復的な開始日時や、「2002 年 1 月 9 日の 09:00 に」のような非反復的な開始日時を作成することも可能。	MM-dd-yyyy-k:mm:ss という java.text.SimpleDateFormat 形式を使用して、日時を指定する。この形式の詳細については、『J2EE Javadoc』を参照のこと。 指定した日時をすでに過ぎてしまった場合、その曜日 (E)、時 (H)、分 (m)、および秒 (s) と、現在の日時から開始日時を再計算する。	NULL

[JDBC]

属性	説明	値の範囲	デフォルト値
[Enable JDBCLoggingJDBC ログ記録を有効化]	サーバが JDBC トランザクションをログするかどうかを指定する。トランザクションは、独立した JDBC ログファイルに保存される。	ブール True = 選択 False = 未選択されていない	未選択
[JDBCLog File NameJDBC ログ ファイル名]	JDBC ログ ファイルの名前を返す。		NULL

[JTA]

属性	説明	値の範囲	デフォルト値
[トランザクション ログファイルのプレフィックス]	トランザクション ログファイルのプレフィックスを設定する。		./

[デバッグ]

属性	説明	値の範囲	デフォルト値
[リモート例外のログ出力]	リモート例外をログに記録する場合に true に設定する。	ブール True = 選択されている False = 選択されていない	選択されていない
[スタック トレースのログ出力]	true に設定すると、例外メッセージにサーバ サイドのスタック トレースが含まれる。	ブール True = 選択されている False = 選択されていない	選択

[デプロイメント]

[EJB]

属性	説明	値の範囲	デフォルト値
[サーバの EJB を選択]	EJB のドロップダウン リストから、デプロイする EJB を選択する。複数の EJB を [選択可] リストから選択するには、〔Ctrl〕を押しながら EJB を選択する。	1 つまたは複数の使用できる EJB	NULL

[Web アプリケーション]

属性	説明	値の範囲	デフォルト値
[Web アプリケーション コンポーネント]	利用可能な Web アプリケーション コンポーネントを選択できる。	リスト	NULL

[JCA]

属性	説明	値の範囲	デフォルト値
[コネクタ コンポーネント]	利用可能なコネクタ コンポーネントを選択できる。	リスト	NULL

[仮想ホスト]

属性	説明	値の範囲	デフォルト値
[仮想ホスト]	利用可能な仮想ホスト コンポーネントを選択できる。	リスト	NULL

[起動 / 停止]

属性	説明	値の範囲	デフォルト値
[起動クラス]	リストから起動クラスを選択できる。	リスト	NULL
[停止クラス]	リストから停止クラスを選択できる。	リスト	NULL
[Server Start BEA Home]			NULL
[Server Start Root Directory]			NULL
[Server Start Class Path]			NULL
[Server Start Arguments]			NULL
[Server Start Security Policy File]			NULL

[サービス]

[JDBC]

属性	説明	値の範囲	デフォルト値
[JDBC 接続プール]	このコンフィグレーションで使用する接続プールを選択できる。	リスト	NULL

[WLEC]

属性	説明	値の範囲	デフォルト値
[WLEC 接続プール]	このコンフィグレーションで使用する WLEC 接続プールを選択できる。	リスト	NULL

[Jolt]

属性	説明	値の範囲	デフォルト値
[Jolt 接続プール]	このコンフィグレーションで使用する WLEC 接続プールを選択できる。	リスト	NULL

[JMS]

属性	説明	値の範囲	デフォルト値
[JMS サーバ]	このコンフィグレーションで使用する JMS サーバを選択できる。	リスト	NULL
[JMS 接続ファクトリ]	このコンフィグレーションで使用する JMS 接続ファクトリを選択できる。	リスト	NULL
[JMS 送り先]	このコンフィグレーションで使用する JMS 送り先を選択できる。	リスト	NULL

属性	説明	値の範囲	デフォルト値
[JMS スレッド プール サイズ]	JMS スレッド プールのサイズを指定する。 **注意 :** 受信された RMI 呼び出しは、JMS 実行キューまたはスレッド プールが存在する場合はその中で実行される。存在しない場合はデフォルトの実行キューで実行される。 追加の処理（初期の RMI スレッドで完了できなかった処理）はデフォルトの実行キューで実行される。 JMS 固有のスレッド プールを設定する場合、JMS スレッド プール はほかの実行スレッドから使用されず、ほかの実行スレッドは JMS スレッド プールから使用されないない点が異なる。	整数（0 以上） 5 より小さい値に設定すると、自動的に 5 に設定される	クライアント上のデフォルト サイズは 0（非 JMS スレッド プール） サーバ上のデフォルト サイズは 15（サーバ上は常に JMS スレッド プール）

[XML]

属性	説明	値の範囲	デフォルト値
[XML レジストリ]	このコンフィグレーションで使用する XML レジストリを選択できる。	リスト	NULL
[キャッシュ メモリ サイズ]	XML エンティティ キャッシュに使用できるディスクスペースのサイズを設定する。	整数（キロバイト）	0
[キャッシュ タイムアウト間隔]	キャッシュがタイムアウトする間隔を設定する。	整数（秒）	120

[メール]

属性	説明	値の範囲	デフォルト値
[メール セッション]	このコンフィグレーションで使用するメール セッションを選択できる。	リスト	NULL

[FileT3]

属性	説明	値の範囲	デフォルト値
[FileT3]	利用可能な FileT3 エンティティのリストから選択できる。	リスト	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

79 サーブレット

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- 名前
- 再ロード数
- 起動総数
- プール最大容量
- 実行時間総計
- 最長実行時間
- 最短実行時間
- 平均実行時間

80 サーブレット

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- 名前
- 再ロード数
- 起動総数
- プール最大容量
- 実行時間総計
- 最長実行時間
- 最短実行時間
- 平均実行時間

81 サーブレット セッション

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- 名前
- 再ロード数
- 起動総数
- プール最大容量
- 実行時間総計
- 最長実行時間
- 最短実行時間
- 平均実行時間

82 停止クラス

以下に、Administration Console を使用して、停止クラスのコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

停止クラスのコンフィグレーション

1. [起動と停止] ノードをクリックします。右ペインに [起動と停止] テーブルが表示され、ドメインで定義されているすべての停止クラスが示されます。
2. [新しい Shutdown Class のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しい停止クラスの設定に関連するタブが示されます。
3. [名前]、[クラス名]、[デプロイ順]、および [引数] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前で停止クラスのインスタンスを作成します。左ペインの [起動と停止] ノードの下に、新しいインスタンスが追加されます。

停止クラスのクローンの作成

1. [起動と停止] ノードをクリックします。右ペインに [起動と停止] テーブルが表示され、ドメインで定義されているすべての停止クラスが示されます。
2. クローンを作成する停止クラスの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、停止クラスのクローンの作成に関連するタブが示されます。
3. [名前]、[クラス名]、および [引数] 属性フィールドに値を入力します。

4. [作成] をクリックして、[名前] フィールドに指定した名前で停止クラスのインスタンスを作成します。左ペインの [起動と停止] ノードの下に、新しいインスタンスが追加されます。

停止クラスの削除

1. [起動と停止] ノードをクリックします。右ペインに [起動と停止] テーブルが表示され、ドメインで定義されているすべての停止クラスが示されます。
2. 削除する停止クラスの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして停止クラスを削除します。[起動と停止] ノードの下の停止クラス アイコンが削除されます。

停止クラスの割り当て

1. 左ペインの [起動と停止] の下で、割り当てる停止クラスのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. 停止クラスに割り当てる 1 つまたは複数のターゲットを [選択可] カラムで選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	停止クラスの名前を返す。	完全修飾クラス名	NULL
[クラス名]	クラス名を返す。		NULL
[引数]	停止クラスの引数を設定する。 複数の引数がある場合はカンマで区切る。 例 : `first=MyFirstName, last=MyLastName`		NULL
[デプロイ順]	クラスが実装される順番。		

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	このデプロイメントの対象サーバを設定する。		[Lweblogic.management.configuration.TargetMBean;@6b7920

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	このデプロイメントの対象クラスタを設定する。		[Lweblogic.management.configuration.TargetMBean;@6b7920

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

83 実行時 SNMP トラップ

次の属性によるソートを行うことができます。

- [名前]
- [ホスト]
- [ポート]
- [コミュニティ]

84 実行時 SNMP

次の属性によるソートを行うことができます。

- [有効化]
- [SNMP ポート]
- [MIB データの更新間隔]
- [サーバ状態チェック間隔係数]
- [コミュニティ プレフィックス]
- [デバッグ レベル]

85 SNMP 属性変更

[コンフィグレーション]

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、属性変更フィルタの名前を設定する。	文字列	MyAttribute Change
[MBean 属性タイプ]	この属性は、変更をモニタする対象の属性を格納したコンフィグレーション MBean のタイプを設定する。	MBean タイプ	NULL
[MBean 属性名]	この属性は、変更をモニタする対象の属性を格納したコンフィグレーション MBean の名前を設定する。	MBean 名	NULL
[属性名]	この属性は、変更をモニタする対象の属性の名前を設定する。		

86 実行時 SNMP 属性変更

次の属性によるソートを行うことができます。

- [名前]
- [MBean 属性タイプ]
- [MBean 属性名]
- [属性名]

87 SNMP カウンタ モニタ

[Configuration]

[General]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、モニタの名前を設定する。	文字列	MyCounter Monitor
[オフセット]	この属性には、しきい値を超過したときにしきい値に加算される整数値を設定する。この値が 0 の場合、エージェントが属性をポーリングしたときに属性値が しきい値 フィールドの値以上であれば、そのたびにトラップが生成される。値が 1 以上の場合、しきい値を超えるたびにこの値がしきい値に加算される。	整数	0

属性	説明	値の範囲	デフォルト値
[しきい値]	この属性には、選択した属性について、設定値に到達または設定値を超過したときにエンタープライズ固有のトラップを生成するしきい値を設定する。	整数	0
[係数]	その属性には、しきい値を超えるたびにしきい値から差し引かれる値を設定する。	整数	
[モニタする MBean タイプ]	この属性は、モニタ対象の属性を格納する MBean のタイプを設定する。	文字列	NULL
[モニタする MBean 名]	この属性は、モニタ対象の属性を格納する MBean の名前を設定する。	文字列	NULL
[モニタする属性名]	この属性は、モニタ対象の属性の名前を設定する。	文字列	NULL
[ポーリング 間隔]	この属性は、エージェントが属性値をチェックする頻度（秒）を設定する。	整数（秒数）	0

[サーバ]

属性	説明	値の範囲	デフォルト値
[有効なサーバ]	この属性では、指定した属性をこのモニタによってチェックする対象のサーバをユーザがリストから選択できる。	リスト	NULL

88 実行時 SNMP カウンタ モニタ

次の属性によるソートを行うことができます。

- [名前]
- [オフセット]
- [しきい値]
- [係数]
- [モニタする MBean タイプ]
- [モニタする MBean 名]
- [モニタする属性名]
- [ポーリング 間隔]
- [有効なサーバ]

89 SNMP ゲージ モニタ

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、モニタの名前を設定する。	文字列	MyGauge Monitor
[モニタする MBean タイプ]	この属性は、モニタ対象の属性を格納する MBean のタイプを設定する。	文字列	NULL
[モニタする MBean 名]	この属性は、モニタ対象の属性を格納する MBean の名前を設定する。	文字列	NULL
[モニタする属性名]	この属性は、モニタ対象の属性の名前を設定する。	文字列	NULL
[ポーリング 間隔]	この属性は、エージェントが属性値をチェックする頻度（秒）を設定する。	整数（秒数）	0

属性	説明	値の範囲	デフォルト値
[最大しきい値]	この属性は、トラップが生成されるしきい値を設定する。属性値がここに設定した値に到達するか、または設定した値を超えるとトラップが生成される。	整数（秒数）	0
[最小しきい値]	この属性は、トラップが生成されるしきい値を設定する。属性値がここに設定した値に到達するか、または設定した値を下回るとトラップが生成される。	整数（秒数）	0

[サーバ]

属性	説明	値の範囲	デフォルト値
[有効なサーバ]	この属性では、指定した属性をこのモニタによってチェックする対象のサーバをユーザがリストから選択できる。	リスト	NULL

90　実行時 SNMP ゲージ モニタ

次の属性によるソートを行うことができます。

- [モニタする MBean タイプ]
- [モニタする MBean 名]
- [モニタする属性名]
- [ポーリング 間隔]
- [最大しきい値]
- [最小しきい値]
- [有効なサーバ]

91 SNMP JMX モニタ

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、モニタの名前を設定する。	文字列	NULL
[モニタする MBean タイプ]	この属性は、モニタ対象の属性を格納する MBean のタイプを設定する。	文字列	NULL
[モニタする MBean 名]	この属性は、モニタ対象の属性を格納する MBean の名前を設定する。	文字列	NULL
[モニタする属性名]	この属性は、モニタ対象の属性の名前を設定する。	文字列	NULL
[ポーリング 間隔]	この属性は、エージェントが属性値をチェックする頻度（秒）を設定する。	整数（秒数）	0

属性	説明	値の範囲	デフォルト値
[最大しきい値]	この属性は、トラップが生成されるしきい値を設定する。属性値がここに設定した値に到達するか、または設定した値を超えるとトラップが生成される。	整数（秒数）	0
[最小しきい値]	この属性は、トラップが生成されるしきい値を設定する。属性値がここに設定した値に到達するか、または設定した値を下回るとトラップが生成される。	整数（秒数）	0

[サーバ]

属性	説明	値の範囲	デフォルト値
[有効なサーバ]	この属性では、指定した属性をこのモニタによってチェックする対象のサーバをユーザがリストから選択できる。	リスト	NULL

92 SNMP ログ フィルタ

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、ログ フィルタの名前を設定する。	文字列	MyLog Filter
[重大度]	この属性には、トラップが生成される重大度レベルを設定する。		Error
[サブシステム名]	この属性には、モニタ対象のサブシステム名を設定する。	文字列	NULL
[ユーザ ID]	この属性には、トラップが生成されたときに通知を受けるユーザ ID を設定する。	文字列	NULL
[メッセージ ID]	この属性には、メッセージから検索されるメッセージ ID を設定する。		NULL

属性	説明	値の範囲	デフォルト値
[メッセージ サブ文字列]	この属性は、メッセージ テキストから検索される文字列を設定する。	文字列	NULL

93 実行時 SNMP ログ フィルタ

次の属性によるソートを行うことができます。

- [名前]
- [重大度]
- [サブシステム名]
- [ユーザ ID]
- [メッセージ ID]
- [メッセージ サブ文字列]

94 SNMP プロキシ

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、プロキシの名前を設定する。	文字列	MyProxy
[ポート]	この属性には、ほかの SNMP エージェントと通信するためのポート番号を設定する。	有効なポート	0
[Oid ルート]	この属性には、エージェントに割り当てられている OID ツリーの部位のルート（またはトップ ノード）を指示する絶対 OID を設定する。	有効な OID ルート	NULL
[コミュニティ]	この属性には、SNMP マネージャからのリクエスト内でほかのエージェントが想定するコミュニティ名を設定する。		なし

属性	説明	値の範囲	デフォルト値
[タイムアウト]	この属性には、別の SNMP エージェントに転送されたリクエストへの応答を WebLogic SNMP プロキシ エージェントが待機する時間を設定する。応答がないままこの時間が経過すると、WebLogic SNMP エージェントはリクエスト送信元のマネージャにエラーを通知する。	整数	5000

95 実行時 SNMP プロキシ

次の属性によるソートを行うことができます。

- [名前]
- [ポート]
- [Oid ルート]
- [コミュニティ]
- [タイムアウト]

96 SNMP 文字列モニタ

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この属性には、モニタの名前を設定する。	文字列	MyString Monitor
[比較文字列]	この属性には、トラップを生成するかどうかを決定するためにモニタ対象の値と比較される文字列を設定する。	文字列	NULL
[違う時に通知する]	この属性では、[比較文字列] フィールドに入力した値と属性値が異なる場合にトラップを生成するかどうかをユーザが選択する。	チェックボックス	選択されていない
[一致する時に通知する]	この属性では、[比較文字列] フィールドに入力した値と属性値が一致する場合にトラップを生成するかどうかをユーザが選択する。	チェックボックス	選択されていない

属性	説明	値の範囲	デフォルト値
[モニタする MBean タイプ]	この属性は、モニタ対象の属性を格納する MBean のタイプを設定する。	文字列	NULL
[モニタする MBean 名]	この属性は、モニタ対象の属性を格納する MBean の名前を設定する。	文字列	NULL
[モニタする属性名]	この属性は、モニタ対象の属性の名前を設定する。	文字列	NULL
[ポーリング 間隔]	この属性は、エージェントが属性値をチェックする頻度（秒）を設定する。	整数（秒数）	0

[サーバ]

属性	説明	値の範囲	デフォルト値
[有効なサーバ]	この属性値では、指定した属性をこのモニタによってチェックする対象のサーバをユーザがリストから選択できる。	リスト	NULL

97 実行時 SNMP 文字列モニタ

次の属性によるソートを行うことができます。

- [名前]
- [比較文字列]
- [違う時に通知する]
- [一致する時に通知する]
- [モニタする MBean タイプ]
- [モニタする MBean 名]
- [モニタする属性名]
- [ポーリング 間隔]
- [有効なサーバ]

98 SNMP トラップの送り先

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	この送り先にユーザが付ける名前。	文字列	MySNMP Trap Destination
[ホスト]	ホスト名または IP アドレスを表す文字列。これは、WebLogic SNMP エージェントによって送られる SNMP トラップ通知の送り先である SNMP マネージャのマシンである。	文字列	NULL
[ポート]	ターゲットの SNMP マネージャに SNMP トラップ通知を送信するために使われるポート。		162

属性	説明	値の範囲	デフォルト値
[コミュニティ]	SNMP トラップのコミュニティ名。これは、ターゲットの SNMP マネージャにトラップ通知を送信するためのパスワードとして機能する。		

99 SNMP

[Monitor SNMP Attributes]

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[有効化]	SNMP Service は管理サーバの一部であり、SNMP エージェント機能を提供する。エージェントはドメインを対象に、すべての WebLogic リソースをモニタする。コンフィグレーションの変更を有効にするには、管理サーバを再起動しなければならない。	True = 有効 False = 無効	False

属性	説明	値の範囲	デフォルト値
[SNMP ポート]	SNMP マネージャからのリクエストを WebLogic SNMP エージェントがリスンするポート。	有効なリスン ポート	161
[MIB データの更新間隔]	SNMP エージェントはすべての属性値のキャッシュを管理し、このキャッシュから属性値を取得することによってマネージャのリクエストに応答する。[MIB データの更新間隔] は、SNMP エージェントがキャッシュを完全リフレッシュする間隔（秒数）である。エージェントはリフレッシュを行うとき、WebLogic SNMP MIB に表現されているすべての WebLogic 属性について GET 要求を実行する。	整数（秒数）	120

属性	説明	値の範囲	デフォルト値
[サーバ状態チェック間隔係数]	SNMP エージェントはこの数値を [MIB データの更新間隔] の値に乗じて、ドメイン内の管理対象サーバが最新状態かどうかのチェックを行う頻度を決定する。エージェントはこの値を MIB キャッシュから取得する。[サーバ状態チェック間隔係数] が 1 の場合、WebLogic SNMP エージェントは、管理対象サーバが最新状態かどうかを [MIB データの更新間隔] に指定された間隔でチェックする。	1 以上の整数	MIB キャッシュから取得

属性	説明	値の範囲	デフォルト値
[コミュニティ プレフィックス]	SNMP マネージャと通信するためのテキストパスワードとして機能する、SNMP コミュニティ名を生成するために使われる文字列。SNMP マネージャから送られてくるコミュニティ プレフィックスがこの属性の設定値と一致しない場合、SNMP エージェントはリクエスト元に authenticationFailuretrap を返す。	文字列 この値が `community_prefix@server_name` のような形式の場合、エージェントは指定された管理対象サーバを対象としたデータだけを返す。 この値が `community_prefix@domain_name` のような形式の場合、エージェントはドメイン内の全サーバを対象としたデータを返す。SNMP マネージャが `community_prefix` だけを送信する場合、エージェントは管理サーバを対象としたデータを取得するだけである。	NULL
[デバッグ レベル]	管理者に送られるデバッグ メッセージのレベルを設定する。0 に設定すると、デバッグ メッセージは生成されない。0 より大きい値の場合は、エージェント コードの動作を示すメッセージが生成される。値が大きいほど、メッセージの詳細さが増す。	0 ～ 3 の範囲の整数	0

100 ソケット

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- プロトコル
- リモート アドレス

101 起動クラス

以下に、Administration Console を使用して、起動クラスのコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

起動クラスのコンフィグレーション

1. [起動と停止] ノードをクリックします。右ペインに [起動と停止] テーブルが表示され、ドメインで定義されているすべての起動クラスが示されます。
2. [新しい Startup Class のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しい起動クラスの設定に関連するタブが示されます。
3. [名前]、[クラス名]、[デプロイ順]、および [引数] 属性フィールドに値を入力します。

   [引数] フィールドで複数の引数を指定する場合は、次の例で示すようにカンマで区切ります。

   ```
   first=MyFirstName,last=MyLastName
   ```
4. [失敗したらサーバを起動しない] を有効にするには、このチェックボックスを選択します。この機能を無効にするには、チェックボックスを選択解除します。
5. [作成] をクリックして、[名前] フィールドに指定した名前で起動クラスのインスタンスを作成します。左ペインの [起動と停止] ノードの下に、新しいインスタンスが追加されます。

起動クラスのクローンの作成

1. [起動と停止] ノードをクリックします。右ペインに [起動と停止] テーブルが表示され、ドメインで定義されているすべての起動クラスが示されます。
2. クローンを作成する起動クラスの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、起動クラスのクローンの作成に関連するタブが示されます。
3. [名前]、[クラス名]、[デプロイ順]、および [引数] 属性フィールドに値を入力します。
4. [失敗したらサーバを起動しない] を有効にするには、このチェックボックスを選択します。この機能を無効にするには、チェックボックスを選択解除します。
5. [作成] をクリックして、[名前] フィールドに指定した名前で起動クラスのインスタンスを作成します。左ペインの [起動と停止] ノードの下に、新しいインスタンスが追加されます。

起動クラスの削除

1. [起動と停止] ノードをクリックします。右ペインに [起動と停止] テーブルが表示され、ドメインで定義されているすべての起動クラスが示されます。
2. 削除する起動クラスの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして起動クラスを削除します。[起動と停止] ノードの下の起動クラス アイコンが削除されます。

起動クラスの割り当て

1. 左ペインの [起動と停止] の下で、割り当てる起動クラスのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. 起動クラスに割り当てる 1 つまたは複数のターゲットを [選択可] カラムで選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	起動クラス名を返す。	完全修飾クラス名	NULL
[クラス名]			
[引数]	起動クラスの引数を設定する。 複数の引数を指定する場合は、次の例で示すようにカンマで区切る。 `first=MyFirstName,last=MyLastName`		
[失敗したらサーバを起動しない]	障害発生時にサーバが起動を中止するかどうかを指定する。	ブール True = 選択されている False = 選択されていない	選択されていない
[デプロイ順]	クラスが実装される順番。		

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	このデプロイメントの対象サーバを設定する。		[Lweblogic.management.configuration.TargetMBean;@5e3974

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	このデプロイメントの対象クラスタを設定する。		[Lweblogic.management.configuration.TargetMBean;@5e3974

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」を参照してください。

102 実行時ステートレス EJB

以下の統計によって、ステートレス EJB をモニタすることができます。

統計	説明
Idle Beans Count	現在アイドル中の Bean 数
Beans In Use Count	現在使用中の Bean 数
Waiter Total Count	現在待機中のトランザクション数
Timeout Total Count	タイムアウトになったトランザクション数
Transactions Committed Total Count	コミットされたトランザクション合計
Transactions Rolled Back Total Count	ロールバックされたトランザクション数
Transactions Timed Out Total Count	タイムアウトになったトランザクション合計

103 実行時ステートフル EJB

以下の統計によって、ステートフル EJB をモニタすることができます。

統計	説明
Cached Beans Current Count	キャッシュされた Bean 数
Cache Access Count	キャッシュのアクセス回数
Cache Hit Count	
Activation Count	アクティブ化された Bean 数
Passivation Count	パッシブな Bean 数
Lock Entries Current Count	
Lock Manager Access Count	
Waiter Total Count	待機中のトランザクション合計
Timeout Total Count	タイムアウトになったトランザクション合計

104 ターゲット グループ

ターゲット グループの作成

1. [Target Groups] ノードをクリックします。右ペインに [Target Groups] テーブルが表示され、ドメインで定義されているすべてのターゲット グループが示されます。
2. [Create a New Target Group] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しいターゲット グループの設定に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. 右下隅の [作成] ボタンをクリックして、[名前] フィールドに指定した名前でターゲット グループのインスタンスを作成します。左ペインの [Target Groups] ノードの下に、新しいインスタンスが追加されます。

ターゲット グループのクローンの作成

1. [Target Groups] ノードをクリックします。右ペインに [Target Groups] テーブルが表示され、ドメインで定義されているすべてのターゲット グループが示されます。
2. クローンを作成するターゲット グループの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、対象グループのクローンの作成に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. 右下隅の [作成] ボタンをクリックして、[名前] フィールドに指定した名前でターゲット グループのインスタンスを作成します。左ペインの [Target Groups] ノードの下に、新しいインスタンスが追加されます。

ターゲット グループの削除

1. [Target Groups] ノードをクリックします。右ペインに [Target Groups] テーブルが表示され、ドメインで定義されているすべてのターゲット グループが示されます。
2. 削除するターゲット グループの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックしてターゲット グループを削除します。[Target Groups] ノードの下にあるターゲットグループ アイコンが削除されます。

ターゲット グループの割り当て

1. 左ペインの [Target Groups] の下で、割り当てるターゲット グループのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [メンバ] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. [選択済み] カラムで、ターゲット グループに割り当てる 1 つまたは複数のターゲットを選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	ターゲット グループの名前を返す。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

105 実行時トランザクション名

デフォルトでは、このペインを使用すると、ユーザが次の基準でトランザクションを並べ替えて表示させることができます。

- 名前
- トランザクション
- コミット数
- ロールバック数
- タイムアウト ロールバック数
- リソース ロールバック数
- アプリケーション ロールバック数
- システム ロールバック数
- ヒューリスティック数
- 平均コミット時間

表示するカラムや並べ替えの順序を変更するには、[このビューをカスタマイズ] テキスト リンクをクリックします。

106 実行時トランザクション リソース

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前
- トランザクション
- コミット数
- ロールバック数
- ヒューリスティック数
- ヒューリスティック コミット数
- ヒューリスティック ロールバック数
- 混合ヒューリスティック数
- ヒューリスティック障害数

表示するカラムや並べ替えの順序を変更するには、[このビューをカスタマイズ] テキスト リンクをクリックします。

107 Unix マシン

以下に、Administration Console を使用して、UNIX マシンのコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『管理者ガイド』の「WebLogic Server の起動と停止」および「WebLogic Server とクラスタのコンフィグレーション」を参照してください。

Unix マシンのコンフィグレーション

1. [マシン] ノードをクリックします。右ペインに [マシン] テーブルが表示され、ドメインで定義されているすべての Unix マシンが示されます。
2. [新しい Unix Machine のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する Unix マシンの設定に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. [作成] をクリックします。
5. [ノード マネージャ] タブをクリックして、デフォルト値をそのまま使用するか変更します。
6. [適用] をクリックして変更を保存します。

Unix マシンのクローンの作成

1. [マシン] ノードをクリックします。右ペインに [マシン] テーブルが表示され、ドメインで定義されているすべての Unix マシンが示されます。

2. クローンを作成する Unix マシンの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、Unix マシンのクローンの作成に関連するタブが示されます。
3. [名前] 属性フィールドに値を入力します。
4. [クローン] をクリックして、[名前] フィールドに指定した名前で Unix マシンのインスタンスを作成します。左ペインの [マシン] ノードの下に、新しいインスタンスが追加されます。
5. [適用] をクリックして変更を保存します。
6. [ノード マネージャ] タブをクリックして、デフォルト値をそのまま使用するか変更します。
7. [適用] をクリックして変更を保存します。

Unix マシンの削除

1. [マシン] ノードをクリックします。右ペインに [マシン] テーブルが表示され、ドメインで定義されているすべての Unix マシンが示されます。
2. 削除する Unix マシンの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、Unix マシンを削除します。[マシン] ノードの下にある Unix マシン アイコンが削除されます。

Unix マシンの割り当て

1. 左ペインの [マシン] の下で、割り当てる Unix マシンのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [サーバ] タブをクリックします。

3. [選択可] カラムで、Unix マシンに割り当てる 1 つまたは複数のターゲットを選択します。
4. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
5. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	Unix マシンの名前を返す。	文字列	MyUnixMachine

[サーバ]

属性	説明	値の範囲	デフォルト値
[サーバ]	この Unix マシン用のサーバを選択できる。	リスト	NULL

[ノード マネージャ]

属性	説明	値の範囲	デフォルト値
[リスン アドレス]	マシンにリスン アドレスを割り当てる。		LocalHost
[リスン ポート]	マシンにリスン ポートを割り当てる。		7002
[Post-Bind UID を有効化]	ポストバインド UID を有効または無効にできる。	ブール True = 選択されている False = 選択されていない	選択されていない
[Post-Bind UID]	UNIX UID を返す。このマシンで動作するサーバは、すべての特権起動アクションの実行後、この UNIX UID の下で動作する。この値を設定した場合、これは有効な UNIX UID。	文字列	nobody
[Post-Bind GID を有効化]	ポストバインド GID を有効または無効にできる。	ブール True = 選択されている False = 選択されていない	選択されていない
[Post-Bind GID]	UNIX GID を返す。このマシンで動作するサーバは、すべての特権起動アクションの実行後、この UNIX GID の下で動作する。この値を設定した場合、これは有効な UNIX GID。	文字列	nobody

属性	説明	値の範囲	デフォルト値
[認証]	サーバのデジタル証明書用ファイルを指定する。		NULL
[認証パスワード]	デジタル証明書のパスワードを設定する。		NULL
[確実に信頼されたファイル]	サーバの信頼性のある CA ファイルを指定する。		NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

108 Unix レルム

以下の表に、Unix レルムの作成や管理に必要な属性を示します。レルムの詳細については、『管理者ガイド』の「セキュリティの管理」を参照してください。

UNIX セキュリティ レルムを使用するには、キャッシング レルムを有効にし、[基本レルム] フィールドに UNIX セキュリティ レルムのクラス名を入力する必要があります。

Unix レルムのコンフィグレーション

1. 左ペインの [レルム] ノードをクリックします。右ペインに [レルム] テーブルが表示され、ドメインで定義されているすべての Unix レルムが示されます。
2. [新しい Unix Realm のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するレルムの設定に関連するタブが示されます。
3. [名前] および [認証プログラム] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前でレルム インスタンスを作成します。左ペインの [レルム] ノードの下に、新しいインスタンスが追加されます。

Unix レルムのクローンの作成

1. 左ペインの [レルム] ノードをクリックします。右ペインに [レルム] テーブルが表示され、ドメインで定義されているすべての Unix レルムが示されます。

2. クローンを作成するレルムの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、レルムのクローンの作成に関連するタブが示されます。
3. [名前] および [認証プログラム] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前でレルム インスタンスを作成します。左ペインの [レルム] ノードの下に、新しいインスタンスが追加されます。

Unix レルムの削除

1. 左ペインの [レルム] ノードをクリックします。右ペインに [レルム] テーブルが表示され、ドメインで定義されているすべての Unix レルムが示されます。
2. 削除するレルムの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、レルムを削除します。[レルム] ノードの下にあるレルム アイコンが削除されます。

UNIX セキュリティ レルムを使用するには、キャッシング レルムを有効にし、[基本レルム] フィールドに UNIX セキュリティ レルムのクラス名を入力する必要があります。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	AccountingRealm など、Unix セキュリティ レルムの名前を指定する。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL
[レルム クラス名]	Unix セキュリティ レルムが所属する Java クラスの名前を返す。Java クラスは、WebLogic Server の CLASSPATH に含まれている必要がある。	この属性は変更不可	weblogic.security.acl.unixrealm.UnixRealm
[認証プログラム]	Unix セキュリティ レルム内のユーザを認証するために使用するプログラムの名前を指定する。	文字列	ほとんどの場合、プログラム名は wlauth

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

109 ユーザ

ユーザの追加

1. 左ペインの [ユーザ] ノードをクリックします。右ペインにダイアログが表示され、ユーザを処理するためのコントロールが示されます。
2. 新しいユーザの作成用の [名前] テキスト入力フィールドに、新しいユーザの名前を入力します。
3. [パスワード] および [パスワード確認] テキスト入力フィールドに、新しいユーザのパスワードを入力します。
4. [作成] をクリックして、指定したユーザを追加します。そのユーザが [ユーザ] リストに追加されます。
5. [変更は、レルムの実装に保存しなければなりません] テキスト リンクをクリックします。これにより、変更は永続的になります。

ユーザの削除

1. 左ペインの [ユーザ] ノードをクリックします。右ペインにダイアログが表示され、ユーザを処理するためのコントロールが示されます。
2. 削除するユーザの名前をユーザの削除の [名前] フィールドに入力します。複数のユーザを削除するには、各ユーザをスペースで区切ってフィールドに入力します。
3. [削除] をクリックして、指定したユーザを削除します。そのユーザが [ユーザ] リストから削除されます。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	ユーザ、すなわち WebLogic Server リソースにアクセスするエンティティの名前。	文字列の名前では、大文字 / 小文字を区別する。	
[パスワード]	ユーザのパスワード。	パスワードは、8 文字以上の長さでなければならない。大文字 / 小文字を区別する。	

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

110 仮想ホスト

以下に、Administration Console を使用して仮想ホストをコンフィグレーションまたは管理する手順を説明します。仮想ホストの詳細については、『管理者ガイド』の「WebLogic Server Web コンポーネントのコンフィグレーション」を参照してください。

仮想ホストのコンフィグレーション

1. [仮想ホスト] ノードをクリックします。右ペインに [仮想ホスト] テーブルが表示され、ドメインで定義されているすべての仮想ホストが示されます。
2. [新しい Virtual Host のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する仮想ホストの設定に関連するタブが示されます。
3. [名前] および [仮想ホスト名] 属性フィールドに値を入力し、ドロップダウン リストから [デフォルト Web アプリケーション] を選択します。
4. [ログ] タブと [HTTP] タブをそれぞれクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
5. [作成] をクリックして、[名前] フィールドに指定した名前で Web サーバのインスタンスを作成します。左ペインの [仮想ホスト] ノードの下に、新しいインスタンスが追加されます。
6. 仮想ホストを有効にするためにサーバを再起動します。

仮想ホストの詳細については、『管理者ガイド』の「仮想ホスティングのコンフィグレーション」を参照してください。

仮想ホストのクローンの作成

1. [仮想ホスト] ノードをクリックします。右ペインに [仮想ホスト] テーブルが表示され、ドメインで定義されているすべての仮想ホストが示されます。
2. クローンを作成する仮想ホストの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、仮想ホストのクローンの作成に関連するタブが示されます。
3. [名前] および [仮想ホスト名] 属性フィールドに値を入力し、ドロップダウン リストから [デフォルト Web アプリケーション] を選択します。
4. [ログ] タブと [HTTP] タブをそれぞれクリックして、属性フィールドを変更するか、デフォルト値をそのまま使用します。
5. [作成] をクリックして、[名前] フィールドに指定した名前で Web サーバのインスタンスを作成します。左ペインの [仮想ホスト] ノードの下に、新しいインスタンスが追加されます。

仮想ホストの詳細については、『管理者ガイド』の「WebLogic Server Web コンポーネントのコンフィグレーション」を参照してください。

仮想ホストの削除

1. [仮想ホスト] ノードをクリックします。右ペインに [仮想ホスト] テーブルが表示され、ドメインで定義されているすべての仮想ホストが示されます。
2. 削除する仮想ホストの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、仮想ホストを削除します。[仮想ホスト] ノードの下にある Web サーバ アイコンが削除されます。

仮想ホストの詳細については、『管理者ガイド』の「WebLogic Server Web コンポーネントのコンフィグレーション」を参照してください。

仮想ホストの割り当て

1. 左ペインで、割り当てる仮想ホストのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. [選択可] カラムで、仮想ホストに割り当てる 1 つまたは複数のターゲットを選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

仮想ホストの詳細については、『管理者ガイド』の「WebLogic Server Web コンポーネントのコンフィグレーション」を参照してください。

仮想ホストへの Web アプリケーションの割り当て

1. 左ペインの [Web アプリケーション] ノードをクリックします。
2. 割り当てる Web アプリケーションを選択します。
3. 右ペインで [対象] タブをクリックします。
4. [仮想ホスト] タブをクリックします。
5. [選択可] カラムで仮想ホストをクリックし、右矢印ボタンを使用して仮想ホストを [選択済み] カラムに移動します。

仮想ホストの詳細については、『管理者ガイド』の「WebLogic Server Web コンポーネントのコンフィグレーション」を参照してください。

仮想ホストのすべてのインスタンスのモニタ

1. [仮想ホスト] ノードをクリックします。右ペインに [仮想ホスト] テーブルが表示され、ドメインで定義されているすべての仮想ホストが示されます。
2. モニタする仮想ホストの行で [すべてのインスタンスのモニタ] アイコンをクリックします。右ペインにダイアログが表示され、サーバ ドメイン全体にデプロイされている仮想ホストのすべてのインスタンスが示されます。

仮想ホストの詳細については、『管理者ガイド』の「WebLogic Server Web コンポーネントのコンフィグレーション」を参照してください。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	仮想ホスト名を返す。	文字列	NULL
[仮想ホスト名]	この仮想ホストがどのホストに代わって要求を処理するのかを返す。	文字列	NULL
[デフォルト Web アプリケーション]	デフォルトの Web アプリケーションの名前を設定する。	文字列	NULL

[ログ]

属性	説明	値の範囲	デフォルト値
[ログを有効化]	HTTP アクセス ログを作成するかどうかを設定する。	ブール True = アクセス ログが作成される False = アクセス ログが作成されない	true
[ログファイル名]	HTTP アクセス ログの作成が有効になっている場合、この属性はログ ファイルの名前を設定する。	文字列	NULL
[ログ ファイル フォーマット]	HTTP アクセス ログの作成が有効になっている場合、この属性はログ ファイルのタイプを設定する。	[Common] = 標準 HTTP アクセス ログ形式を使用 [Extended] = 拡張 HTTP アクセス ログ形式を使用	Common
[ログ ファイル バッファ サイズ]	[ローテーション タイプ] が [サイズ] に設定されている場合、この属性はログ ファイルの最大サイズを設定する（単位は KB）。ログ ファイルがこのサイズに達すると、新しいログ ファイルが作成される。	整数	8 KB

属性	説明	値の範囲	デフォルト値
[ローテーション タイプ]	HTTP ログ ファイルを日付またはサイズのどちらに基づいてローテーションするかを指定する。	[サイズ] = ログはそのサイズに基づいてローテーションされる。[ログファイル バッファ サイズ] 属性のデフォルト値をオーバーライドすることもできる。 [時間] = ログは日付に基づいてローテーションされる。[ローテーション間隔] 属性のデフォルト値をオーバーライドや [ローテーション開始時間] 属性を設定することもできる。	サイズ
[ローテーション間隔]	[ローテーション タイプ] が [時間] に設定されている場合、この属性はアクセス ログのローテーション期間を分数で指定する。	整数	2147483647 分
[ログ ファイル更新間隔]	[ローテーション タイプ] が [時間] に設定されている場合、この属性は新しいデータがログ ファイルに書き込まれる間隔を秒数で設定する。	整数	60 秒
[ローテーション開始時間]	[ローテーション タイプ] が [時間] に設定されている場合、この属性はアクセス ログの最初のローテーションの日時を指定する。	日付の形式は、java.text.SimpleDateFormat（MM-dd-yyyy-k:mm:ss）に従う	

属性	説明	値の範囲	デフォルト値
[最大ログファイルサイズ]	ログ ファイルの最大サイズを設定できる。	整数（KB）	

[HTTP]

属性	説明	値の範囲	デフォルト値
[デフォルトサーバー名]	WebLogic Server では、リクエストをリダイレクトするときに、[デフォルト サーバ名] で指定した文字列を使用して、HTTP 応答ヘッダとして返されるホスト名を設定する。 ファイアウォールまたはロード バランサを使用し、ブラウザからリクエストをリダイレクトして、元のリクエストで送られたものと同じホスト名を参照する場合に便利。	文字列	NULL
[Keep Alive を有効化]	HTTP キープアライブを有効にするかどうかを設定する。	ブール True = 有効 False = 無効	選択
[Send Server Header]	false に設定すると、サーバ名は HTTP 応答と共に送られない。ヘッダの領域が限られる無線アプリケーションの場合に便利。	ブール True = 有効 False = 無効	True
[Keep Alive 時間]	非アクティブな HTTP 接続を閉じるまで WebLogic Server が待機する秒数。	整数	30

属性	説明	値の範囲	デフォルト値
[Https Keep Alive 時間]	非アクティブな HTTPS 接続を閉じるまで WebLogic Server が待機する秒数。	整数	60
[WAP 有効化]	この属性を選択すると、セッション ID は JVM 情報に含まれなくなる。これは、URL のサイズが 128 文字に制限される WAP デバイスで URL 書き換えを使用する場合に必要になる。[WAP 有効化] を選択すると、クラスタでのレプリケートされたセッションの使用に影響することがある。	有効 無効	無効
[POST タイムアウト 秒]	WebLogic Server が HTTP POST データ中の連続データを受信する間隔のタイムアウトを秒単位で設定する。POST データを使用してサーバを過負荷状態にしようとするサービス拒否攻撃を防ぐために使用する。	整数	0
[最大 Post 時間]	WebLogic Server が HTTP POST データ中の連続データを待機する時間を秒単位で設定する。	整数	0
[最大 Post サイズ]	HTTP POST データ中の連続データの最大サイズを設定する。	整数	0

HTTP アクセス ログ設定の詳細については、『WebLogic Server 管理者ガイド』の「HTTP アクセス ログの設定」を参照してください。

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	この仮想ホストをデプロイするサーバを選択する。	リスト	NULL

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	この Web サーバと共に使用する 1 つまたは複数の対象を選択するためのリスト。	リスト	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	値は最大 256 文字の英数字	NULL

仮想ホストの詳細については、『管理者ガイド』の「仮想ホスティングのコンフィグレーション」を参照してください。

111 Web アプリケーション

以下に、Administration Console を使用して、Web アプリケーションのインストールに必要な属性を設定する手順を説明します。Web アプリケーションの詳細については、『Web アプリケーションの組み立てとコンフィグレーション』を参照してください。

新しい Web アプリケーションのインストール

1. 左ペインの [Web アプリケーション] ノードをクリックします。右ペインに [Web アプリケーション] テーブルが表示され、デプロイされたすべての Web アプリケーションが示されます。
2. テキスト入力フィールドに . jar ファイルのパスを入力するか、[参照] ボタンをクリックしてファイル システムを参照し、インストールする .jar ファイルを選択します。
3. [Upload] をクリックして、.jar ファイルをインストールします。左ペインの [Web アプリケーション] ノードの下に、新しい Web アプリケーションが追加されます。[新しい Web Application のインストール] を選択すると、アプリケーション ディレクトリに .jar/.ear ファイルがコピーされます。Web アプリケーション名は、デフォルトでは war ファイルと同じファイル名になります。

新しい Web アプリケーションのコンフィグレーション

1. 左ペインで [Web アプリケーション] ノードをクリックします。右ペインに Web アプリケーション テーブルが表示され、デプロイされたすべての Web アプリケーションが示されます。
2. テキスト入力フィールドに .jar ファイルのパスを入力するか、[参照] ボタンをクリックしてファイル システムを参照し、インストールする .jar ファイルを選択します。

[新しい Web Application のコンフィグレーション] オプションを選択すると、Web アプリケーション名、URI、パスなどのオプションも指定できます。[新しい Web Application のコンフィグレーション] オプションを選択すると次のような利点があります。

1. jar ファイルがアプリケーション ディレクトリにコピーされない。
2. この Web アプリケーションのインストール先を指定できる。

[コンフィグレーション]

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	Web アプリケーションの名前を設定する。Web アプリケーションのコンポーネントにアクセスするために URL で使用されるコンテキスト パス。	文字列	NULL
[パス]	ファイル システム内の Web アプリケーションのフル パス。展開ディレクトリ形式の場合は、そのディレクトリへのパス。.war 形式でアーカイブされている場合は、.war ファイルのパス。 絶対パスか、またはこの Web アプリケーションを格納する J2EE アプリケーションのパス属性に基づく相対パスを指定する。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[デプロイ順]	デプロイメントに使用する順序。順序付けは同じクラスのほかのデプロイ可能なユニット（EJB、Web アプリケーションなど）と関連する。デプロイ順の値が最小のデプロイメントが最初にデプロイされる。 デプロイ順の値が等しいデプロイメントの順序付けは保証されていない。クラスタ間の順序付けは保証されていない。	値の範囲 : 最小 : 0 最大 : MAX_VALUE で指定した整数値	1000
[デプロイ]	有効の場合、Web アプリケーションはデプロイされる。	ブール デプロイ = 選択されている デプロイされない = 未選択	選択されていない

[ファイル]

属性	説明	値の範囲	デフォルト値
[インデックス ディレクトリ]	適切なインデックス ファイルが見つからない場合、HTML ディレクトリ リストを自動的に生成するかどうかを設定する。	ブール True = 適切なインデックス ファイルが見つからない場合、ディレクトリ リストが返される False = ディレクトリ リストは返されない	選択
[再ロード間隔 (秒)]	WebLogic Server がサーブレットの変更をチェックするまでの待ち時間を設定する。	-1 = WebLogic Server はサーブレットの変更をチェックしない 0 = WebLogic Server は常にサーブレットが変更されたかどうかをチェックして、変更されたサーブレットを再ロードする 1 ～ 最大整数値 = WebLogic Server がサーブレットの変更をチェックするまで待つ秒数	1
[大文字 / 小文字を区別する]	HTTP リクエストを解決するときに、ファイル拡張子の大文字 / 小文字を区別するかどうかを設定する。	ブール True = 大文字 / 小文字を区別する False = 大文字 / 小文字を区別しない	選択されていない

[その他]

属性	説明	値の範囲	デフォルト値
[シングル スレッド サーブレット プール サイズ]	SingleThreadMode インスタンス プールに使用するプールのサイズを定義する。	整数	5
[認証レルム名]	ブラウザに表示される [Basic Authentication HTTP] ダイアログ ボックスのレルムを認定する。	文字列	weblogic

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	この Web サーバと共に使用する 1 つまたは複数の対象を選択するためのリスト。	リスト	NULL

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	この Web サーバと共に使用する 1 つまたは複数の対象を選択するためのリスト。	リスト	NULL

[仮想ホスト]

属性	説明	値の範囲	デフォルト値
[WebServers-Virtual Host]	この Web サーバと共に使用する 1 つまたは複数の対象を選択するためのリスト。	リスト	NULL

[モニタ]

属性	説明	値の範囲	デフォルト値
[セッションモニタを有効化]	有効の場合、セッションの実行時 Mbean が作成される。無効の場合は、作成されない。	ブール 有効 = 選択 無効 = 未選択	選択されていない

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザが任意に情報を指定するためのスペースを提供する。	文字列	NULL

Web アプリケーションの詳細については、『Web アプリケーションの組み立てとコンフィグレーション』を参照してください。

デプロイメント記述子の作成については、「Web アプリケーションのデプロイメント記述子の記述」を参照してください。

112 実行時 Web アプリケーション コンポーネント

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- サーバ
- マシン
- ソース情報
- サーブレット
- セッション
- 最大セッション数
- 総セッション数

113 Web アプリケーション デプロイメント記述子エディタ ヘルプ

以下の各節では、Web アプリケーション デプロイメント記述子エディタの使用方法についての情報を記載しています。

- Web アプリケーション デプロイメント記述子エディタの概要
- Web アプリケーション デプロイメント記述子エディタによる Web アプリケーションのコンフィグレーション
- 新しい Web アプリケーションの作成
- Web アプリケーション デプロイメント記述子エディタの使用方法
- 既存のデプロイメント記述子の編集
- Web アプリケーション コンポーネントのコンフィグレーション手順（コンポーネント名に続けて、デプロイメント記述子内でコンポーネントを表現するために使われるトップレベルの XML 要素を記述しています）
  - Web アプリケーションの基本属性のコンフィグレーション (`<icon>`、`<display-name>`、`<description>`、`<distributable>`)
  - コンテキスト パラメータのコンフィグレーション (`<context-param>`))
  - フィルタのコンフィグレーション (`<filter>`、`<filter-mapping>`)
  - リスナのコンフィグレーション (`<listener>`)
  - サーブレットのコンフィグレーション (`<servlet>`、`<servlet-mapping>`)
  - ウェルカム ページのコンフィグレーション (`<welcome-file-list>`)
  - セッション タイムアウトのコンフィグレーション (`<session-config>`)
  - MIME マッピングのコンフィグレーション (`<mime-mapping>`)
  - エラー ページのコンフィグレーション (`<error-page>`)
  - JSP タグ ライブラリのコンフィグレーション (`<taglib>`)

- リソース環境参照のコンフィグレーション (<resource-env-ref>)
- リソース参照のコンフィグレーション (<resource-ref>)
- セキュリティ制約のコンフィグレーション (<security-constraint>)
- ログインのコンフィグレーション (<login-config>)
- セキュリティ ロールのコンフィグレーション (<security-role>)
- 環境エントリのコンフィグレーション (<env-entry>)
- EJB 参照のコンフィグレーション (<ejb-ref>)

■ 新しい Webapp Ext のコンフィグレーション (weblogic.xml)

- セッション記述子のコンフィグレーション (<session-descriptor>)
- JSP 記述子のコンフィグレーション (<jsp-descriptor>)
- セキュリティ ロール割り当てのコンフィグレーション (<security-role-assignment>)
- リファレンス記述子のコンフィグレーション (<reference-descriptor>)
- 文字セット パラメータのコンフィグレーション (<charset-params>)
- コンテナ記述子のコンフィグレーション (<container-descriptor>)

■ WebLogic Web Service のコンフィグレーション

Web アプリケーション デプロイメント記述子エディタの概要

Web アプリケーション デプロイメント記述子エディタでは、Web アプリケーションを定義するデプロイメント記述子を編集します。Web アプリケーションは J2EE のデプロイメント ユニットで、JSP、サーブレット、HTML ページなどの Web リソースのコレクションを定義します。Web アプリケーションでは、EJB などの外部リソースへの参照を定義することもできます。

Web アプリケーションごとに、2 つのデプロイメント記述子 web.xml および weblogic.xml が存在します。これらは両方とも、Web アプリケーション デプロイメント記述子エディタを使用して編集できます。デプロイメント記述子 web.xml は、Sun Microsystems の Servlet 仕様で定義されています。weblogic.xml は、WebLogic Server に固有のデプロイメント記述子です。Web アプリケーションの詳細については、http://edocs.beasys.co.jp/e-docs/wls61/webapp/index.html で公開されている WebLogic Server ドキュメントの『Web アプリケーションのアセンブリとコンフィグレーション』を参照してください。

Web アプリケーション デプロイメント記述子エディタでは、既存のデプロイメント記述子の編集だけが可能で、新しいデプロイメント記述子を作成することはできません。

新しい Web アプリケーションの作成

新しい Web アプリケーションを作成するには

1. Web アプリケーションのコンポーネント （サーブレット、JSP、および HTML ページ）を作成します。
2. 作成したコンポーネントを、ファイル システム内で Web アプリケーション用に定義したディレクトリ構造に配置します。詳細については、「ディレクトリ構造」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/basics.html#dir）を参照してください。
3. スケルトンのデプロイメント記述子を作成します。スケルトンの作成は、次の 2 つのうちいずれかの方法で行います。
   - 「Web アプリケーションのパッケージ化」（http://edocs.beasys.co.jp/e-docs/wls61/programming/packaging.html#pack005）で説明されている手順に従います。
   - web.xml という名前のテキスト ファイルを作成し、Web アプリケーションの WEB-INF ディレクトリに保存します。このファイルには、以下のテキストを記述する必要があります。

     ```
     <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//
     DTD Web Application 2.3//EN"
     "http://java.sun.com/dtd/web-app_2_3.dtd">
     ```

```
<web-app>
</web-app>
```

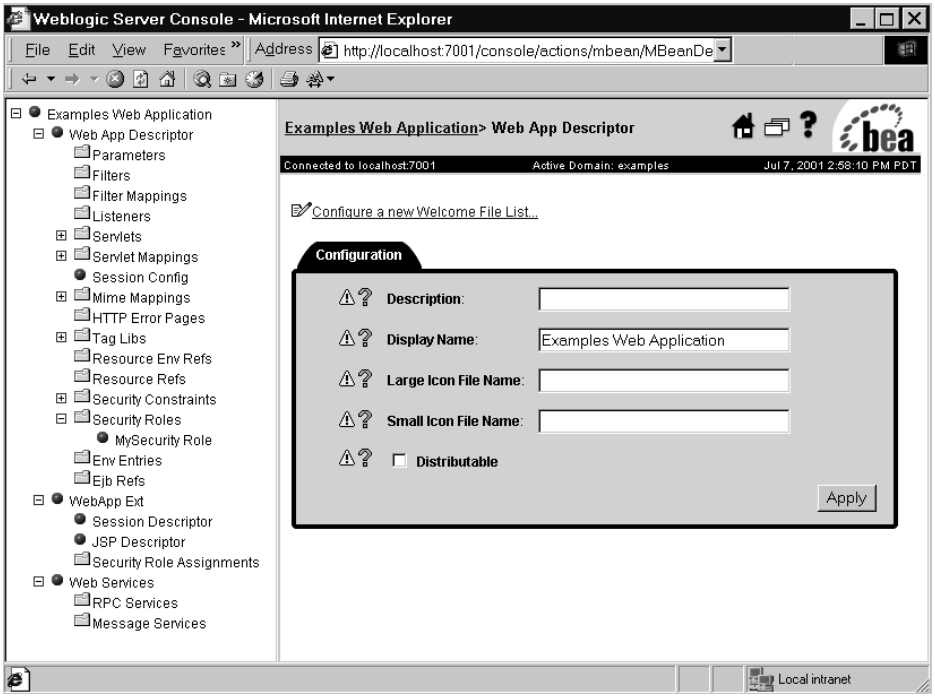
4. 次の節で説明するように、デプロイメント記述子エディタを使用して、Web アプリケーションの機能をコンフィグレーションします。

Web アプリケーション デプロイメント記述子エディタの使用方法

1. まだ動作していない場合、WebLogic Administration Server を起動します。
2. ブラウザで WebLogic Server Administration Console を起動します。
3. 現在のドメインで Web アプリケーションが既に定義済みの場合、左パネルで [Web アプリケーション] ノードを展開してステップ 5. に進みます。
4. 現在のドメインで Web アプリケーションがまだコンフィグレーションされていない場合、新しい Web アプリケーションをコンフィグレーションします。
   a. 右パネルで [新しい Web Application のコンフィグレーション] をクリックします。
   b. [名前] フィールドに、Web アプリケーションの名前を入力します。
   c. [URI] フィールドに、Web アプリケーションが格納されているディレクトリの名前を入力します。
   d. [Path] フィールドに、Web アプリケーションが格納されているディレクトリのパスを入力します。
   e. [作成] をクリックします。新しい Web アプリケーションが、左パネルで [Web アプリケーション] の下に表示されます。
5. Web アプリケーションを右クリックし、[Web アプリケーション 記述子の編集] を選択します。デプロイメント記述子を編集するための新しいブラウザウィンドウが開きます。
6. Web アプリケーションの機能をコンフィグレーションします。Web アプリケーションの機能をコンフィグレーションする手順については、「Web アプリケーション デプロイメント記述子エディタによる Web アプリケーションのコンフィグレーション」を参照してください。

7. デプロイメント記述子の編集が終了したら、左側のナビゲーション ツリーで最上位のノードを選択します。
8. 右パネルで [Validate] をクリックします。検証では、デプロイメント記述子がその DTD（文書型定義）に従っていることのチェックだけが行われます。コンフィグレーションしたコンポーネントが存在していることの検証や、その他の種類の検証は行いません。
9. デプロイメント記述子を検証したら、右パネルで [Persist] をクリックします。デプロイメント記述子に対して行われた変更は、[Persist] ボタンをクリックした時点でデプロイメント記述子ファイルに書き込まれます。

図 113-1 [Web アプリケーション 記述子の編集 ...]



既存のデプロイメント記述子の編集

1. 「Web アプリケーション デプロイメント記述子エディタの使用方法」の手順に従います。
2. 左パネルのナビゲーション ツリーから、編集対象のデプロイメント記述子の要素を選択します。一部の要素については、左側のナビゲーション ツリーで親のノードを右クリックし、要素を作成するための項目を選択する必要があります。詳細については、「Web アプリケーション デプロイメント記述子エディタによる Web アプリケーションのコンフィグレーション」を参照してください。
3. 右パネルで、必要に応じてフィールドの値を入力または編集します。
4. [作成] または [適用] ボタンをクリックします。
5. デプロイメント記述子の編集が終了したら、左側のナビゲーション ツリーで最上位のノードを選択します。
6. 右パネルで [Validate] をクリックします。検証では、デプロイメント記述子がその DTD（文書型定義）に従っていることのチェックだけが行われます。コンフィグレーションしたコンポーネントが存在していることの検証や、その他の種類の検証は行いません。
7. 右パネルで [Persist] をクリックします。デプロイメント記述子に対して行われた変更は、[Persist] ボタンをクリックした時点でデプロイメント記述子ファイルに書き込まれます。

Web アプリケーション デプロイメント記述子エディタによる Web アプリケーションのコンフィグレーション

この節では、Web アプリケーションに必要なデプロイメント情報を作成する方法について説明します。この作業を行う前に、「新しい Web アプリケーションの作成」および「Web アプリケーション デプロイメント記述子エディタの使用方

法」での説明に従って、基本のデプロイメント記述子を作成し、Web アプリケーション デプロイメント記述子エディタ で Web アプリケーションを開いておく必要があります。

以下の各項では、Web アプリケーションのコンポーネントをコンフィグレーションする手順を説明します。手順の説明に続けて、コンフィグレーション可能な属性の表を示します。この表は、手順の説明からリンクによって参照することもできます。

Web アプリケーションの基本属性のコンフィグレーション

1. 左パネルで [Web App Descriptor] ノードをクリックします。
2. [Web App Descriptor] パネルで、必要に応じて以下の属性をコンフィグレーションします。
   - Description
   - Display Name
   - Large Icon File Name
   - Small Icon File Name
   - Distributable

コンテキスト パラメータのコンフィグレーション

コンテキスト パラメータは、サーブレット コンテナに名前 / 値のペアを渡すために使われます。名前 / 値のペアで構成される任意の数のコンテキスト パラメータを入力することができます。

1. 左パネルで [Parameters] ノードを右クリックし、[新しい Parameter のコンフィグレーション] を選択します。
2. [Description] フィールドに、パラメータの説明を入力します。（省略可能）
3. [Param Name] フィールドに、パラメータの名前を入力します。
4. [Param Value] フィールドに、パラメータの値を入力します。

5. [作成] をクリックします。

これらのパラメータには、コード内で `javax.servlet.ServletContext.getInitParameter()` および `javax.servlet.ServletContext.getInitParameterNames()` メソッドを使用してアクセスできます。

フィルタのコンフィグレーション

フィルタは Web アプリケーションのリソースに対するリクエストを傍受し、補助的な機能を実行します。フィルタをコンフィグレーションするには、フィルタを作成し、サーブレットまたは URL パターンにマッピングします。コンフィグレーション可能なフィルタまたはフィルタ マッピングの数に制限はありません。詳細については、「フィルタ」（`http://edocs.beasys.co.jp/e-docs/wls61/webapp/filters.html`）を参照してください。

フィルタは Sun Microsystems の Servlet 2.3 仕様の最終勧告ドラフトの一部です。アプリケーションでフィルタを使用する場合、仕様は最終的に承認されておらず、将来変更される可能性がある点に注意してください。

1. 左パネルで [Filters] ノードを右クリックし、[新しい Filter のコンフィグレーション] を選択します。
2. 以下のフィールドに値を入力します。
   - Filter Name（必須）このフィルタの名前
   - Description（省略可能）
   - Display Name（省略可能）
   - Small Icon File Name（省略可能）
   - Large Icon File Name（省略可能）
   - Filter Class（必須）フィルタを実行する Java クラスの完全なクラス名。例 : `myApp.filters.myFilter`
3. [作成] をクリックします。

4. 左パネルで [Filter Mappings] ノードを右クリックし、[新しい FilterMapping のコンフィグレーション] を選択します。フィルタのマッピングは、この Web アプリケーションで既にコンフィグレーション済みの URL パターンまたはサーブレットに対して行うことができます。それぞれのフィルタに対して複数のマッピングを作成できます。
5. [Filter] ドロップダウン リストから、マッピングを作成するフィルタを選択します。
6. このフィルタを URL パターンにマッピングする場合、[Url Pattern] フィールドに URL パターンを入力します。
7. このフィルタをサーブレットにマッピングする場合、[Servlet] ドロップダウン リストからサーブレットを選択します。サーブレットをフィルタにマッピングするには、サーブレットがコンフィグレーション済みである必要があります。
8. [作成] をクリックします。
9. フィルタ初期化パラメータを作成します。フィルタ初期化パラメータは、フィルタ クラスで `FilterConfig.getInitParameter()` または `FilterConfig.getInitParameters()` メソッドを使用して読み取ることができます。
   a. 左パネルで [Filters] ノードを展開します。
   b. 初期化パラメータを作成するフィルタのノードを展開します。
   c. [Parameters] ノードを右クリックし、[新しい Parameter のコンフィグレーション] を選択します。
   d. [Description] フィールドに、パラメータの説明を入力します。（省略可能）
   e. [Param Name] フィールドに、パラメータの名前を入力します。
   f. [Param Value] フィールドに、パラメータの値を入力します。
   g. [作成] をクリックします。
10. 作成するフィルタ マッピングごとに、これまでの手順を繰り返します。

リスナのコンフィグレーション

リスナは HTTP セッションまたはサーブレットのコンテキスト イベントに応答する Java クラスです。リスナは任意の数だけ作成できます。詳細については、「アプリケーション イベントとリスナ」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/app_events.html）を参照してください。

アプリケーション イベントおよびリスナは、Sun Microsystems の Servlet 2.3 仕様の最終勧告ドラフトの一部です。アプリケーションでアプリケーション イベントおよびリスナを使用する場合、仕様は最終的に承認されておらず、将来変更される可能性がある点に注意してください。

1. 左パネルで [Listeners] ノードを右クリックし、[新しい Listener のコンフィグレーション] を選択します。
2. リスナ クラスの完全名を [Listener Class Name] フィールドに入力します。
3. [作成] をクリックします。

サーブレットのコンフィグレーション

サーブレットをコンフィグレーションするには、まずサーブレットを定義し、次にサーブレットを URL パターンにマッピングします。サーブレットは任意の数だけ作成できます。詳細については、「サーブレットのコンフィグレーション」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/components.html#configuring-servlets）を参照してください。

1. 左パネルで [Servlet] ノードを右クリックし、[新しい servlet のコンフィグレーション] を選択します。
2. 以下のフィールドに値を入力します。
   - Description（省略可能）
   - Display Name（省略可能）
   - Small Icon File Name（省略可能）
   - Large Icon File Name（省略可能）
   - Servlet Name （必須）

- Servlet Class このサーブレットを実行する Java クラス
  または
  Jsp File このサーブレットによって実行される JSP ファイル
- Load On Startup （省略可能）

3. [作成] をクリックします。
4. [Servlet Mappings] を右クリックし、[新しい ServletMapping のコンフィグレーション] を選択します。
5. [Servlet] ドロップダウン リストから、マッピングするサーブレットを選択します。
6. このサーブレットの URL パターンを入力します。URL パターンの詳細については、「サーブレットのマッピング」（`http://edocs.beasys.co.jp/e-docs/wls61/webapp/components.html#servlet-mapping`）を参照してください。
7. [作成] をクリックします。
8. サーブレットの初期化パラメータを作成します。（省略可能）
   a. 左パネルで [Servlet] ノードを展開します。
   b. 初期化パラメータを作成するサーブレットのノードを展開します。
   c. 左パネルで [Parameters] ノードを右クリックし、[新しい Parameter のコンフィグレーション] を選択します。
   d. [Description] フィールドに、パラメータの説明を入力します。（省略可能）
   e. [Param Name] フィールドに、パラメータの名前を入力します。
   f. [Param Value] フィールドに、パラメータの値を入力します。
   g. [作成] をクリックします。

   初期化パラメータを取得するには、親の `javax.servlet.GenericServlet` クラスから `getInitParameter(String name)` メソッドを呼び出します。パラメータの名前が渡されると、このメソッドは `String` 型のパラメータ値を返します。
9. サーブレットの Security Role Refs を作成します。
   a. 左パネルで [Servlet] ノードを展開します。

b. セキュリティ ロール参照を作成するサーブレットのノードを展開します。

c. [Security Role Refs] ノードを右クリックし、[新しい SecurityRoleRef のコンフィグレーション] を選択します。

d. [Description] フィールドに、セキュリティ ロール参照の説明を入力します。（省略可能）

e. [Role Link] ドロップダウン リストから、セキュリティ ロールを選択します。事前に [Security Roles] パネルで、セキュリティ ロールを定義しておく必要があります。

f. [Role Name] フィールドに、サーブレットのコード内で使われるロール名を入力します。

ウェルカム ページのコンフィグレーション

ディレクトリ名による HTTP リクエストを受信すると、WebLogic Server はここで設定するリストの先頭に指定されたファイルを返します。先頭のファイルが見つからない場合、リスト内の次のファイルが検索されます。詳細については、「ウェルカム ページのコンフィグレーション」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/components.html#welcome_pages）を参照してください。

1. 左パネルで [Web App Descriptor] を右クリックし、[新しい WelcomeFileList のコンフィグレーション] を選択します。
2. [Welcome Files] リストに、ファイル名を 1 行ずつに区切って入力します。
3. [作成] をクリックします。

セッション タイムアウトのコンフィグレーション

セッション タイムアウトでは、この Web アプリケーションで HTTP セッションが失効する時間を設定します。

1. 左パネルで [Session Config] ノードを選択します。
2. [Session Timeout] の値を編集します。

3. [適用] をクリックします。

MIME マッピングのコンフィグレーション

MIME マッピングでは、ファイル拡張子と MIME タイプの間のマッピングを定義します。MIME マッピングは任意の数だけ作成できます。

1. 左パネルで [MIME Mappings] ノードを右クリックし、[新しい MimeMapping のコンフィグレーション] を選択します。
2. [Mime Type] フィールドに、有効な MIME タイプを入力します。
3. [Extension] フィールドに、MIME タイプにマッピングするファイル拡張子を入力します。
4. [作成] をクリックします。

エラー ページのコンフィグレーション

エラー ページは、HTTP エラー コードまたは Java 例外への応答として表示されるように設定された JSP または HTTP ページです。各種の HTTP エラー コードまたは Java 例外に応答する、さまざまなエラー ページを設定することができます。詳細については、「HTTP エラー応答のカスタマイズ」（`http://edocs.beasys.co.jp/e-docs/wls61/webapp/components.html#error-page`）を参照してください。

1. 左パネルで [Error Pages] を右クリックし、[新しい ErrorPage のコンフィグレーション] を選択します。
2. [Error Code] フィールドに、HTTP エラー コードを入力します。
   または
   [Exception Type] フィールドに、Java の例外クラスを入力します。
3. エラーに応答して表示されるリソースの名前を入力します。入力する名前の先頭には / を付けます。たとえば、`/myErrorPg.html` のように入力します。
4. [作成] をクリックします。

JSP タグ ライブラリのコンフィグレーション

JSP タグ ライブラリには、ユーザが記述する JSP タグを定義する Java クラスおよび記述子が含まれます。作成できるタグ ライブラリの数に制限はありません。1 つまたは複数の JSP タグ ライブラリをコンフィグレーションできます。詳細については、『WebLogic JSP タグ エクステンション プログラマーズ ガイド』（http://edocs.beasys.co.jp/e-docs/wls61/taglib/index.html）を参照してください。

1. 左パネルで [Tag Libs] ノードを右クリックし、[新しい TagLib のコンフィグレーション] を選択します。
2. [URI] フィールドに、JSP の taglib ディレクティブでこのタグ ライブラリを参照するための名前を（Web アプリケーションの WEB-INF ディレクトリからの相対位置で）入力します。
3. [Location] フィールドに、タグ ライブラリまたはタグ ライブラリの jar ファイルの位置を（Web アプリケーションのルート ディレクトリからの相対位置で）入力します。
4. [作成] をクリックします。

リソース環境参照のコンフィグレーション

リソース環境参照は、Sun Microsystems の Servlet 2.3 仕様の最終勧告ドラフトの一部であり、現時点では WebLogic Server に実装されていません。

1. 左パネルで [Resource Env Refs] ノードを右クリックし、[新しい ResourceEnvRef のコンフィグレーション] を選択します。
2. [Description] フィールドに、このリソース環境参照の説明を入力します。（省略可能）
3. [Ref Name] フィールドに、このリソース環境参照の名前を入力します。
4. [Ref Type] フィールドに、このリソース環境参照の Java 型を入力します。
5. [作成] をクリックします。

リソース参照のコンフィグレーション

リソース参照は、外部リソースのルックアップ名を定義します。デプロイメント時の実際の位置にマッピングされるこの「仮想」名によって、サーブレットのコード内でリソースをルックアップすることができます。リソース参照は任意の数だけ作成できます。

1. 左パネルで [Resource Refs] ノードを右クリックし、[新しい ResourceRef のコンフィグレーション] を選択します。
2. [Description] フィールドに、このリソース参照の説明を入力します。（省略可能）
3. [Ref Name] フィールドに、このリソース参照の名前を入力します。
4. [Ref Type] フィールドに、このリソース参照の Java 型を入力します。Java 型は完全なパッケージ名で入力します。
5. [Auth] フィールドに、認可タイプを入力します。有効な値は `APPLICATION` または `CONTAINER` です。
6. [Sharing Scope] フィールドに、共有スコープを入力します。有効な値は `Sharable` または `Unsharable` です。
7. [作成] をクリックします。

セキュリティ制約のコンフィグレーション

セキュリティ制約は、指定されたリソースにセキュリティを適用します。セキュリティ制約は任意の数だけ作成できます。詳細については、「Web アプリケーション内のリソースへのアクセスに制約を適用する」（`http://edocs.beasys.co.jp/e-docs/wls61/webapp/security.html#resources-webapp`）を参照してください。

1. 左パネルで [Security Constraints] ノードを右クリックし、[新しい SecurityConstraint のコンフィグレーション] を選択します。
2. [Display Name] フィールドに、このセキュリティ制約の名前を入力します。
3. [作成] をクリックします。

4. 左パネルで [Security Constraint] ノードを展開します。
5. 新しく追加したセキュリティ制約のノードを展開します。
6. Web Resource Collection を作成します。Web リソース コレクションは、このセキュリティ制約が適用される URL パターンおよび HTTP メソッドを定義します。
   a. [Web Resource Collection] ノードを右クリックし、[新しい WebResourceCollection の作成] を選択します。
   b. [Web Resource Name] フィールドに、この Web リソース コレクションの名前を入力します。
   c. [Description] フィールドに、この Web リソース コレクションの説明を入力します。（省略可能）
   d. [URL Pattern] フィールドに、1 つ以上の URL パターンを 1 行ずつ区切って入力します。URL パターンは、このセキュリティ制約が適用されるリソースを定義します。
   e. [HTTP Method] ボックスに、HTTP メソッドを入力します。通常、入力する値は `GET` または `POST` です。
   f. [作成] をクリックします。
7. Auth Constraint を作成します。
   a. 編集しているセキュリティ制約の名前を右クリックし、[新しい AuthConstraint のコンフィグレーション] を選択します。
   b. [Description] フィールドに、この認可制約の名前を入力します。（省略可能）
   c. [作成] をクリックします。定義済みのセキュリティ ロールの一覧が [選択可] パネルに表示されます。
   d. 矢印ボタンを使って、定義済みリスト内のセキュリティ ロールを選択済みリストに移動します。注意 : ここでの割り当てを行う前に、1 つ以上のセキュリティ ロールを作成しておく必要があります。詳細については、「セキュリティ ロールのコンフィグレーション」を参照してください。
   e. [適用] をクリックします。
8. User Data Constraint を作成します。

a. 編集中のセキュリティ制約の名前を右クリックし、[新しい UserDataConstraint のコンフィグレーション] を選択します。

b. [Description] フィールドに、このユーザ データ制約の説明を入力します。（省略可能）

c. [Transport Guarantee] ドロップダウン リストから、転送の保証を選択します。有効な値は NONE、INTEGRAL、または CONFIDENTIAL です。

d. [作成] をクリックします。

ログインのコンフィグレーション

ログインのコンフィグレーションでは、ユーザ認証の方式、このアプリケーションで使われるレルム名、フォームによるログイン機能に必要な属性を定義します。

この要素が存在する場合、Web アプリケーションに定義された Security Constraints が適用されるリソースへのアクセス時に、ユーザが認証を受ける必要があります。認証されたユーザは、アクセス権のあるほかのリソースにアクセスするための認可を受けることができます。

1. [Web App Descriptor] ノードを右クリックし、[Config 新しい LoginConfig のコンフィグレーション] を選択します。
2. [Auth Method] ドロップダウン リストから、認可方式を選択します。有効な値は BASIC、FORM、または CLIENT-CERT です。
3. [Realm Name] フィールドに、認証レルムの名前を入力します。
4. [Auth Method] を FORM に設定した場合、認証フォームを含むファイルの URI を [Login Page] フィールドに入力します。
5. [Auth Method] を FORM に設定した場合、認証失敗時にユーザがリダイレクトされるページを含むファイルの URI を [Error Page] フィールドに入力します。

セキュリティ ロールのコンフィグレーション

セキュリティ ロールは、ロールに割り当てられるプリンシパル（通常はユーザ名）のセキュリティ コンテキストを定義します。セキュリティ ロールは任意の数だけ作成できます。詳細については、「Web アプリケーション内のリソースへのアクセスに制約を適用する」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/security.html#resources-webapp）を参照してください。

1. 左パネルで [Security Roles] ノードを右クリックし、[新しい SecurityRole のコンフィグレーション] を選択します。
2. [Description] フィールドに、このセキュリティ ロールの説明を入力します。（省略可能）
3. [Role Name] フィールドに、このセキュリティ ロールの名前を入力します。
4. [作成] をクリックします。

環境エントリのコンフィグレーション

アプリケーションに対して環境エントリを宣言します。環境エントリは任意の数だけ作成できます。

1. 左パネルで [Env Entries] ノードを右クリックし、[新しい EnvEntry のコンフィグレーション] を選択します。
2. [Description] フィールドに、この環境エントリの説明を入力します。（省略可能）
3. [Env Entry Name] フィールドに、この環境エントリの名前を入力します。
4. [Env Entry Value] フィールドに、この環境エントリの値を入力します。
5. [Env Entry Type] ドロップダウンリストから、この環境エントリのタイプを選択します。

EJB 参照のコンフィグレーション

EJB 参照では、EJB リソースへの参照を定義します。この参照は、WebLogic に固有のデプロイメント記述子ファイルである weblogic.xml にマッピングを定義することによって、デプロイメント時の実際の EJB の位置にマッピングされます。EJB 参照は任意の数だけ作成できます。

1. 左パネルで [Ejb Refs] ノードを右クリックし、[新しい EjbRef のコンフィグレーション] を選択します。
2. [Description] フィールドに、この EJB 参照の説明を入力します。（省略可能）
3. [EJBRef Name] フィールドに、この EJB 参照の名前を入力します。
4. [EJBRef Type] フィールドに、参照される EJB の Java クラスを入力します。
5. [Home Interface Name] フィールドに、EJB のホーム インターフェースの名前を入力します。
6. [Remote Interface Name] フィールドに、EJB のリモート インターフェースの名前を入力します。
7. EJB がエンタープライズ アプリケーション アーカイブ（EAR）にパッケージ化される場合、EAR のデプロイメント記述子の <ejb-name> 要素で参照される EJB の名前を [EJBLink Name] フィールドに入力します。
8. [Run As] フィールドに、この Web アプリケーションに定義されているセキュリティ ロールの名前を入力します。（省略可能）

新しい Webapp Ext のコンフィグレーション

左パネルの [WebApp Ext] ノードは、WebLogic に固有のデプロイメント記述子である weblogic.xml ファイルを表現します。weblogic.xml 記述子では、WebLogic Server に固有の属性をコンフィグレーションします。weblogic.xml 内の属性をコンフィグレーションするには、まずデプロイメント記述子エディタでこの記述子のノードを作成する必要があります。

weblogic.xml 記述子のノードを作成するには

1. 左パネルで、トップレベル ノードを右クリックします （ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。
2. [新しい WebAppExtDescriptor のコンフィグレーション] を選択します。
3. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）
4. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）
5. [作成] をクリックします。

セッション記述子のコンフィグレーション

Session Descriptor では、HTTP セッション関連のパラメータをコンフィグレーションします。

1. 左パネルに WebApp Ext ノードが存在しない場合、このノードを作成します。
   a. 左パネルで、トップレベル ノードを右クリックします （ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。
   b. [新しい WebAppExtDescriptor のコンフィグレーション] を選択します。
   c. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）
   d. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）
   e. [作成] をクリックします。
   f. 左パネルで [WebApp Ext] ノードを展開します。
2. 左パネルで [Session Descriptor] ノードをクリックします。
3. Session Descriptor の表中の情報を参考にして、右パネルのフィールドの値を入力します。

4. [適用] をクリックします。

JSP 記述子のコンフィグレーション

JSP Descriptor では、JSP 関連のパラメータを設定します。

1. 左パネルに WebApp Ext ノードが存在しない場合、このノードを作成します。
   a. 左パネルで、トップレベル ノードを右クリックします （ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。
   b. [新しい WebAppExtDescriptor のコンフィグレーション] を選択します。
   c. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）
   d. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）
   e. [作成] をクリックします。
   f. 左パネルで [WebApp Ext] ノードを展開します。
2. 左パネルで [JSP Descriptor] ノードをクリックします。
3. JSP Descriptor の表中の情報を参考にして、右パネルのフィールドの値を入力します。
4. [適用] をクリックします。

セキュリティ ロール割り当てのコンフィグレーション

セキュリティ ロールの割り当ては、セキュリティ ロールを 1 つまたは複数のプリンシパルにマッピングします。プリンシパルはセキュリティ レルム内に定義しなければなりません。

1. 左パネルに WebApp Ext ノードが存在しない場合、このノードを作成します。

a. 左パネルで、トップレベル ノードを右クリックします（ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。

b. [新しい WebAppExtDescriptor のコンフィグレーション] を選択します。

c. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）

d. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）

e. [作成] をクリックします。

f. 左パネルで [WebApp Ext] ノードを展開します。

2. 左パネルで [Security Role Assignment] ノードを右クリックし、[新しい SecurityRoleAssignment のコンフィグレーション] を選択します。
3. [Role] ドロップダウン リストからセキュリティ ロールを選択します。
4. [Principal Names] テキスト ボックスに、1 つ以上の Principal Names を 1 行ずつに区切って追加します。セキュリティ レルム内で有効なプリンシパルを指定する必要があります。
5. [作成] をクリックします。

文字セット パラメータのコンフィグレーション

文字セット パラメータを設定することにより、Unicode を使用しない運用を行うためのコードセットの動作を定義できます。Java の文字セットから IANA 文字セットへのマッピングを指定することもできます。詳細については、「HTTP リクエストのエンコーディングの指定」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/components.html#encoding-http）および「IANA 文字セットから Java 文字セットへのマッピング」（http://edocs.beasys.co.jp/e-docs/wls61/webapp/components.html#map-iana）を参照してください。

1. 左パネルに WebApp Ext ノードが存在しない場合、このノードを作成します。

a. 左パネルで、トップレベル ノードを右クリックします （ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。

b. [Configure a new WebAppExtDescriptor 新しい WebAppExtDescriptor のコンフィグレーション] を選択します。

c. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）

d. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）

e. [作成] をクリックします。

f. 左パネルで [WebApp Ext] ノードを展開します。

2. 左パネルで [WebApp Ext] ノードを右クリックし、[新しい CharsetParams のコンフィグレーション] を選択します。

3. Input Charset Descriptors を作成します。

a. [Charset Params] ノードを展開します。

b. [Input Charset Descriptors] ノードを右クリックし、[新しい InputCharsetDescriptor のコンフィグレーション] を選択します。

c. [Resource Path] フィールドに、この文字セット記述子に適用されるパスを入力します。

d. [Java Charset Name] フィールドに、有効な Java の文字セット名を入力します。

e. [作成] をクリックします。

4. Character Set Mapping を作成します。

a. [Charset Params] ノードを展開します。

b. [Charset Mappings] ノードを右クリックし、[新しい CharsetMapping のコンフィグレーション] を選択します。

c. [IANA Charset Name] フィールドに、有効な IANA 文字セット名を入力します。

d. [Java Charset Name] フィールドに、有効な Java の文字セット名を入力します。

e. [作成] をクリックします。

リファレンス記述子のコンフィグレーション

リファレンス記述子は、サーバ リソースの JNDI 名を、Web アプリケーションで使われる名前にマッピングします。[Resource Description] パネルでは、データソースなどのリソースをその JNDI 名にマッピングします。[Ejb Reference] パネルでは、EJB をその JNDI 名にマッピングします。

1. 左パネルに WebApp Ext ノードが存在しない場合、このノードを作成します。

   a. 左パネルで、トップレベル ノードを右クリックします （ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。

   b. [新しい WebAppExtDescriptor のコンフィグレーション] を選択します。

   c. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）

   d. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）

   e. [作成] をクリックします。

   f. 左パネルで [WebApp Ext] ノードを展開します。

2. 左パネルで [WebApp Ext] ノードを右クリックし、[新しい ReferenceDescriptor のコンフィグレーション] を選択します。

3. Resource Descriptions を作成します。1 つまたは複数のリソース記述子を作成できます。

   a. [Resource Descriptor] ノードを展開します。

   b. [Resource Descriptions] ノードを右クリックし、[新しい ResourceDescription のコンフィグレーション] を選択します。

   c. [Jndi Name] フィールドに、JNDI ツリーでのオブジェクトの名前の名前を入力します。

d. [Resource Reference] ドロップダウンリストからリソース参照を選択します。事前に [Resource Refs] タブで、リソース参照をコンフィグレーションしておく必要があります。

e. [作成] をクリックします。

4. EJB Reference Description を作成します。

a. [Resource Descriptor] ノードを展開します。

b. [Ejb Reference Descriptions] ノードを右クリックし、[新しい EjbReferenceDescription のコンフィグレーション] を選択します。

c. [Jndi Name] フィールドに、JNDI ツリーでのオブジェクトの名前の名前を入力します。

d. [Ejb Reference] ドロップダウンリストから EJB 参照を選択します。事前に [Ejb Refs] タブで、EJB 参照をコンフィグレーションしておく必要があります。

e. [作成] をクリックします。

コンテナ記述子のコンフィグレーション

サーブレットから JNDI に転送されるリクエストの認証が必要なときは、転送時の認証チェックを有効にします。

1. 左パネルに WebApp Ext ノードが存在しない場合、このノードを作成します。

a. 左パネルで、トップレベル ノードを右クリックします （ノードには *myWebApp* Web Application というラベルが付いています。*myWebApp* は、編集中の Web アプリケーションの名前です）。

b. [新しい WebAppExtDescriptor のコンフィグレーション] を選択します。

c. 右パネルの [Description] フィールドに、Web アプリケーションの説明を入力します。（省略可能）

d. アプリケーションを実行している WebLogic Server のバージョンを入力します。（省略可能）

e. [作成] をクリックします。

f. 左パネルで [WebApp Ext] ノードを展開します。

2. [WebApp Ext] ノードを右クリックし、[新しい ContainerDescriptor のコンフィグレーション] を選択します。
3. 転送されるリクエストの認証を有効にするには、[Check Auth on Forward Enabled] チェック ボックスをオンにします。
4. リダイレクト動作をコンフィグレーションするには、[redirect-with-absoute-url] チェックボックスをチェックします。
5. [作成] をクリックします。

Web App Descriptor

属性	説明	値の範囲	デフォルト値
Description	テキストによるサーブレットの説明。	文字列	なし
Display Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Large Icon File Name	GUI ツールで Web アプリケーションを表現するために使われる、（32 × 32 ピクセルの）大きなサイズの .gif または .jpg 画像の位置を指定する。現時点で、この要素は WebLogic Server では使用しない。	文字列	なし
Small Icon File Name	GUI ツールで Web アプリケーションを表現するために使われる、（16 × 16 ピクセルの）小さなサイズの .gif または .jpg 画像の位置を指定する。現時点で、この要素は WebLogic Server では使用しない。	文字列	なし
Distributable	この要素は、WebLogic Server では使用しない。	なし	なし

Filters

属性	説明	値の範囲	デフォルト値
Filter Name	このフィルタの名前。	文字列	MyFilter
Description	テキストによるサーブレットの説明。	文字列	なし
Display Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Small Icon File Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Large Icon File Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Filter Class	フィルタの完全修飾クラス名。	クラス名	なし

Filter Mappings

属性	説明	値の範囲	デフォルト値
Filter	URL パターンのマッピングの相手であるフィルタの名前。この名前は、<filter-name> 要素でフィルタに割り当てた名前に対応する。	文字列	なし
Url Pattern	URL の解決に使われるパターンを記述する。WebLogic Server によって、URL のうち「http://host:port + WebAppName」に続く部分が <url-pattern> 要素の値と比較される。パターンがマッチすると、この要素にマッピングされたフィルタが呼び出される。 以下はパターンの例である。 /soda/grape/* /foo/* /contents *.foo URL の書式は、Servlet 2.2 仕様のセクション 10 で定められた規則に従う必要がある。	文字列	なし

属性	説明	値の範囲	デフォルト値
Servlet	呼び出された場合にこのフィルタが実行されるサーブレットの名前。	このデプロイメント記述子に定義された有効なサーブレット名	なし

Listeners

属性	説明	値の範囲	デフォルト値
Listener Class Name	Web アプリケーションのイベントに応答するクラスの名前。	クラス名	なし

Servlets

属性	説明	値の範囲	デフォルト値
Description	テキストによるサーブレットの説明。	文字列	なし
Display Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Small Icon File Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Large Icon File Name	この要素は、WebLogic Server では使用しない。	文字列	なし
Servlet Name	サーブレットの標準名を定義する。デプロイメント記述子内のほかの場所でサーブレット定義を参照するために使われる。	文字列	MyServlet
Servlet Class	サーブレットの完全修飾クラス名。 サーブレット本体内では、<servlet-class> タグまたは <jsp-file> タグのどちらか一方だけを使用できる。	クラス名	なし

属性	説明	値の範囲	デフォルト値
Jsp File	Web アプリケーション内部の JSP ファイルへの絶対パス。Web アプリケーションのルートディレクトリからの相対位置で指定する。 サーブレット本体内では、<servlet-class> タグまたは <jsp-file> タグのどちらか一方だけを使用できる。	文字列	なし
Load On Startup	このサーブレットは WebLogic Server の起動時に初期化される。この要素は省略可能で、サーブレットのロード順を示す正の整数を指定する。この値の小さいサーブレットから順にロードされる。	正の整数 値を指定しないか、指定した値が正の整数でない場合、WebLogic Server が起動シーケンスでサーブレットをロードする順序は任意となる。	0

Parameters

属性	説明	値の範囲	デフォルト値
Description	初期化パラメータのテキストによる説明。		MyError Page
Param Name	この初期化パラメータの名前を定義する。	文字列	なし

属性	説明	値の範囲	デフォルト値
Param Value	この初期化パラメータの `String` 型の値を定義する。	文字列	なし

Security Role Refs

属性	説明	値の範囲	デフォルト値
Description	テキストによるロールの説明。	文字列	MyError Page
Role Name	サーブレットのコードで使われるセキュリティ ロールまたはプリンシパルの名前を定義する。	文字列	なし
Role Link	セキュリティ ロールの名前を定義する。	文字列 - デプロイメント記述子で、後から `<security-role>` 要素に定義されるセキュリティ ロール	なし

Servlet Mappings

属性	説明	値の範囲	デフォルト値
Servlet	URL パターンのマッピングの相手であるサーブレットの名前。この名前は、<servlet> 宣言タグでサーブレットに割り当てた名前に対応する。	宣言済みのサーブレット	なし
URL Pattern	URL の解決に使われるパターンを記述する。WebLogic Server によって、URL のうち「http://host:port + WebAppName」に続く部分が <url-pattern> 要素の値と比較される。パターンがマッチすると、この要素にマッピングされたサーブレットが呼び出される。 以下はパターンの例である。 /soda/grape/* /foo/* /contents *.foo URL の書式は、Servlet 2.2 仕様のセクション 10 で定められた規則に従う必要がある。	文字列	MyServlet Mapping

Mime Mappings

属性	説明	値の範囲	デフォルト値
Mime Type	定義済みの MIME タイプを表す文字列。例 : `text/plain`	文字列	MyMime Mapping
Extension	拡張子を表すテキスト。例 : `txt`	文字列	なし

Session Config

属性	説明	値の範囲	デフォルト値
Session Timeout	この Web アプリケーションでセッションが失効するまでの時間 (分)。この要素に設定する値は、右に示した特殊な値を設定する場合を除いて、WebLogic 固有のデプロイメント記述子である `weblogic.xml` 内の Session Descriptor の `TimeoutSecs` パラメータに設定された値よりも優先される。	上限値 : Integer.MAX_VALUE ˜ 60 特殊な値 : ♦ -2 = Session Descriptor の `TimeoutSecs` によって設定される値を使用する ♦ -1 = セッションはタイムアウトしない。Session Descriptor に設定された値は無視される	-2

Welcome Files

属性	説明	値の範囲	デフォルト値
Welcome Files	デフォルトのウェルカム ファイルとして使用するファイルの名前（index.html など）。1 つまたは複数のウェルカム ファイルを指定できる。	文字列	なし

Error Pages

属性	説明	値の範囲	デフォルト値
Error Code	有効な HTTP エラー コード（例 : `404`）	文字列	MyError Page
Exception Type	Java の例外タイプの完全修飾クラス名（例 : `java.lang`）	文字列	なし
Location	エラーへの応答として表示されるリソースの位置。先頭に / を付けて指定する。例 : `/myErrorPg.html`	文字列	なし

Tag Libs

属性	説明	値の範囲	デフォルト値
URI	Web アプリケーションで使われるタグ ライブラリを識別する URI を、web.xml ドキュメントの位置からの相対位置で記述する。 指定した URI と、JSP ページ上の taglib ディレクティブで使われている URI 文字列が一致する場合に、このタグ ライブラリが使われる。	文字列	MyTag Lib
Location	タグ ライブラリのファイル名を、Web アプリケーションのルートからの相対位置で指定する。タグ ライブラリの記述子ファイルは、HTTP リクエストを通じて自由にアクセスされることを防ぐために、WEB-INF ディレクトリの下に格納することが推奨される。	文字列	なし

Resource Env Refs

resource-env-ref 要素では、コンポーネントの環境内のリソースと関連付けられる管理対象オブジェクトを指す、コンポーネントの参照の宣言を定義します。

属性	説明	値の範囲	デフォルト値
Description	テキストによる説明。	文字列	なし
Ref Name	リソース環境参照の名前。この値は、コード内で使われる環境エントリ名である。	文字列	MyResource Env Ref
Ref Type	リソース環境参照のタイプ。	文字列	なし

Resource Refs

属性	説明	値の範囲	デフォルト値
Description	テキストによる説明。	文字列	なし
Ref Name	JNDI ツリー内で使われるリソース名。Web アプリケーション内のサーブレットは、この名前を使用してリソースへの参照をルックアップする。	文字列	MyResource Ref
Ref Type	参照名に対応するリソースの Java 型。Java 型の完全なパッケージ名を使用する。	Java 型	なし
Auth	リソース サインオンに適用するセキュリティを設定する。	APPLICATION: アプリケーション コンポーネントのコードは、プログラムによってリソース サインオンを実行する。 CONTAINER: WebLogic Server は、login config 要素によって確立されたセキュリティ コンテキストを使用する。	なし
Sharing Scope	特定のリソース マネージャ接続ファクトリ参照を通じて取得される接続が共有可能かどうかを指定する。	Sharable Unsharable	Sharable

Security Constraints

属性	説明	値の範囲	デフォルト値
Display Name	この制約の名前。	文字列	MySecurity Constraint

Auth Constraint

このセキュリティ制約に定義される Web リソースのコレクションに対して、どのグループまたはプリンシパルがアクセス権を持つかを定義します。

属性	説明	値の範囲	デフォルト値
Description	テキストによるこのセキュリティ制約の説明。	文字列	なし
Role Name	このセキュリティ制約に定義されるリソースにアクセス可能なセキュリティ ロールを定義する。セキュリティ ロール名は、`Role Name` を使用してプリンシパルにマッピングされる。	定義済みのセキュリティ ロール	なし

User Data Constraint

クライアントがサーバと通信する方式を定義します。

属性	説明	値の範囲	デフォルト値
Description	テキストによる説明。	文字列	なし
Transport Guarantee	クライアントおよびサーバ間の通信を指定する。 ユーザが INTEGRAL または CONFIDENTIAL 制約を使用して認証されると、WebLogic Server は SSL（Secure Sockets Layer）接続を確立する。	♦ NONE– アプリケーションで転送の保証が不要 ♦ INTEGRAL– アプリケーションにおいて、転送中に変更されることがないような方式で、クライアントおよびサーバ間でデータを送受信する必要がある ♦ CONFIDENTIAL– アプリケーションにおいて、通信に関係しないエンティティが転送内容を傍受できないような方式でデータを送受信する必要がある	なし

Web Resource Collection

このセキュリティ制約が適用される Web アプリケーションのコンポーネントを定義します。

属性	説明	値の範囲	デフォルト値

Web Resource Name	この Web リソース コレクションの名前。	文字列	なし
Description	テキストによるこのセキュリティ制約の説明。	文字列	なし
URL Pattern	1 つまたは複数の URL Pattern 要素を使用して、このセキュリティ制約がどの URL パターンに適用されるかを宣言する。URL Pattern 要素を 1 つも使用しない場合、この Web リソース コレクションは WebLogic Server に無視される。	文字列	なし
HTTP Method	1 つまたは複数の HTTP Method 要素を使用して、どの HTTP メソッド（GET | POST | . . .）に認可制約が適用されるかを宣言する。HTTP Method 要素を省略すると、デフォルトですべての HTTP メソッドにセキュリティ制約が適用される。	GET POST	なし

LoginConfig

属性	説明	値の範囲	デフォルト値

Auth Method	ユーザ認証の方式を指定する。有効な値は以下のとおり。 BASIC - ブラウザ認証を使用する FORM - ユーザが作成する HTML フォームを使用する CLIENT-CERT	BASIC - ブラウザ認証を使用する FORM - ユーザが作成する HTML フォームを使用する CLIENT-CERT	BASIC
Realm Name	ユーザ資格を認証するために参照されるレルムの名前。省略すると、デフォルトで WebLogic レルムが使われる。詳細については、「セキュリティ レルムの指定」（http://edocs.beasys.co.jp/e-docs/wls61/adminguide/cnfgsec.html#cnfgsec004）を参照のこと。	文字列	なし
Login Page	ユーザ認証に使われる Web リソースの URI。ドキュメント ルートからの相対位置で指定する。リソースは HTML ページ、JSP、または HTTP サーブレットのいずれかで、特定の命名規約に準拠したフォームを含む HTML ページを返さなければならない。	文字列	なし

Error Page	認証ログインの失敗時にユーザに送られる Web リソースの URI。ドキュメント ルートからの相対位置で指定する。	文字列	なし

Security Roles

属性	説明	値の範囲	デフォルト値
Description	テキストによるこのセキュリティ ロールの説明。	文字列	なし
Role Name	ロール名。ここで指定する名前に対応し、セキュリティ レルム内のプリンシパルにロールをマッピングするエントリが、WebLogic 固有のデプロイメント記述子である `weblogic.xml` に存在しなければならない。	文字列	system

Env Entries

属性	説明	値の範囲	デフォルト値
Description	テキストによる説明。	文字列	なし
Env Entry Name	環境エントリの名前。	文字列	MyEnvironment Entry
Env Entry Value	環境エントリの値。	文字列	なし
Env Entry Type	環境エントリのタイプ。	文字列	なし

Ejb Refs

属性	説明	値の範囲	デフォルト値
Description	テキストによる参照の説明。	文字列	なし
EJBRef Name	Web アプリケーションで使われる EJB の名前。この名前は、WebLogic 固有のデプロイメント記述子である `weblogic.xml` において JNDI ツリーにマッピングされる。	文字列	MyEJB Ref
EJBRef Type	参照される EJB に対して想定される Java クラス型。	文字列	なし
Home Interface Name	EJB のホーム インターフェースの完全修飾クラス名。	文字列	なし
Remote Interface Name	EJB のリモート インターフェースの完全修飾クラス名。	文字列	なし
EJBLink Name	J2EE アプリケーション パッケージの作成時に使われる EJB の名前 (`<ejb-name>`)。	文字列	なし
Run As	参照先の EJB のそのセキュリティ コンテキストが適用されるセキュリティ ロール。	この Web アプリケーションで定義済みのセキュリティ ロール	なし

Security Role Assignment

属性	説明	値の範囲	デフォルト値
Role	セキュリティ ロールの名前を指定する。	有効なセキュリティ ロール	なし
Principal Names	セキュリティ レルム内で定義されたプリンシパルの名前を指定する。プリンシパルをロールにマッピングするために、複数の <principal-name> 要素を使用できる。セキュリティ レルムの詳細については、「WebLogic セキュリティのプログラミング」（http://edocs.beasys.co.jp/e-docs/wls61/security/index.html）を参照のこと。	セキュリティ レルム内で定義されたプリンシパル	なし

Reference Descriptor

Resource Descriptions

属性	説明	値の範囲	デフォルト値
Res souce Reference	リソース参照の名前を指定する。	文字列	なし
Jndi Name	リソースの JNDI 名を指定する。	Java の文字セット名	なし

EJB Reference Description

属性	説明	値の範囲	デフォルト値
Ejb Reference	Web アプリケーションで使われる EJB 参照の名前を指定する。	文字列	なし
Jndi Name	参照の JNDI 名を指定する。	Java の文字セット名	なし

Session Descriptor

:

パラメータ名	デフォルト値	パラメータ値
URLRewritingEnabled	true	URL 書き換えを有効にする。これは、セッション ID をコード化して URL に埋め込むことによって、ブラウザでクッキーが無効にされている場合にもセッション追跡を可能にする機能である。
IDLength	52	セッション ID のサイズを設定する。 最小値は 8 バイトで、最大値は `Integer.MAX_VALUE` である。 WAP アプリケーションを記述する場合、WAP プロトコルではクッキーがサポートされないため、URL 書き換えを使用しなければならない。また、一部の WAP デバイスでは（パラメータを含めた）URL の長さに 128 文字の制限があるため、URL 書き換えを使用して転送できるデータの量も制限される。パラメータを通じてより多くの情報を伝達できるようにするには、このパラメータを使用して、WebLogic Server によってランダムに生成されるセッション ID のサイズを制限する。
CookieComment	Weblogic Server Session Tracking Cookie	クッキー ファイル内でセッション追跡クッキーを識別するコメントを指定する。 このパラメータを設定しない場合のデフォルト値は「`WebLogic Session Tracking Cookie`」である。アプリケーションに合わせた、より特殊な名前の指定が可能である。

パラメータ名	デフォルト値	パラメータ値
CookieDomain	NULL	ブラウザがリクエストを行うときにクッキー情報を送信するサーバを識別する。たとえば、CookieDomain を .mydomain.com に設定すると、*.mydomain.com ドメイン内のすべてのサーバにクッキーが返される。 ドメイン名には少なくとも 2 つのコンポーネントが必要である。設定値「*.com」または「*.net」は無効である。 このパラメータを設定しない場合のデフォルト値は、クッキーを発行したサーバである。
CookieMaxAgeSecs	-1	セッション クッキーがクライアント上で失効するまでの寿命を秒単位で設定する。 値が 0 の場合、クッキーは直ちに失効する。 最大の値は MAX_VALUE で、これはクッキーが失効しないことを表す。 -1 に設定すると、クッキーはユーザがブラウザを終了した時点で失効する。
CookieName	JSESSIONID	セッション クッキーの名前を定義する。設定しない場合のデフォルト値は JSESSIONID である。このパラメータには、アプリケーションに合わせてさらに特殊な名前を設定できる。
CookiePath	NULL	ブラウザによるクッキー送信先のパスを指定する。 このパラメータを設定しない場合のデフォルト値は /（スラッシュ）である。このとき、ブラウザは WebLogic Server が管理するすべての URL にクッキーを送信する。このパスをより狭い範囲のマッピングに設定し、ブラウザによるクッキー送信先のリクエスト URL を限定することができる。

パラメータ名	デフォルト値	パラメータ値
InvalidationIntervalSecs	60	タイムアウトして無効になったセッションに対してハウスクリーニング チェックを実行し、古いセッションを削除してメモリを解放するまで WebLogic Server が待機する時間を秒単位で設定する。このパラメータは、トラフィックの多いサイト上で最適なパフォーマンスを得られるように、WebLogic Server をチューニングするために使用する。 最小値は 1（毎秒）である。最大値は 604,800（週に 1 回）である。パラメータを設定しない場合のデフォルト値は 60 秒である。
JDBCConnectionTimeoutSecs	120	JDBC 接続をタイムアウトするまで WebLogic Server が待機する時間を秒単位で設定する。x は間隔を表す秒数である。
PersistenceStoreDir	session_db	PersistentStoreType を file に設定する場合、このパラメータで、WebLogic Server がセッションを格納するディレクトリパスを設定する。ディレクトリ パスは、一時ディレクトリからの相対パスまたは絶対パスのどちらかで指定する。一時ディレクトリは、Web アプリケーションの WEB-INF ディレクトリの下に生成されるディレクトリ、または context-param 要素の javax.servlet.context.tmpdir によって指定されるディレクトリのどちらかである。 **各セッションのサイズに有効セッション数**を乗じて、すべてのセッションを格納するのに十分なディスク容量を確保しておくこと。セッションのサイズは、PersistenceStoreDir に作成されるファイルを調べることによって確認できる。 このディレクトリを複数のサーバ間での共有ディレクトリにすることによって、ファイルベースの永続性を使用するセッションをクラスタ対応にすることができる。 このディレクトリは手動で作成しなければならない。

パラメータ名	デフォルト値	パラメータ値
PersistenceStorePool	なし	永続性ストレージのために使用する JDBC 接続プールの名前を指定する。
PersistentStoreType	memory	以下のいずれかの永続ストア方式を設定する。 ♦ memoryó セッションの永続ストレージを無効にする ♦ fileó ファイルベースの永続性を使用する（前の PersistenceStoreDir の説明も参照） ♦ jdbcó 永続セッションの格納にデータベースを使用する （前の PersistenceStorePool の説明も参照） ♦ replicatedó memory と同じだが、クラスタを構成するサーバ間でセッション データをレプリケートする ♦ cookieó すべてのセッション データを、ユーザのブラウザ内のクッキーに格納する
CookiesEnabled	True	セッション クッキーの使用はデフォルトで有効であり、また推奨されているが、このプロパティを false に設定するとセッション クッキーを無効にすることができる。このオプションを無効にして、URI 書き換えの使用をテストすることができる。
TrackingEnabled	True	true に設定すると、セッション追跡が有効になる。

パラメータ名	デフォルト値	パラメータ値
TimeoutSecs	3600	セッションをタイムアウトするまで WebLogic Server が待機する時間を秒単位で設定する。x はセッションのアクティビティ間の秒数である。 最小値は 1、デフォルト値は 3600、最大値は integer MAX_VALUE である。 ヒット数の多いサイトでは、セッションのタイムアウトを調整することによってアプリケーションのチューニングが可能である。ブラウザクライアントにはセッションを終了するあらゆる機会を与えることが望ましいが、その一方で、ユーザがサイトを立ち去った場合、またはその他の理由によってセッションを終了した場合には、そのセッションからサーバを速やかに解放することが望ましい。 このパラメータの設定は、web.xml 内の session-timeout 要素の値（分単位で定義する）によって優先的に置き換えることができる。詳細については、「Session Timeout」を参照のこと。
ConsoleMainAttribute		WebLogic Server Administration Console でセッション モニタを有効にする場合、このパラメータに、モニタ対象の各セッションを識別するために使用するセッション パラメータの名前を設定する。
PersistentStoreCookieName	WLCOOKIE	クッキーベースの永続性で使われるクッキーの名前を設定する。

JSP Descriptor

パラメータ名	デフォルト値	パラメータ値
Compile Command	javac、またはコンフィグレーション対象のサーバに対して定義された Java コンパイラ WebLogic Server Administration Console の /tuning タブ	生成される JSP サーブレットのコンパイルに使われる標準 Java コンパイラの絶対パス名を指定する。 処理を高速にするためには、IBM の Jikes や Symantec の sj などの異なったコンパイラを指定する。
Compiler Class	なし	WebLogic Server の仮想マシンにで実行される Java コンパイラの名前（javac や sj などの実行可能コンパイラの代わりに使われる）。
Compile Flags	なし	コンパイラに 1 つまたは複数のコマンドライン フラグを渡す。複数のフラグは引用符で囲み、スペースで区切って指定する。次に例を示す。 `java weblogic.jspc -compileFlags "-g -v" myFile.jsp`
Working Dir	初期生成されるディレクトリ	JSP に対して生成された Java クラス ファイルおよびコンパイル済みクラス ファイルを WebLogic Server が保存するディレクトリの名前。
Verbose	true	true に設定すると、デバッグ情報がブラウザ、コマンド プロンプト、および WebLogic Server のログ ファイルに出力される。

パラメータ名	デフォルト値	パラメータ値
keepgenerated	`false`	JSP のコンパイル プロセスで中間ステップとして生成される Java ファイルを保存する。このパラメータを `true` に設定しない場合、中間の Java ファイルはコンパイル後に削除される。
Page Check Seconds	`1`	JSP ファイルに変更があり、再コンパイルが必要かどうかを WebLogic Server がチェックする間隔を秒単位で設定する。ファイルが変更されている場合、依存性も同時にチェックされ、再帰的に再ロードされる。 このパラメータを 0 に設定すると、リクエストのたびにページがチェックされる。`-1` に設定すると、ページのチェックおよび再コンパイルは無効になる。
Encoding	使用しているプラットフォームでのデフォルトのエンコーディング	JSP ファイルで使われるデフォルトの文字セットを指定する。標準の Java 文字セット名（http://java.sun.com/j2se/1.3/docs/guide/intl/encoding.doc.htm を参照）を使用する。 このパラメータを設定しない場合のデフォルト値は、プラットフォームで使われているエンコーディングである。 この設定値は、JSP ページのディレクティブ（JSP コードに記述される）によって優先的に置き換えられる。次に例を示す。 `<%@ page contentType="text/html; charset=custom-encoding"%>`
Package Prefix	`jsp_servlet`	すべての JSP ページがコンパイルされるパッケージを指定する。
No Try Blocks	`false`	JSP ファイルで、カスタムの JSP タグが多く存在するかまたは深くネストされ、コンパイル時に `java.lang.VerifyError` 例外が発生する場合、このフラグを使用すると JSP を正しくコンパイルすることができる。
Precompile	`false`	true に設定すると、WebLogic Server は起動時にすべての JSP を自動的にコンパイルする。

パラメータ名	デフォルト値	パラメータ値
Compiler Supports Encoding	False	中間形式の .java ファイルを作成するときに WebLogic JSP コンパイラが使用する使われるエンコーディングを指定する。 true に設定すると、JSP コンパイラは JSP ページ上の page ディレクティブの contentType 属性で指定されたエンコーディングを使用する。contentType が指定されていない場合、jsp-descriptor 内の encoding パラメータで定義されたエンコーディングが使われる。 false に設定すると、JSP コンパイラは中間形式の .java ファイルの作成時に、JVM でのデフォルトのエンコーディングを使用する。

Container Descriptor

属性	説明	値の範囲	デフォルト値
Check Auth on Forward Enabled	有効にすると、サーブレットまたは JSP から転送されたリクエストの認証が必要になる。	選択または未選択	未選択

属性	説明	値の範囲	デフォルト値
redirect-with-absoute-url	`javax.servlet.http.HttpServletResponse.SendRedirect()` メソッドが相対 URL、絶対 URL のどちらを使用してリダイレクトするかを制御する。プロキシ HTTP サーバを使用しており、URL を絶対リンクに変換しないようにするには、この要素を `false` に設定する。 デフォルトの動作では、URL を絶対リンクに変換する。	ブール値	true

Charset Params

Input Charset Descriptors

属性	説明	値の範囲	デフォルト値
Resource Path	リクエストの URL に埋め込まれた場合に、[Java Charset Name] フィールドに指定された Java 文字セットを使用するように WebLogic Server に知らせるパス。	文字列	なし

属性	説明	値の範囲	デフォルト値
Java Charset Name	使用する Java の文字セットを指定する。	Java の文字セット名	なし

Character Set Mapping

属性	説明	値の範囲	デフォルト値
IANA Charset Name	[Java Charset Name] フィールドに指定される Java 文字セットにマッピングされる IANA 文字セット名を指定する。	IANA 文字セット名	なし
Java Charset Name	使用する Java の文字セットを指定する。	Java の文字セット名	なし

WebLogic Web Service のコンフィグレーション

デプロイメント記述子エディタでは、値の確認のみを目的として Web Service 関連の情報を表示します。値を変更すると、デプロイされている Web Service が正常に機能しなくなる可能性があります。

Web Services

RPC Services

属性	説明	値の範囲	デフォルト値
JNDI Name	RPC スタイルの Web サービスに準拠するステートレス セッション EJB の JNDI 名。	文字列	なし
Home Interface	ステートレス セッション EJB のホーム インターフェース。	文字列	なし
Remote Interface	ステートレス セッション EJB のリモート インターフェース。	文字列	なし
URI	クライアント アプリケーションが Web サービスを起動するために使用する URI。	文字列	なし

Message Services

属性	説明	値の範囲	デフォルト値
Service Name	WebLogic Server とクライアント アプリケーションの間で受け渡しされる SOAP メッセージを処理する SOAP サーブレットの名前。	文字列	NULL
Destination	WebLogic Server とクライアント アプリケーションの間でデータを送受信する JMS トピックまたはキューの JNDI 名。	文字列	なし
Destination Type	JMS 送り先のタイプ（トピックまたはキュー）。	`Topic` または `Queue`	なし

属性	説明	値の範囲	デフォルト値
Action	メッセージスタイルのこの Web サービスを起動するクライアント アプリケーションが、JMS の送り先にデータを送るのか、それとも JMS の送り先からデータを受け取るのかを指定する。 クライアントが JMS の送り先にデータを送る場合は send を指定し、クライアントが JMS の送り先からデータを受け取る場合は receive を指定する。	Send または Receive	なし
Connection Factory	JMS の送り先への接続を作成するために使われる接続ファクトリの JNDI 名。	文字列	なし
URI	クライアントが Web サービスを起動するために使用する URI。	文字列	なし

114 実行時 WLEC 接続プール

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前
- ドメイン
- プライマリ アドレス
- フェイルオーバ アドレス
- 最小プール サイズ
- 最大プール サイズ

115 WLEC 接続プール

以下に、Administration Console を使用して、WLEC 接続プールのコンフィグレーションや管理に必要な属性を設定する手順を説明します。詳細については、『WebLogic Enterprise Connectivity ユーザーズ ガイド』を参照してください。

新規 WLEC 接続プールのコンフィグレーション

1. 左ペインの [WLEC] ノードをクリックします。右ペインに [WLEC 接続プール] テーブルが表示され、ドメインで定義されているすべての WLEC 接続プールが示されます。
2. [新しい WLEC Connection Pool のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成する WLEC 接続プールの設定に関連するタブが示されます。
3. [名前]、[プライマリ アドレス]、[フェイルオーバー アドレス]、[ドメイン]、[最小プール サイズ]、および [最大プール サイズ] 属性フィールドに値を入力します。
4. 右下隅の [作成] ボタンをクリックして、[名前] フィールドで指定した名前で WLEC 接続プールのインスタンスを作成します。左ペインの [WLEC] ノードの下に、新しいインスタンスが追加されます。
5. [適用] をクリックします。
6. [セキュリティ] タブをクリックして、属性フィールドを変更するか、割り当てられているデフォルト値をそのまま使用します。
7. [適用] をクリックして、変更を保存します。

WLEC 接続プールのクローンの作成

1. 左ペインの [WLEC] ノードをクリックします。右ペインに [WLEC 接続プール] テーブルが表示され、ドメインで定義されているすべての WLEC 接続プールが示されます。
2. クローンを作成する WLEC 接続プールの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、WLEC 接続プールのクローンの作成に関連するタブが示されます。
3. [名前]、[プライマリ アドレス]、[フェイルオーバー アドレス]、[ドメイン]、[最小プール サイズ]、および [最大プール サイズ] 属性フィールドに値を入力します。
4. 右下隅の [クローン] ボタンをクリックして、[名前] フィールドで指定した名前で WLEC 接続プールのインスタンスを作成します。左ペインの [WLEC] ノードの下に、新しいインスタンスが追加されます。
5. [適用] をクリックします。
6. [セキュリティ] タブをクリックして、属性フィールドを変更するか、割り当てられているデフォルト値をそのまま使用します。
7. [適用] をクリックして、変更を保存します。

WLEC 接続プールの削除

1. 左ペインの [WLEC] ノードをクリックします。右ペインに [WLEC 接続プール] テーブルが表示され、ドメインで定義されているすべての WLEC 接続プールが示されます。
2. 削除する WLEC 接続プールの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、WLEC 接続プールを削除します。[WLEC] ノードの下にある WLEC 接続プール アイコンが削除されます。

WLEC 接続プールのすべてのインスタンスのモニタ

1. 左ペインの [WLEC] ノードをクリックします。右ペインに [WLEC 接続プール] テーブルが表示され、ドメインで定義されているすべての WLEC 接続プールが示されます。
2. モニタする WLEC 接続プールの行で [すべてのインスタンスのモニタ] アイコンをクリックします。右ペインにダイアログが表示され、サーバ ドメイン全体にデプロイされている WLEC 接続プールのすべてのインスタンスが示されます。

WLEC 接続プールの割り当て

1. 左ペインの [WLEC 接続プール] の下で、割り当てるプールのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [サーバ] および [クラスタ] タブについて次の手順を実行します。
   a. データ ソースに割り当てる 1 つまたは複数のターゲットを [選択可] カラムで選択します。
   b. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
   c. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

以下の表に、WLEC 接続プールをコンフィグレーションおよび管理するために Administration Console で使用する属性を示します。詳細については、『WebLogic Enterprise Connectivity ユーザーズ ガイド』を参照してください。

[一般]

属性	説明	値の範囲	デフォルト値
[名前]	WLEC 接続プールの名前を返す。各 WLEC 接続プールについて、ユニークな名前でなければならない。	名前には英数字で 256 文字まで使用できるが、カンマまたはスペースは不可	NULL

属性	説明	値の範囲	デフォルト値

[プライマリ アドレス]	WLEC 接続プールと WLEC ドメインの間で接続を確立する IIOP リスナ / ハンドラのためのアドレスのリストを返す。各アドレスのフォーマットは、*//hostname:port*。
	これらのアドレスは、UBBCONFIG で定義された ISL アドレスと一致している必要がある。複数のアドレスは、セミコロンで区切る。例：
	`//main1.com:1024; //main2.com:1044`
	SSL プロトコルを使用するように WLEC 接続プールをコンフィグレーションするには、IIOP リスナ / ハンドラのアドレスと一緒に、corbalocs プレフィックスを使用する。例：*corbalocs://hostname:port*
		文字列	NULL

[フェイルオーバー アドレス]	[プライマリ アドレス] フィールドで定義されたアドレスで接続が確立できない場合に使用される、IIOP リスナ / ハンドラのためのアドレスのリストを返す。複数のアドレスは、セミコロンで区切る。	文字列	NULL

属性	説明	値の範囲	デフォルト値
[ドメイン]	WLEC 接続プールの接続先となる WLEC ドメインの名前を返す。1 つの WLE ドメインにつき 1 つの WLEC 接続しか確立できない。ドメイン名は、WLE ドメインの UBBCONFIG ファイルで RESOURCES セクションにある、domainid パラメータと一致している必要がある。	文字列	NULL
[最小プール サイズ]	WebLogic Server の起動時に WLEC 接続プールに付加されるべき IIOP 接続数を返す。	整数	1
[最大プール サイズ]	WLEC 接続プールから可能な IIOP 接続の最大数を返す。	整数	1

[セキュリティ]

属性	説明	値の範囲	デフォルト値
[ユーザ名]	有効な資格を備えたユーザの名前を返す。このフィールドが必要となるのは、WLE ドメインのセキュリティ レベルが USER_AUTH、ACL または MANDATORY_ACL の場合のみ。	管理者が割り当てた有効なユーザ名	NULL
[ユーザ パスワード]	[ユーザ名] フィールドに指定された有効なユーザのパスワードを返す。このフィールドが必要となるのは、[ユーザ名] フィールドを定義する場合のみ。	管理者が割り当てた有効なユーザ パスワード	NULL
[ユーザ ロール]	この接続プールのユーザ ロールを設定する。このフィールドが必要となるのは、WLE ドメインのセキュリティ レベルが APP_PW、USER_AUTH、ACL または MANDATORY_ACL の場合のみ。	管理者または開発者が定義した有効なユーザ ロール	NULL

属性	説明	値の範囲	デフォルト値
[アプリケーション パスワード]	アプリケーションのパスワードを返す。このフィールドが必要となるのは、WLE ドメインのセキュリティ レベルが APP_PW、USER_AUTH、ACL または MANDATORY_ACL の場合のみ。	文字列	NULL
[最小暗号化レベル]	WLE ドメインと WebLogic Server の間で使用される SSL 暗号化の最低レベルを設定する。ゼロ（0）は、データを署名するが暗号化しないことを示す。40、56、128 は暗号化キーの長さをビットで指定する。この暗号化最小レベルに達していない場合、WLE と WebLogic Server の間の SSL 接続は失敗する。	指定できる値は、0、40、56、128。	40

属性	説明	値の範囲	デフォルト値
[最大暗号化レベル]	WLE ドメインと WebLogic Server の間で使用される SSL 暗号化の最高レベルを設定する。ゼロ（0）は、データを署名するが暗号化しないことを示す。40、56、128 は暗号化キーの長さをビットで指定する。この暗号化最小レベルに達していない場合、WLE と WebLogic Server の間の SSL 接続は失敗する。	指定できる値は、0、40、56、128。	0
[証明書を有効化]	証明書を有効にする。 証明書を有効にすると、WLEC は [ユーザ名] および [アプリケーション パスワード] の各フィールドの値を使って、WLEC の証明書を作成する。 証明書を有効にしない場合、WLEC は WLE ドメインのセキュリティ レベルに応じて、パスワードによる認証を使用するか、認証を行わない。パスワード認証が必要な場合、WLEC は [ユーザ名] および [ユーザ パスワード] フィールドの値を認証に使用する。	ブール 選択されている = 有効 選択されていない = 無効	選択されていない

属性	説明	値の範囲	デフォルト値
[セキュリティ コンテキストを有効化]	WLE ドメインに渡された WebLogic Server ユーザのセキュリティ コンテキストの状態を指定する。	ブール 選択されている = 有効 選択されていない = 無効	選択されていない

[対象]

[サーバ]

属性	説明	値の範囲	デフォルト値
[対象サーバ]	割り当てられたサーバを選択するためのリストを提供する。	リスト	NULL

[クラスタ]

属性	説明	値の範囲	デフォルト値
[対象クラスタ]	この属性は、割り当てられたクラスタを選択するためのリストを提供する。	リスト	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

詳細については、『WebLogic Enterprise Connectivity ユーザーズ ガイド』を参照してください。

116 WLEC 接続

デフォルトでは、このペインを使用すると、ユーザが次の基準でオブジェクトを並べ替えることができます。

- 名前
- ドメイン
- プライマリ アドレス
- フェイルオーバ アドレス
- 最小プール サイズ
- 最大プール サイズ

117 XML エンティティ定義

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[パブリック ID]	この属性を使用すると、解決するエンティティ、またはドキュメントを解析するための文書型定義（DTD）ファイルの公開 ID を定義できる。		MYXMLEntrySpecRegistryEntry
[システム ID]	解決するエンティティ、またはドキュメントを解析するための文書型定義（DTD）ファイルのシステム ID を定義できる。		NULL
[エンティティ URI]	この属性は、外部エンティティを含むファイルのパスを定義するために使用する。このパスは、レジストリのエンティティ ディレクトリを基準とする相対パス。		NULL

属性	説明	値の範囲	デフォルト値
[キャッシュのタイミング]	URL で参照された外部エンティティを、WebLogic Server がキャッシュするタイミングを指定する。エンティティが XML 文書内で最初に参照された時、Weblogic Server が起動した時、またはグローバルなキャッシュの設定を使用する、のいずれかを指定する。	`cache-on-reference`, `cache-at-initialization`, `defer-to-registry-setting`	defer-to-registry-setting
[キャッシュ タイムアウト間隔]	WebLogic Server が外部エンティティをキャッシュしてから、その有効期限が切れるまでの秒数。	整数	-1

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

118 XML レジストリ

XML レジストリの作成

1. 左ペインの [XML] ノードをクリックします。右ペインに [XML レジストリ] テーブルが表示され、ドメインで定義されているすべての XML レジストリが示されます。
2. [新しい XML Registry のコンフィグレーション] テキスト リンクをクリックします。右ペインにダイアログが表示され、新しく作成するレジストリの設定に関連するタブが示されます。
3. [名前]、[Document Builder ファクトリ] および [SAX パーサ ファクトリ] 属性フィールドに値を入力します。
4. [作成] をクリックして、[名前] フィールドに指定した名前で XML レジストリのインスタンスを作成します。左ペインの [XML] ノードの下に、新しいレジストリが追加されます。

XML レジストリのクローンの作成

1. 左ペインの [XML] ノードをクリックします。右ペインに [XML レジストリ] テーブルが表示され、ドメインで定義されているすべての XML レジストリが示されます。
2. クローンを作成する レジストリの行で [クローン] アイコンをクリックします。右ペインにダイアログが表示され、新しいレジストリのクローンの作成に関連するタブが示されます。
3. [名前]、[Document Builder ファクトリ] および [SAX パーサ ファクトリ] 属性フィールドに値を入力します。

4. [作成] をクリックして、[名前] フィールドに指定した名前で XML レジストリのインスタンスを作成します。左ペインの [XML] ノードの下に、新しいレジストリが追加されます。

XML レジストリの削除

1. 左ペインの [XML] ノードをクリックします。右ペインに [XML レジストリ] テーブルが表示され、ドメインで定義されているすべての XML レジストリが示されます。
2. 削除するレジストリの行で [削除] アイコンをクリックします。削除要求の確認を求めるダイアログが右ペインに表示されます。
3. [はい] をクリックして、レジストリを削除します。[XML］ノードの下にあるレジストリ アイコンが削除されます。

XML レジストリの割り当て

1. 左ペインの [XML] の下で、割り当てる XLM レジストリのインスタンス ノードをクリックします。右ペインにダイアログが表示され、このインスタンスに関連するタブが示されます。
2. [対象] タブをクリックします。
3. [選択可] カラムで、レジストリに割り当てる 1 つまたは複数のターゲットを選択します。
4. 移動コントロールをクリックして、選択したターゲットを [選択済み] カラムに移動します。
5. [適用] をクリックして割り当てを保存します。

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[名前]	レジストリの名前を設定できる。	文字列	MyXML Registry
[Document Builder ファクトリ]	サーバのデフォルトのドキュメント ビルダ ファクトリを設定できる。 この属性では、DOM を使用してドキュメントを解析するすべてのサーバ アプリケーションのデフォルト ファクトリが指定される。特定の文書型定義（DTD）のデフォルト ファクトリをオーバライドするには、[XML レジストリ エントリ] ダイアログの [コンフィグレーション] タブにある [パーサ クラス名] 属性を使用する。	文字列	weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl

属性	説明	値の範囲	デフォルト値
[SAX パーサ ファクトリ]	サーバのデフォルトの SAX パーサ ファクトリを設定できる。 この属性では、SAX を使用してドキュメントを解析するすべてのサーバ アプリケーションのデフォルト ファクトリが指定される。 特定の文書型定義（DTD）のデフォルト ファクトリをオーバライドするか、または独自に作成した SAX パーサを使用するには、[XML レジストリ エントリ] ダイアログの [コンフィグレーション] タブにある [パーサ クラス名] 属性を使用する。	文字列	weblogic.apache.xerces.jaxp.SAXParserFactoryImpl
[Transformer ファクトリ]	JAXP インタフェースを実装し、XML 文書を変換する Transformer ファクトリ クラス名。	文字列	`weblogic.apache.xalan.processor.TransformerFactoryImpl`
[キャッシュのタイミング]	URL で参照された外部エンティティを、WebLogic Server がキャッシュするタイミングを指定する。エンティティが XML 文書内で最初に参照された時、または Weblogic Server が起動した時のいずれかを指定する	cache-on-reference、 cache-at-initialization	`cache-on-reference`

[対象]

属性	説明	値の範囲	デフォルト値
[サーバ]	サーバ インスタンスのリストから選択する。複数のインスタンスを選択するには、〔Ctrl〕を押しながら目的のサーバを選択する。	リスト	未選択

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力用の領域を提供する。	文字列	NULL

119 XML レジストリ エントリ

[コンフィグレーション]

属性	説明	値の範囲	デフォルト値
[パブリック ID]	この属性を使用すると、解決するエンティティ、またはドキュメントを解析するための文書型定義（DTD）ファイルの公開 ID を定義できる。		NULL
[システム ID]	解決するエンティティ、またはドキュメントを解析するための文書型定義（DTD）ファイルのシステム ID を定義できる。		NULL
[ルート要素タグ]	解析するドキュメントのルートを定義するために使用する。		
[エンティティ パス]	この属性は、外部エンティティを含むファイルのパスを定義するために使用する。このパスは、レジストリのエンティティ ディレクトリを基準とする相対パス。		NULL

属性	説明	値の範囲	デフォルト値
[パーサ クラス名]	この属性を使用すると、[エンティティ パス] に定義した DTD を使用してドキュメントを解析するための DOM パーサまたは SAX パーサを指定できる。独自に作成したパーサを使用する場合、この属性にそのパーサを指定する。 特定の DTD（独自に作成したパーサの場合は特定のアプリケーション）の場合、この属性を設定すると、[XML レジストリ] ダイアログの [コンフィグレーション] タブにある [Document Builder ファクトリ] および [SAX パーサ ファクトリ] 属性で指定したデフォルトの DOM または SAX パーサがオーバライドされる。	有効なクラス名	NULL

[メモ]

属性	説明	値の範囲	デフォルト値
[メモ]	ユーザ入力（省略可）用の領域を提供する。	文字列	NULL

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

W

X

ア

イ

ウ

オ

カ

キ

ク

ケ

コ

サ

シ

ス

セ

ソ

タ

チ

ツ

テ

ト