



BEA WebLogic Server™

コンフィギュレーション
リファレンス

著作権

Copyright © 2003 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、複製、複製、翻訳、あるいはいかなる電子媒体または機械可読形式への変換も行うことはできません。

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

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

商標または登録商標

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

その他の商標はすべて、関係各社がその権利を有します。

パート番号	マニュアルの日付	ソフトウェアのバージョン
なし	2002年6月28日	BEA WebLogic Server バージョン 7.0

目次

このマニュアルの内容

対象読者	xxi
e-docs Web サイト	xxii
このマニュアルの印刷方法	xxii
関連情報	xxiii
サポート情報.....	xxiii
表記規則	xxiii

config.xml の概要

WebLogic Server の管理と config.xml ファイルについて.....	1-1
config.xml を編集する状況	1-2
config.xml ファイルの内容	1-3
config.xml ファイルの編集	1-4
BEA XML エディタ	1-4
config.xml ファイルの編集時の注意点.....	1-5

Application

説明	2-1
構文	2-1
親要素	2-2
子要素	2-2
属性	2-3

BridgeDestination

説明	3-1
構文	3-1
親要素	3-2
属性	3-3

Cluster

説明	4-1
構文	4-1
親要素	4-2
属性	4-3

COM

説明	5-1
構文	5-1
親要素	5-2
属性	5-2

ConnectorComponent

説明	6-1
構文	6-1
親要素	6-1
属性	6-2

CustomRealm

説明	7-1
構文	7-1
親要素	7-1
属性	7-2

Domain

説明	8-1
構文	8-1
子要素	8-2
属性	8-4

DomainLogFilter

説明	9-1
構文	9-1
親要素	9-2
属性	9-2

EJBComponent

説明	10-1
構文	10-1
親要素	10-2
属性	10-2

EJBContainer

説明	11-1
構文	11-1
親要素	11-2
属性	11-2

EmbeddedLDAP

説明	12-1
構文	12-1
親要素	12-2
属性	12-2

ExecuteQueue

説明	13-1
構文	13-1
親要素	13-1
属性	13-2

FileRealm

説明	14-1
構文	14-1
親要素	14-1
属性	14-2

FileT3

説明	15-1
構文	15-1
親要素	15-1
属性	15-2

IIOP

説明	16-1
構文	16-1
親要素	16-1
属性	16-2

JDBCConnectionPool

説明	17-1
構文	17-1
親要素	17-2
属性	17-2

JDBCDataSource

説明	18-1
構文	18-1
親要素	18-2
属性	18-3

JDBCDataSourceFactory

説明	19-1
構文	19-1
親要素	19-2
属性	19-2

JDBCMultiPool

説明	20-1
構文	20-1
親要素	20-2
属性	20-3

JDBCPoolComponent

説明	21-1
構文	21-1
親要素	21-1
属性	21-2

JDBCTxDataSource

説明	22-1
構文	22-1
親要素	22-2
属性	22-3

JMSBridgeDestination

説明23-1
構文23-1
親要素23-2
属性23-2

JMSConnectionFactory

説明24-1
構文24-1
親要素24-2
属性24-3

JMSDestinationKey

説明25-1
構文25-1
親要素25-1
属性25-2

JMSDistributedQueue

説明26-1
構文26-1
親要素26-1
子要素26-2
属性26-2

JMSDistributedTopic

説明27-1
構文27-1
親要素27-1
子要素27-2

属性	27-2
----------	------

JMSFileStore

説明	28-1
構文	28-1
親要素	28-1
属性	28-2

JMSJDBCStore

説明	29-1
構文	29-1
親要素	29-1
属性	29-2

JMSQueue

説明	30-1
構文	30-1
親要素	30-2
属性	30-3

JMSServer

説明	31-1
構文	31-1
親要素	31-2
子要素	31-2
属性	31-3

JMSSessionPool

説明	32-1
構文	32-1
親要素	32-1

属性32-2
----------	-------

JMSStore

説明33-1
構文33-1
親要素33-1
属性33-2

JMSTemplate

説明34-1
構文34-1
親要素34-2
属性34-3

JMSTopic

説明35-1
構文35-1
親要素35-2
属性35-3

JoltConnectionPool

説明36-1
構文36-1
親要素36-2
属性36-2

JTA

説明37-1
構文37-1
親要素37-2
属性37-3

JTAMigratableTarget

説明	38-1
構文	38-1
親要素	38-1
属性	38-2

JTARecoveryService

説明	39-1
構文	39-1
親要素	39-1
属性	39-2

Log

説明	40-1
構文	40-1
親要素	40-2
属性	40-2

Machine

説明	41-1
構文	41-1
親要素	41-2
子要素	41-2
属性	41-2

MailSession

説明	42-1
構文	42-1
親要素	42-1
属性	42-2

MessagingBridge

説明	43-1
構文	43-1
親要素	43-2
属性	43-3

MigratableRMIService

説明	44-1
構文	44-1
親要素	44-1
属性	44-2

MigratableTarget

説明	45-1
構文	45-1
親要素	45-1
属性	45-2

NetworkChannel

説明	46-1
構文	46-2
親要素	46-2
属性	46-3

NodeManager

説明	47-1
構文	47-1
親要素	47-2
属性	47-2

Realm

説明	48-1
構文	48-1
親要素	48-1
属性	48-2

RMCFactory

説明	49-1
構文	49-1
親要素	49-1
属性	49-2

Security

説明	50-1
構文	50-1
親要素	50-2
属性	50-2

SecurityConfiguration

説明	51-1
構文	51-1
親要素	51-1
属性	51-2

Server

説明	52-1
構文	52-1
親要素	52-3
子要素	52-3
属性	52-5

ServerStart

説明53-1
構文53-1
親要素53-2
属性53-2

ShutdownClass

説明54-1
構文54-1
親要素54-2
属性54-2

SNMPAgent

説明55-1
構文55-1
親要素55-2
属性55-2

SNMPAttributeChange

説明56-1
構文56-1
親要素56-1
属性56-2

SNMPJMXMonitor

説明57-1
構文57-1
親要素57-2
属性57-2

SNMPProxy

説明	58-1
構文	58-1
親要素	58-1
属性	58-2

SNMPTrapDestination

説明	59-1
構文	59-1
親要素	59-1
属性	59-2

SNMPTrapSource

説明	60-1
構文	60-1
親要素	60-1
属性	60-2

SSL

説明	61-1
構文	61-1
親要素	61-2
属性	61-3

StartupClass

説明	62-1
構文	62-1
親要素	62-2
属性	62-2

UnixMachine

説明63-1
構文63-1
親要素63-2
子要素63-2
属性63-2

UnixRealm

説明64-1
構文64-1
親要素64-1
属性64-2

VirtualHost

説明65-1
構文65-1
親要素65-2
属性65-3

WebAppComponent

説明66-1
構文66-1
親要素66-2
属性66-2

WebServer

説明67-1
構文67-1
親要素67-2
属性67-3

WebServiceComponent

説明	68-1
構文	68-1
親要素	68-2
属性	68-2

WLECConnectionPool

説明	69-1
構文	69-1
親要素	69-2
属性	69-3

WTCExport

説明	70-1
構文	70-1
親要素	70-2
属性	70-2

WTCTImport

説明	71-1
構文	71-1
親要素	71-2
属性	71-2

WTCLocalTuxDom

説明	72-1
構文	72-1
親要素	72-2
属性	72-2

WTCPassword

説明	73-1
構文	73-1
親要素	73-2
属性	73-2

WTCRemoteTuxDom

説明	74-1
構文	74-1
親要素	74-2
属性	74-2

WTCResources

説明	75-1
構文	75-1
親要素	75-2
属性	75-2

WTCServer

説明	76-1
構文	76-1
親要素	76-1
子要素	76-2
属性	76-3

WTCtBridgeGlobal

説明	77-1
構文	77-1
親要素	77-2
属性	77-3

WTCtBridgeRedirect

説明	78-1
構文	78-1
親要素	78-2
属性	78-2

XMLEntityCache

説明	79-1
構文	79-1
親要素	79-1
属性	79-2

XMLEntitySpecRegistryEntry

説明	80-1
構文	80-1
親要素	80-2
属性	80-2

XMLParserSelectRegistryEntry

説明	81-1
構文	81-1
親要素	81-2
属性	81-2

XMLRegistry

説明	82-1
構文	82-1
親要素	82-1
子要素	82-2
属性	82-2

このマニュアルの内容

このマニュアルでは、WebLogic Server コンフィグレーション ファイル (config.xml) 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。

- 第 1 章「config.xml の概要」では、config.xml ファイルと config.dtd ファイルについて説明し、コンフィグレーション ファイルの編集ガイドラインについて説明します。
- 残りの章では、config.dtd で定義されている各要素について、構文の構造図と属性のリファレンスを示します。
- 索引ではすべての要素名と属性名へのリンクを示します。

対象読者

これは、サーバ管理者とアプリケーション開発者向けのマニュアルです。WebLogic Server プラットフォーム、XML、および Java プログラミング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。管理者の中には、プロダクション環境で WebLogic Server の処理の一部をコンフィグレーションすることを考えている方もいます。コンフィグレーションタスクは、Web ブラウザ GUI アプリケーションである Administration Console でも行えます。ただし、場合によっては、コンフィグレーション ファイル config.xml を直接変更す

このマニュアルの内容

の方が適切です。このマニュアルでは、`config.xml` ファイルのプロダクションインスタンスに存在する、または存在する可能性のある XML 要素とそれに関連する属性について説明します。

注意： `config.xml` ファイルを変更すると、お客様の環境にある WebLogic Server の運用に影響を与えます。

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。WebLogic Server BEA のホームページで [製品のドキュメント] をクリックします。

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 章ずつ印刷できます。

このマニュアルの PDF 版は、WebLogic Server の 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

サポート情報

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

電子メールのメッセージには、ご使用の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 (www.bea.com) を通じて BEA カスタマ サポートまでお問い合わせください。カスタマ サポートへの連絡方法については、製品パッケージに同梱されているカスタマ サポート カードにも記載されています。

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

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

表記規則

このマニュアルでは、全体を通して以下の表記規則が使用されています。

表記法	適用
[Ctrl] + [Tab]	複数のキーを同時に押すことを示す。
斜体	強調または書籍のタイトルを示す。
等幅テキスト	コード サンプル、コマンドとそのオプション、データ構造体とそのメンバー、データ型、ディレクトリ、およびファイル名とその拡張子を示す。等幅テキストはキーボードから入力するテキストも示す。 例： <pre>import java.util.Enumeration; chmod u+w * config/examples/applications .java config.xml float</pre>
斜体の等幅テキスト	コード内の変数を示す。 例： <pre>String <i>CustomerName</i>;</pre>
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 例： <pre>LPT1 BEA_HOME</pre> または
{ }	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： <pre>java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]</pre>

表記法	適用
	構文の中で相互に排他的な選択肢を区切る。 例： <pre>java weblogic.deploy [list deploy undeploy update] password {application} {source}</pre>
...	コマンドラインで以下のいずれかを示す。 <ul style="list-style-type: none">• 引数を複数回繰り返すことができる。• 任意指定の引数が省略されている。• パラメータや値などの情報を追加入力できる。
.	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。 . .

このマニュアルの内容

config.xml の概要

以降の節では、WebLogic Server™ コンフィグレーション ファイル (config.xml) の概要について説明します。

- 1-1 ページの「WebLogic Server の管理と config.xml ファイルについて」
- 1-4 ページの「config.xml ファイルの編集」
- 1-5 ページの「config.xml ファイルの編集時の注意点」

WebLogic Server の管理と config.xml ファイルについて

WebLogic Server の管理とコンフィグレーションのサービスは、Sun Microsystems の Java™ Management Extensions (JMX) API に基づきます。config.xml ファイルは、WebLogic Server がその実行時に JMX API の BEA 実装を使用して作成および修正する管理対象オブジェクトの永続性ストアです。config.xml の目的は、WebLogic Server が再起動されたときに利用できるよう管理対象オブジェクトに対する変更を格納することです。

通常は、Administration Console を使用して WebLogic Server の管理可能オブジェクトおよびサービスをコンフィグレーションし、WebLogic Server で config.xml ファイルを保持できるようにします。Administration Console などの WebLogic Server ユーティリティを使用して config.xml ファイルを修正するたびに、古いバージョンが WebLogic Server によっ

てアーカイブに配置されます。**WebLogic Server** で保持されるアーカイブ ファイルの数をコンフィグレーションできます。詳細については、『**WebLogic Server のコンフィグレーションと管理**』の「前バージョンの config.xml のアーカイブ」を参照してください。

config.xml はテキスト エディタで修正できる整形式の XML ドキュメントですが、このファイルのことは、特殊な環境でのみ直接更新するデータベースであると考えべきです。

config.xml ファイルは、その内容を検証するための文書型定義 (DTD) を持つ正式な XML ドキュメントではありません。正確にはリポジトリであり、各 XML 要素に、**WebLogic Server** の管理オブジェクトのインメモリ オブジェクトが要求されたときに保存されたデータが格納されます。ただし、config.xml にデータを格納するオブジェクトは **WebLogic Server Management API** の範囲内で定義されるので、config.xml ファイルの内容は実質的に文書化することが可能です。実際、このマニュアルのリファレンス情報は、**WebLogic Server 管理 MBean** の含まれる weblogic.management.configuration パッケージの Javadoc から抽出されています。Javadoc と Administration Console オンライン ヘルプ システムでは、同じ情報を異なる表示で参照できます。

config.xml には、このマニュアルで説明されていないデータを格納できる場合があります。

- 公開される予定がなく、将来のリリースでもサポートされない内部のメカニズムに関連するために、一部の MBean 属性は意図的に説明されていない。
- MBean と属性は **WebLogic Server** で動作するアプリケーションで動的に作成できる。そのようなアプリケーションには、Administration Console、**WebLogic Server** と関係して機能する他の BEA 製品、およびユーザまたはサードパーティ ベンダ提供のアプリケーションがあります。

他の BEA 製品を **WebLogic Server** またはサードパーティ アプリケーションと一緒に使用する場合は、その製品に付属のドキュメントで config.xml 要素および属性に関する追加情報を参照してください。

config.xml を編集する状況

Administration Console を使用せず、config.xml ファイルを直に編集する状況はいくつかあります。そのような状況では、変更は特定のかつ最小限である必要があります。

config.xml ファイルを新たに作成して新しい **WebLogic Server** コンフィグレーションを作成することはしないでください。

警告： WebLogic Server は config.xml を定期的書き換えるので、WebLogic Server の実行中は編集できません。行った変更は失われ、プラットフォームによっては、WebLogic Server に障害が発生します。

config.xml ファイルを修正する前には、必ずバックアップ コピーを作成してください。

以下の状況では、config.xml を編集するのが適切です。

- 複数の WebLogic Server インスタンスをデプロイする場合は、config.xml ファイルのクローンを作成して、新しいサーバごとにその値を編集できる
- 1 つのサーバで定義されているオブジェクトを別のサーバにコピーしたい場合は、そのオブジェクトを定義する XML 要素を config.xml ファイル間でコピーできる
- 問題の解決を支援するため、BEA テクニカル サポートでは Administration Console で設定できない属性の設定を提案する場合がある
- サードパーティ アプリケーションのドキュメントでは、config.xml ファイルの修正が要求されている

config.xml は整形形式の XML ファイルなので、Apache Xerces や JDOM などの XML パーサアプリケーションを使用して、config.xml に対する特定の反復的な変更をスクリプト化することができます。作成したスクリプトは必ず徹底的にテストし、変更の前には必ず config.xml のバックアップ コピーを作成してください。

config.xml ファイルの内容

config.xml ファイルは XML 要素群で構成されています。Domain 要素はトップレベルの要素であり、Domain 内のすべての要素は Domain 要素の子です。Domain 要素には、Server 要素、Cluster 要素、Application 要素などの子要素があります。子要素の中にさらに子要素がある場合もあります。

たとえば、Server 要素には、子要素として WebServer、SSL、および Log があります。Application 要素には EJBComponent と WebAppComponent という子要素があります。

各要素には 1 つまたは複数のコンフィグレーションできる属性があります。属性には、それに対応する属性がコンフィグレーション API にあります。たとえば、Server 要素には ListenPort 属性があり、weblogic.management.configuration.ServerMBean クラスにも

ListenPort 属性があります。コンフィグレーション可能な属性は、読み書き可能です。つまり、**ServerMBean** には `getListenPort()` メソッドと `setListenPort()` メソッドがあります。

属性が `config.xml` の別の要素 (**Server** の **SSL** 属性など) を参照する場合、その属性の値は参照される要素の **Name** 属性の値です (すべての `config.xml` 要素に **Name** 属性がある)。属性が他の要素の集合またはリストを参照する場合、その属性の値は参照される要素の名前のカンマ区切りのリストです。

属性値はすべて引用符で囲む必要があります。ブール属性の値は、"true" または "false" です。数値の属性は、指定されたデータ型に解析できる、引用符で囲まれた一続きの数字および記号として記述されます。たとえば、整数値にカンマや小数点は含まれてはなりません。属性で負の値が許可される場合は先頭にマイナス記号を置くことができます。

config.xml ファイルの編集

ドメイン コンフィグレーションの変更やモニタには、**Administration Console** を使用することをお勧めします。また、次の方法でも `config.xml` ファイルを変更できます。

- **WebLogic Server API (Application Programmatic Interface)** を使って、他のプログラムからドメイン内のリソースのコンフィグレーション属性を変更する。
- **WebLogic Server** コマンドラインユーティリティを使ってドメインの属性にアクセスする。このユーティリティでは、ドメイン管理を自動化するスクリプトを作成できます。

BEA XML エディタ

BEA XML エディタは、XML ファイルを作成および編集するためのシンプルでユーザフレンドリなツールです。**BEA XML エディタ**では、**DTD** または **XML スキーマ**と照らし合わせて **XML** ドキュメントを検証できます。`config.xml` に **DTD** または **XML スキーマ**はありませんが、**XML エディタ**を使用すれば、少なくとも、**WebLogic Server** で解析できる整形形式の **XML** ドキュメントを作成することはできます。ただし、要素と属性の名前の綴りが適切かどうか、属性の値が有効かどうかを検証することはできません。

BEA dev2dev Online の「**BEA XML Editor**」を参照してください。

config.xml ファイルの編集時の注意点

コンフィグレーション ファイルを手動で編集する場合、あらかじめ次の点に注意してください。

1. config.xml ファイルを編集するときは、常に編集前に保存しておきます。
2. ドメインがアクティブなときには、ドメイン用の config.xml ファイルを編集しないでください。ドメインがアクティブなときにコンフィグレーション ファイルを手動で変更した場合、行った変更がシステムによって上書きされるおそれがあります。さらに、ドメインがアクティブなときに手動で行ったすべての変更は、システムの実行時には無視されます。
3. 子要素の Name 属性を変更しないでください。WebLogic Server では、子要素の名前は親要素の名前と一致している必要があります。たとえば、Server 要素の Name 属性の値が myserver の場合、Server 要素の子要素である COM、JTAMigratableTarget、KernalDebug、Log、SSL、ServerStart、SystemDataStore、および WebServer 要素の Name 属性の値も同じ myserver である必要があります。
4. コマンドライン ユーティリティで config.xml を編集しているとき、またはその編集後には検証や値のチェックが行われないので、編集したコンフィグレーション ファイルを初めてロードしたとき、つまり、ドメインのサーバを再起動したときに型チェックが行われます。その時点で無効な XML または無効な属性値が検出された場合、ドメインは起動できません。

Application

説明

アプリケーションは、EAR ファイルまたは EAR ディレクトリに格納される J2EE アプリケーションを表します。EAR ファイルは、WAR、EJB、および RAR コネクタ コンポーネントなどの一連のコンポーネントで構成され、各コンポーネントは 1 つまたは複数の対象にデプロイすることができます。対象はサーバまたはクラスタです。アプリケーションがスタンドアロン モジュールとして提供される場合、この MBean は単なる合成ラッパー アプリケーションです。

構文

```
<Application
  DeploymentTimeout=number"
  DeploymentType=( "EAR" | "EXPLODED EAR" | "COMPONENT" | "EXPLODED COMPONENT" )
  LoadOrder=number"
  Name=String"
  Notes=String"
  Path=String"
  StagingMode=( "nostage" | "stage" | "external_stage" )
/>
```

親要素

- Domain

子要素

- ConnectorComponent
- EJBComponent
- WebServiceComponent
- WebAppComponent
- JDBCPoolComponent

属性

表 2-1 Application の属性

属性	説明	指定できる値とデフォルト値
DeploymentTimeout	このアプリケーションの、クラスタに対するデプロイメントタスクに付与された時間(ミリ秒単位)。この時間を経過してもデプロイメントタスクがアクティブな状態である場合は、タスクがキャンセルされる。デフォルトの時間は 60 分。サーバがデプロイメントのタイムアウトをチェックする頻度は、約 1 分ごと。クラスタに対するデプロイメントだけがタイムアウトになる。	必須: いいえ デフォルト値: 3600000
DeploymentType	このアプリケーションのカテゴリを指定する。コンフィグレーションで指定されない場合、この属性は継承される。	必須: いいえ デフォルト値: UNKNOWN
LoadOrder	サーバの起動時にアプリケーションがロードされる順序を指定する。値が小さいアプリケーションから順にロードされる。 アプリケーションの順序は、2 フェーズプロトコルを使用してデプロイされるアプリケーションに対してのみサポートされる。	必須: いいえ デフォルト値: 100
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ

表 2-1 Application の属性 (続き)

属性	説明	指定できる値とデフォルト値
Path	<p>管理サーバ上にある、元のソース アプリケーション ファイルの場所。相対パスは、管理サーバをインストールしたディレクトリのルートを基準にする。サーバのアップグレード時に発生する問題を最小限に抑えるため、絶対パスの使用を推奨。</p> <p>アプリケーションがステージングされていない場合 (StagingMode が nostage の場合)、パスは対象サーバ上で有効なものでなければならない。</p> <p>エンタープライズアプリケーション (EAR) へのパスは、.ear ファイルの場所、またはアーカイブされていない場合は EAR のルート。たとえば、パス「myapps/app.ear」は有効。アプリケーションがスタンドアロン モジュールの場合、パスはモジュールの親ディレクトリになる。たとえば、モジュールが myapps/webapp/webapp.war にある場合、パス「myapps/webapp」は正しいが、パス「myapps/webapp/webapp.war」は不適切。</p>	Administration Console のフィールド表示 : [パス]

表 2-1 Application の属性 (続き)

属性	説明	指定できる値とデフォルト値
StagingMode	<p>このアプリケーションのステージングモードを返す。このモードは、管理対象サーバの StagingMode 属性をオーバーライドする。ステージングでは、アプリケーション ファイルが管理サーバから目的の管理対象サーバのステージングディレクトリに配布される。ステージング値は以下のとおり。</p> <ol style="list-style-type: none"> 1. nostage: アプリケーション ファイルは別の場所にコピーされない。 2. stage: アプリケーション ファイルがデプロイメントで目標とされたサーバにコピーされる。 3. external_stage: WebLogic Server ではなくユーザが、アプリケーション ファイルをサーバのステージングディレクトリにコピーする。 	<p>Administration Console のフィールド表示 : [ステージングモード]</p> <p>必須 : いいえ</p>

BridgeDestination

説明

このクラスは、JMS に対応していない製品用のメッセージングブリッジ送り先を表します。

各メッセージングブリッジは、ブリッジングされている2つの送り先で構成されます。1つはメッセージの受信元になるソース送り先、もう1つはメッセージの送信先になる対象送り先です。

注意：WebLogic JMS には、JMS 以外のメッセージング製品にアクセスするために「一般ブリッジ送り先」というフレームワークが用意されていますが、JMS 以外のメッセージング製品をサポートするアダプタは用意されていません。したがって、サードパーティ 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アダプタをコンフィグレーションする必要があります。

構文

```
<BridgeDestination
  AdapterJNDIName="String"
  Classpath="String"
  Name="String"
  Notes="String"
```

BridgeDestination

```
Properties=" java.util.Properties"  
UserName="String"  
UserPassword="String"  
</>
```

親要素

- Domain

属性

表 3-1 BridgeDestination の属性

属性	説明	指定できる値とデフォルト値
AdapterJNDIName	指定した送り先との通信に使用されるアダプタの JNDI 名を定義する。この名前は、アダプタのデプロイメント記述子ファイルで指定され、WebLogic Server JNDI でアダプタをバインドする WebLogic Server コネクタ コンテナによって使用される。	Administration Console のフィールド表示: [JNDI アダプタ名] デフォルト値: eis.jms.WLSConnectionFactoryNDIXA
Classpath	ブリッジ送り先の CLASSPATH を定義する。主に、WebLogic JMS の別リリースへの接続に使用される。 接続する送り先がバージョン 6.0 以前の WebLogic Server で動作している場合、古い WebLogic Server 実装のクラスの場合を示す CLASSPATH をブリッジ送り先に指定する必要がある。 <i>注意:</i> サードパーティの JMS 製品に接続する場合は、WebLogic Server の CLASSPATH で製品の CLASSPATH をブリッジ送り先に指定する必要がある。	Administration Console のフィールド表示: [アダプタ クラスパス] 必須: いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ

表 3-1 BridgeDestination の属性

属性	説明	指定できる値とデフォルト値
Properties	<p>ブリッジ送り先のプロパティを指定する文字列値。送り先のプロパティは、セミコロン (;) で区切る必要がある。</p> <p>以下のプロパティは、すべての JMS 実装で必要となる。</p> <p>ConnectionURL= 送り先への接続の確立に使用される URL。</p> <p>ConnectionFactoryJNDIName= 接続の作成に使用される JMS 接続ファクトリの JNDI 名。</p> <p>DestinationJNDIName= JMS 送り先の JNDI 名。</p> <p>DestinationType= キューまたはトピックのいずれか。</p> <p>InitialContextFactory= JNDI コンテキストの取得に使用されるファクトリ。</p>	<p>Administration Console のフィールド表示: [プロパティ (key=value)]</p> <p>必須: いいえ</p>

表 3-1 BridgeDestination の属性

属性	説明	指定できる値とデフォルト値
UserName	<p>ブリッジ送り先にアクセスする場合にアダプタで 사용되는ユーザ名を定義する (省略可能)。</p> <p>注意：指定した送り先に対する処理はすべて、このユーザ名と、対応するパスワードを使って行われる。したがって、メッセージングブリッジを機能させるには、ソースおよび対象送り先のユーザ名 / パスワードに基底の送り先へのアクセス権限が付与されている必要がある。</p>	<p>Administration Console のフィールド表示：[ユーザ名]</p> <p>必須：いいえ</p>
UserPassword	<p>ブリッジ送り先にアクセスする場合にアダプタで 사용되는ユーザ パスワードを定義する。</p>	<p>Administration Console のフィールド表示：[ユーザ パスワード]</p> <p>必須：いいえ</p> <p>暗号化：はい</p>

BridgeDestination

Cluster

説明

この Bean は、ドメイン内のクラスタを表します。サーバは、クラスタの論理名を使って `ServerMBean.setCluster` を呼び出すことによって、クラスタに参加します。コンフィグレーションでは、ゼロまたは1つ以上のクラスタを定義できます。クラスタは、論理名によってロックアップできます。クラスタの名前は、その論理クラスタ名を表します。

構文

```
<Cluster
  ClientCertProxyEnabled=( "true" | "false" )
  ClusterAddress="String"
  DefaultLoadAlgorithm=( "round-robin" | "weight-based" | "random" )
  HttpTraceSupportEnabled=( "true" | "false" )
  IdlePeriodsUntilTimeout="number"
  MulticastAddress="String"
  MulticastBufferSize="number of kilobytes"
  MulticastPort="number"
  MulticastSendDelay="number"
  MulticastTTL="number"
  Name="String"
  Notes="String"
```

Cluster

```
    WeblogicPluginEnabled=( "true" | "false" )  
/>
```

親要素

- Domain

属性

表 4-1 Cluster の属性

属性	説明	指定できる値とデフォルト値
ClientCertProxyEnabled	<p>値を true に設定すると、クライアントからの ID 証明書が、プロキシサーバプラグインによって、クラスタのすべてのサーバインスタンスにデプロイされているすべての Web アプリケーションに渡される。</p> <p>プロキシサーバプラグインは、WL-Proxy-Client-Cert ヘッダの各 ID 証明書をエンコードし、そのヘッダを WebLogic Server インスタンスに渡す。各 WebLogic Server インスタンスは、それがセキュアなソースから来ているものと信頼してヘッダから証明書情報を取得し、その情報を使用してユーザを認証する。</p> <p>true を指定した場合は、weblogic.security.net.ConnectionFilter を使用して、各 WebLogic Server インスタンスが、プロキシサーバプラグインが動作しているマシンからの接続のみを受け入れるようにする。接続フィルタを使用しないで true を指定すると、WL-Proxy-Client-Cert ヘッダのなりすましが可能なためにセキュリティが脆弱になる。</p> <p>値 true は、クラスタ内の各サーバインスタンスが ServerMBean#setClientCertProxyEnabled(boolean) で指定した値をオーバーライドする。デフォルトの場合 (または false を指定した場合)</p> <ul style="list-style-type: none"> 各サーバインスタンスでは、そのアプリケーションがプロキシサーバプラグインから送信された証明書を信頼するかどうかを指定できる サーバインスタンスでその ClientCertProxyEnabled 属性の値を設定しない場合 (または false が指定された場合) は、各 Web アプリケーションの weblogic.xml デプロイメント記述子で、その Web アプリケーションがプロキシサーバプラグインから送信された証明書を信頼するかどうかを指定する デフォルトでは (またはデプロイメント記述子で false が指定された場合)、ユーザはプロキシサーバプラグインから Web アプリケーションにログインできない 	<p>Administration Console のフィールド表示 : [クライアント証明書プロキシを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p> <p>セキュアな値 : false</p>

表 4-1 Cluster の属性 (続き)

属性	説明	指定できる値とデフォルト値
ClusterAddress	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定義する。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。ネットワーク チャンネルがコンフィグレーションされている場合は、クラスタ アドレスはチャンネルごとに設定できる。	Administration Console のフィールド表示 : [クラスタ アドレス] 必須 : いいえ
DefaultLoadAlgorithm	特定のサービスに対して何も指定されていない場合、レプリケートされるサービス間のロード バランシングに使用されるアルゴリズムを定義する。	Administration Console のフィールド表示 : [デフォルトのロード バランス アルゴリズム] 必須 : いいえ デフォルト値 : round-robin
HttpTraceSupportEnabled	攻撃者が HTTP TRACE トレース機能を悪用し、HTTP ヘッダ内の情報 (クッキー、認証データなど) にアクセスする可能性がある。Web ブラウザに他のクロス ドメイン脆弱性が存在する場合、HTTP TRACE メソッドをサポートするすべてのドメインから、機密性の高いヘッダ情報が読み取られるおそれがある。この属性は、HTTP TRACE のサポートを無効にするために使用する。 HttpTraceSupportEnabled 属性は、クラスタ全体で使用できるように ClusterMBean と ServerMBean の両方で複製される。ClusterMBean は ServerMBean をオーバーライドする。	必須 : いいえ デフォルト値 : false

表 4-1 Cluster の属性 (続き)

属性	説明	指定できる値とデフォルト値
IdlePeriodsUntilTimeout	クラスタ メンバーが、クラスタのメンバーをタイムアウトするまで待機する期間の最大数。	必須: いいえ デフォルト値: 3 最小値: 3
MulticastAddress	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 アドレスを定義する。	Administration Console のフィールド表示: [マルチキャスト アドレス] 必須: いいえ デフォルト値: 237.0.0.1
MulticastBufferSize	マルチキャスト ソケットの送信または受信バッファ サイズを定義する。	Administration Console のフィールド表示: [マルチキャスト バッファ サイズ] 必須: いいえ 単位: KB デフォルト値: 64 最小値: 64
MulticastPort	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 ポートを定義する。	Administration Console のフィールド表示: [マルチキャスト ポート] 必須: いいえ デフォルト値: 7001 最小値: 1 最大値: 65535
MulticastSendDelay	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送信が遅延されるミリ秒数を定義する。	Administration Console のフィールド表示: [マルチキャスト送信遅延] 必須: いいえ デフォルト値: 12 最小値: 0 最大値: 100

表 4-1 Cluster の属性 (続き)

属性	説明	指定できる値とデフォルト値
MulticastTTL	クラスタのマルチキャスト アドレスの存続時間値を設定する。	Administration Console のフィールド表示 : [マルチキャスト生存時間] 必須 : いいえ デフォルト値 : 1 最小値 : 1 最大値 : 255
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示 : [メモ] 必須 : いいえ
WeblogicPluginEnabled	WLS <code>HttpRequest.getRemoteAddr()</code> では、その戻り値で <code>X-Forwarded-For</code> を利用していた。これは、HTTP ヘッダが容易に模倣可能であり、最終的に不正な値が返されるのでセキュリティ ホールとなる。この問題は独自ヘッダ <code>WL-Proxy-Client-IP</code> を導入することで改善され、このヘッダは WLS がプラグインを使用するようにコンフィグレーションされている場合にのみ使用される。これは <code>ClusterMBean</code> と <code>ServerMBean</code> の両方で複製されるので、属性 <code>WeblogicPluginEnabled</code> はクラスタ全体で使用できる。 <code>ClusterMBean</code> は <code>ServerMBean</code> をオーバーライドする。	Administration Console のフィールド表示 : [WebLogic プラグインを有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : false

COM

説明

この Bean は、COM のサーバ全体のコンフィグレーションを表します。

構文

```
<COM
  ApartmentThreaded=( "true" | "false" )
  MemoryLoggingEnabled=( "true" | "false" )
  NTAAuthHost="String"
  Name="String"
  NativeModeEnabled=( "true" | "false" )
  Notes="String"
  PrefetchEnums=( "true" | "false" )
  VerboseLoggingEnabled=( "true" | "false" )
/>
```

親要素

- Server

属性

表 5-1 COM の属性

属性	説明	指定できる値とデフォルト値
ApartmentThreaded	COM をネイティブ モードで初期化するためのフラグを設定する。デフォルトでは、jCOM は <code>COINIT_MULTITHREADED_FLAG</code> を使用して COM を初期化する。ネイティブ モード時にサーバが「Class Not Registered Message」というログを記録する場合、このプロパティを設定する (<code>COINIT_APARTMENTTHREADED</code>)。	Administration Console のフィールド表示:[アパートメントスレッド] 必須:いいえ デフォルト値: false
MemoryLoggingEnabled	メモリ使用状況のロギングを有効にする。	Administration Console のフィールド表示:[メモリログを有効化] 必須:いいえ デフォルト値: false
NTAuthHost	クライアント認証用のプライマリ ドメイン コントローラのアドレス。このプロパティを設定しない場合、COM クライアントは認証されない。	Administration Console のフィールド表示:[NTAuth Host] 必須:いいえ

表 5-1 COM の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須 : いいえ
NativeModeEnabled	ネイティブ DLL を使うと、Java オブジェクトが COM オブジェクトと対話できる。Windows でのみサポートされる。	Administration Console の フィールド表示 : [ネイティブ モードを有効化] 必須 : いいえ デフォルト値 : false
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
PrefetchEnums	一部の COM メソッドは、COM VariantEnumeration 型を返す。 java2com ツールは、返された型を標準 Java java.lang.Enumeration に自動的に変換する。しかし、これは完全な一致とはならない。COM Enumeration には hasMoreElements() 呼び出しに相当するものがないからである。クライアントは、 NoSuchElementException が発生するまで nextElement を呼び出し続ける必要がある。このプロパティを設定すると、jCOM はバックグラウンドで次の要素をプリフェッチして、hasMoreElements が呼び出されたときに正しい値を返す。	Administration Console の フィールド表示 : [列挙子のプリフェッチ] 必須 : いいえ デフォルト値 : false
VerboseLoggingEnabled	冗長なロギングを有効にする。	必須 : いいえ デフォルト値 : false

ConnectorComponent

説明

この Bean はリソース アダプタを定義します。

構文

```
<ConnectorComponent
  DeploymentOrder="number"
  Name="String"
  Notes="String"
  Targets="list of Target names"
  URI="String"
/>
```

親要素

- Application

属性

表 6-1 ConnectorComponent の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>Administration Console のフィールド表示: [デプロイ順]</p> <p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: 2³¹-1</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須: いいえ</p>
URI	<p>アプリケーション コンポーネントを示す URI を設定する (通常は管理サーバ上にある)。</p>	<p>Administration Console のフィールド表示: [URI]</p>

CustomRealm

説明

構文

```
<CustomRealm
  ConfigurationData="java.util.Properties"
  Name="String"
  Notes="String"
  Password="String"
  RealmClassName="String"
/>
```

親要素

- Domain

属性

表 7-1 CustomRealm の属性

属性	説明	指定できる値とデフォルト値
ConfigurationData	カスタムセキュリティレルムのセキュリティストアに接続するために必要な情報。	Administration Console のフィールド表示: [コンフィグレーション情報 (key=value)] 必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes		Administration Console のフィールド表示: [メモ] 必須: いいえ
Password	カスタムセキュリティレルムのパスワード。指定したパスワードは、WebLogic Server によって暗号化される。	Administration Console のフィールド表示: [パスワード] 必須: いいえ 暗号化: はい セキュアな値: この値は <code>CustomRealmMBean</code> を初期化する場合は未定義でなければならない
RealmClassName	カスタムセキュリティレルムを実装する Java クラスの名前。	Administration Console のフィールド表示: [レルムクラス名] 必須: いいえ

Domain

説明

WebLogic ドメイン。

WebLogic ドメインは、「Domain」という特殊な対象です。

構文

```
<Domain
  AdministrationMBeanAuditingEnabled=( "true" | "false" )
  AdministrationPort=number
  AdministrationPortEnabled=( "true" | "false" )
  ConfigurationVersion=String
  ConsoleContextPath=String
  ConsoleEnabled=( "true" | "false" )
  Name=String
  Notes=String
/>
```

子要素

- Application
- BridgeDestination
- Cluster
- CustomRealm
- DomainLogFilter
- EJBContainer
- EmbeddedLDAP
- FileT3
- FileRealm
- JDBCDataSource
- JDBCDataSourceFactory
- JDBCMultiPool
- JDBCConnectionPool
- JDBCTxDataSource
- JMSServer
- JMSTemplate
- JMSBridgeDestination
- JMSConnectionFactory
- JMSDestinationKey
- JMSDistributedQueue
- JMSDistributedTopic
- JMSFileStore
- JMSJDBCStore

- JMSStore
- JMSJDBCStore
- JTA
- JoltConnectionPool
- Machine
- MessagingBridge
- MigratableTarget
- MigratableRMIService
- NetworkChannel
- MailSession
- Realm
- RMCFactory
- ShutdownClass
- SecurityConfiguration
- Security
- Server
- ShutdownClass
- SNMPAgent
- SNMPAttributeChange
- SNMPJMXMonitor
- SNMPTrapSource
- SNMPProxy
- SNMPTrapDestination
- StartupClass
- UnixMachine
- UnixRealm

- VirtualHost
- WLECConnectionPool
- WTCServer
- XMLEntityCache
- XMLRegistry

属性

表 8-1 Domain の属性

属性	説明	指定できる値とデフォルト値
AdministrationMBeanAuditingEnabled	AdministrationMBeanAuditingEnabled が true に設定されている場合、管理サーバがすべての管理 MBean 操作を記録する。	Administration Console のフィールド表示: [Configuration MBean Auditing Feature Enabled] 必須: いいえ デフォルト値: false
AdministrationPort	ドメインに対する共通のセキュアな管理ポート。管理ポートでは SSL を使用するため、ドメインで管理ポートが有効になっている場合、すべてのサーバで SSL を備える必要がある。管理ポートが有効である場合、同じドメイン内のすべてのサーバは、ドメインの管理ポートを使用して管理ポートを設定するか、またはサーバの管理ポートを使用して管理ポートをオーバーライドする必要がある。管理対象サーバは、管理サーバに接続するために <code>-Dweblogic.management.server=https://admin_server:administration_port</code> を使用しなければならない。	Administration Console のフィールド表示: [ドメイン管理ポート] 必須: いいえ デフォルト値: 9002 最小値: 1 最大値: 65534

表 8-1 Domain の属性 (続き)

属性	説明	指定できる値とデフォルト値
AdministrationPortEnabled	ドメインに対して管理ポートを有効にするかどうかを示す。これにより、ドメイン内のすべてのサーバで、サーバの管理ポートの設定が強制的に同じになる。	Administration Console のフィールド表示: [ドメイン管理ポートを有効化 (SSL をコンフィグレーションしてください)] 必須: いいえ デフォルト値: false セキュアな値: true
ConfigurationVersion	コンフィグレーションのリリース識別子。この識別子は、コンフィグレーションのバージョンを示すために使用する。サーバで生成されるすべてのコンフィグレーションは、実行中のサーバのリリース識別子を使って作成される。バージョンの形式は、 major.minor.servicepack.rollingpatch 。バージョンのすべての部分が必須とは限らない。つまり、「7」だけでも有効である。	必須: いいえ
ConsoleContextPath	WLS コンソールのコンテキストパスを指定する。	Administration Console のフィールド表示: [コンソールコンテキストパス] 必須: いいえ デフォルト値: console
ConsoleEnabled	このドメインの WLS コンソールを自動デプロイするかどうかを示す。	Administration Console のフィールド表示: [コンソール有効化] 必須: いいえ デフォルト値: true セキュアな値: false

表 8-1 Domain の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示: [メモ] 必須: いいえ

DomainLogFilter

説明

この MBean は、ドメイン ログ ファイルに記録されるログ メッセージを限定するためのフィルタを表します。メッセージは、フィルタを通過するために指定されたすべての基準を通過する必要があります。必要に応じて、この MBean の複数のインスタンスを定義できます。複数のインスタンスが存在する場合、メッセージは少なくとも 1 つのフィルタを通過しなければドメイン ログファイルに記録されません。

構文

```
<DomainLogFilter
  Name="String"
  Notes="String"
  SeverityLevel=( "64" | "32" | "16" | "8" | "4" | "2" | "1" )
  SubsystemNames="list of Strings"
  UserIds="list of Strings"
/>
```

親要素

- Domain

属性

表 9-1 DomainLogFilter の属性

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示:[名前] 必須:いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示:[メモ] 必須:いいえ

表 9-1 DomainLogFilter の属性 (続き)

属性	説明	指定できる値とデフォルト値
SeverityLevel	<p>このフィルタがドメイン ログにメッセージを転送する基準となる最低の重大度。指定した重大度以上のすべてのメッセージがドメイン ログに転送される。次に重大度を昇順で示す。</p> <ul style="list-style-type: none"> • INFO(64)。通常の処理を報告する。 • WARNING(32)。不審な処理またはコンフィグレーションが行われたが、通常の処理に影響する恐れはない。 • ERROR(16)。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では、サービスの中断や停止なしでエラーに対処できる。 • NOTICE(8)。警告メッセージ。サーバの通常の処理には影響しないと考えられる不審な処理またはコンフィグレーションが行われた。 • CRITICAL(4)。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。 • ALERT(2)。システムの特定のサービスだけが使用不能の状態にある。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。 • EMERGENCY(1)。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态を示す。 	<p>Administration Console のフィールド表示 : [重大度]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 16</p>

表 9-1 DomainLogFilter の属性 (続き)

属性	説明	指定できる値とデフォルト値
SubsystemNames	このフィルタに該当するメッセージの、ドメイン ログへの転送元となるサブシステム。サブシステムを選択しなかった場合、フィルタに該当するメッセージがすべてのサブシステムから送信される。	Administration Console のフィールド表示 : [サブシステム] 必須 : いいえ
UserIds	ドメイン ログに送信されるメッセージに関連付けるユーザ ID。各メッセージには、メッセージが生成されたセキュリティ コンテキストからユーザ ID が付加される。1 つ以上のユーザ ID が関連付けられたメッセージのみを転送するフィルタを作成することもできる。 ID を指定しない場合、すべてのユーザ ID からメッセージがドメイン ログに送信される。	Administration Console のフィールド表示 : [ユーザ ID] 必須 : いいえ

EJBComponent

説明

WebLogic Server が EJB モジュール用に管理するすべてのコンフィグレーション情報の最上位インタフェース。

構文

```
<EJBComponent
  DeploymentOrder=number"
  ExtraEjbcOptions=String"
  ExtraRmicOptions=String"
  ForceGeneration=( "true" | "false" )
  JavaCompiler=String"
  JavaCompilerPostClassPath=String"
  JavaCompilerPreClassPath=String"
  KeepGenerated=( "true" | "false" )
  Name=String"
  Notes=String"
  Targets=list of Target names"
  TmpPath=String"
  URI=String"
  VerboseEJBDeploymentEnabled=String"
/>
```

親要素

- Application

属性

表 10-1 EJBComponent の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>Administration Console のフィールド表示: [デプロイ順]</p> <p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
ExtraEjbcOptions	<p>jar ファイルの動的 ejbc の途中で ejbc に渡される追加オプションを返す (-J-mx128m など)。注意: この属性のデフォルトは null でなければならない。EJBComponent に対して ExtraEJBOptions を指定しない場合、デフォルト値は Server.ExtraEJBOptions から取得される。</p>	<p>Administration Console のフィールド表示: [追加 Ejbc オプション]</p> <p>必須: いいえ</p>

表 10-1 EJBComponent の属性 (続き)

属性	説明	指定できる値とデフォルト値
ExtraRmicOptions	サーバサイド生成中に <code>rmic</code> に渡される追加のオプションを返す。注意：この属性のデフォルトは <code>null</code> でなければならない。EJBComponent に対して <code>ExtraRmicOptions</code> を指定しない場合、デフォルト値は <code>Server.ExtraRmicOptions</code> から取得される。	Administration Console のフィールド表示：[追加 Rmic オプション] 必須：いいえ
ForceGeneration	<code>ejbc</code> がラッパー クラスを強制的に再生成する場合は <code>true</code> を返し、再生成の必要があると判断した場合にのみファイルを再生成する場合は <code>false</code> を返す。	Administration Console のフィールド表示：[強制的に生成] 必須：いいえ デフォルト値：false
JavaCompiler	EJB のコンパイルに使用する Java コンパイラ (「 <code>sj</code> 」 や 「 <code>javac</code> 」 など) のパスを返す。注意：この属性のデフォルトは <code>null</code> でなければならない。この EJBComponent に対して <code>JavaCompiler</code> を指定しない場合、デフォルト値は <code>EJBContainerMBean</code> から、次に <code>Server.JavaCompiler</code> から取得される。	Administration Console のフィールド表示：[Java コンパイラ] 必須：いいえ
JavaCompilerPostClassPath	Java コードをコンパイルする必要があるときに Java コンパイラのクラスパスの後ろに付加されるオプションを返す。	必須：いいえ
JavaCompilerPreClassPath	Java コードをコンパイルする必要があるときに Java コンパイラのクラスパスの前に付加されるオプションを返す。	必須：いいえ
KeepGenerated	<code>ejbc</code> が生成したソース ファイルを残す場合は <code>true</code> を返し、コンパイル後に削除する場合は <code>false</code> を返す。	Administration Console のフィールド表示：[生成したソース ファイルを残す] 必須：いいえ

表 10-1 EJBComponent の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	Administration Console の フィールド表示 : [対象] 必須 : いいえ
TmpPath	ejbc が生成済みのファイルを格納する一時ディレクトリを返す。	Administration Console の フィールド表示 : [テンポラリパス] 必須 : いいえ デフォルト値 : tmp_ejb
URI	アプリケーション コンポーネントを示す URI を設定する (通常は管理サーバ上にある)。	Administration Console の フィールド表示 : [URI]
VerboseEJBDeploymentEnabled	EJB の verbose デプロイメントが有効である場合、true を返す。	必須 : いいえ デフォルト値 : false

EJBContainer

説明

この MBean は、EJB コンテナ全体の設定を指定するために使用されます。これらの設定は、特定の EJBComponentMBean でオーバーライドできます。

構文

```
<EJBContainer
  ExtraEjbcOptions="String"
  ExtraRmicOptions="String"
  ForceGeneration=( "true" | "false" )
  JavaCompiler="String"
  JavaCompilerPostClassPath="String"
  JavaCompilerPreClassPath="String"
  KeepGenerated=( "true" | "false" )
  Name="String"
  Notes="String"
  TmpPath="String"
  VerboseEJBDeploymentEnabled="String"
/>
```

親要素

- Domain

属性

表 11-1 EJBContainer の属性

属性	説明	指定できる値とデフォルト値
ExtraEjbcOptions	jar ファイルの動的 ejbc の途中で ejbc に渡される追加オプションを返す (-J-mx128m など)。注意：この属性のデフォルトは null でなければならない。EJBComponent に対して ExtraEJBOptions を指定しない場合、デフォルト値は Server.ExtraEJBOptions から取得される。	必須：いいえ
ExtraRmicOptions	サーバサイド生成中に rmic に渡される追加のオプションを返す。注意：この属性のデフォルトは null でなければならない。EJBComponent に対して ExtraRmicOptions を指定しない場合、デフォルト値は Server.ExtraRmicOptions から取得される。	必須：いいえ
ForceGeneration	ejbc がラッパー クラスを強制的に再生成する場合は true を返し、再生成の必要があると判断した場合にのみファイルを再生成する場合は false を返す。	必須：いいえ デフォルト値：false

表 11-1 EJBContainer の属性 (続き)

属性	説明	指定できる値とデフォルト値
JavaCompiler	EJB のコンパイルに使用する Java コンパイラ (「sj」 や 「javac」 など) のパスを返す。注意 : この属性のデフォルトは null でなければならない。この EJBComponent に対して JavaCompiler を指定しない場合、デフォルト値は EJBContainerMBean から、次に Server.JavaCompiler から取得される。	必須 : いいえ
JavaCompilerPostClassPath	Java コードをコンパイルする必要があるときに Java コンパイラのクラスパスの後ろに付加されるオプションを返す。	必須 : いいえ
JavaCompilerPreClassPath	Java コードをコンパイルする必要があるときに Java コンパイラのクラスパスの前に付加されるオプションを返す。	必須 : いいえ
KeepGenerated	ejbc が生成したソース ファイルを残す場合は true を返し、コンパイル後に削除する場合は false を返す。	必須 : いいえ デフォルト値 : true
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
TmpPath	ejbc が生成済みのファイルを格納する一時ディレクトリを返す。	必須 : いいえ デフォルト値 : tmp_ejb
VerboseEJBDeploymentEnabled	EJB の verbose デプロイメントが有効である場合、true を返す。	必須 : いいえ デフォルト値 : false

EmbeddedLDAP

説明

この MBean は、組み込み LDAP サーバのコンフィグレーション プロパティを定義します。

構文

```
<EmbeddedLDAP
  BackupCopies="number"
  BackupHour="number"
  BackupMinute="number"
  CacheEnabled=( "true" | "false" )
  CacheSize="number"
  CacheTTL="number"
  Credential="String"
  Name="String"
  Notes="String"
/>
```

親要素

- Domain

属性

表 12-1 EmbeddedLDAP の属性

属性	説明	指定できる値とデフォルト値
BackupCopies	組み込み LDAP サーバのバックアップコピー数。	Administration Console のフィールド表示:[バックアップコピー数] 必須: いいえ デフォルト値: 7 最小値: 0 最大値: 65534
BackupHour	組み込み LDAP サーバをバックアップする時刻(時)。	Administration Console のフィールド表示:[バックアップ時間(時間)] 必須: いいえ デフォルト値: 23 最小値: 0 最大値: 23

表 12-1 EmbeddedLDAP の属性 (続き)

属性	説明	指定できる値とデフォルト値
BackupMinute	組み込み LDAP サーバをバックアップする時刻 (分)。この属性は BackupHour 属性と組み合わせて使用され、組み込み LDAP サーバをバックアップする時刻を決定する。	Administration Console のフィールド表示 : [バックアップ時間 (分)] 必須 : いいえ デフォルト値 : 5 最小値 : 0 最大値 : 59
CacheEnabled	組み込み LDAP サーバでキャッシュを使用するかどうかを指定する。	Administration Console のフィールド表示 : [キャッシュを有効化] 必須 : いいえ デフォルト値 : true
CacheSize	組み込み LDAP サーバで使用されるキャッシュのサイズ (単位は K)。	Administration Console のフィールド表示 : [キャッシュサイズ] 必須 : いいえ デフォルト値 : 32 最小値 : 0
CacheTTL	キャッシュの存続期間 (TTL) の秒数を取得する。	Administration Console のフィールド表示 : [キャッシュ TTL] 必須 : いいえ デフォルト値 : 60 最小値 : 0
Credential	組み込み LDAP サーバへの接続に使用される資格 (通常はパスワード)。	Administration Console のフィールド表示 : [資格] 必須 : いいえ 暗号化 : はい

表 12-1 EmbeddedLDAP の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ

ExecuteQueue

説明

この MBean は、実行キューとそれに関連付けられたスレッドプールをコンフィグレーションするために使用されます。

構文

```
<ExecuteQueue  
  Name="String"  
  Notes="String"  
  ThreadCount="number"  
>  
</>
```

親要素

- Server

属性

表 13-1 ExecuteQueue の属性

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示: [メモ] 必須: いいえ
ThreadCount	このキューに割り当てられるスレッド数を返す。	Administration Console の フィールド表示: [スレッド数] 必須: いいえ デフォルト値: 15 最小値: 1 最大値: 65536

FileRealm

説明

構文

```
<FileRealm
  MaxACLS="number"
  MaxGroups="number"
  MaxUsers="number"
  Name="String"
  Notes="String"
/>
```

親要素

- Domain
- Domain

属性

表 14-1 FileRealm の属性

属性	説明	指定できる値とデフォルト値
MaxACLs	ファイル レalmでサポートされる ACL の (正の) 最大数。最大値には必ずしも従わなくてよいが、最大値に達すると警告が表示される。	Administration Console の フィールド表示 : [最大 ACL] 必須 : いいえ デフォルト値 : 1000 最小値 : 1 最大値 : 10000
MaxGroups	ファイル レalmでサポートされるグループの最大数。	Administration Console の フィールド表示 : [最大グループ数] 必須 : いいえ デフォルト値 : 1000 最小値 : 1 最大値 : 10000
MaxUsers	ファイル レalmでサポートされるユーザの最大数。	Administration Console の フィールド表示 : [最大ユーザ数] 必須 : いいえ デフォルト値 : 1000 最小値 : 1 最大値 : 10000

表 14-1 FileRealm の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ

FileT3

説明

File T3 のコンフィグレーション エントリです。

構文

```
<FileT3
  DeploymentOrder=number
  Name=String
  Notes=String
  Path=String
  Targets=list of Target names
/>
```

親要素

- Domain

属性

表 15-1 FileT3 の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須：いいえ</p> <p>デフォルト値：1000</p> <p>最小値：0</p> <p>最大値：2³¹-1</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示：[名前]</p> <p>必須：いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示：[メモ]</p> <p>必須：いいえ</p>

表 15-1 FileT3 の属性 (続き)

属性	説明	指定できる値とデフォルト値
Path	<p>T3 ファイル サービスをサーバ ホスト上に配置するのに使用するパスを定義する。</p> <p>次に例を示す。ファイル システム名 <code>users</code> をサーバ ホスト上のパス <code>/usr/local/tmp</code> にマップするには、Name 属性の値に <code>users</code>、Path 属性の値に <code>/usr/local/tmp</code> をそれぞれ指定する。</p>	<p>Administration Console のフィールド表示 : [パス]</p> <p>必須 : いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須 : いいえ</p>

IIOP

説明

IIOP プロパティのコンフィグレーションを示します。

構文

```
<IIOP
  CompleteMessageTimeout="number of seconds"
  IdleConnectionTimeout="number of seconds"
  MaxMessageSize="number of bytes"
  Name="String"
  Notes="String"
/>
```

親要素

- Server

属性

表 16-1 IIOp の属性

属性	説明	指定できる値とデフォルト値
CompleteMessageTime out	完全な IIOp メッセージを取得するまでの最大秒数を指定する。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。	必須: いいえ 単位: 秒 デフォルト値: 60 最小値: 0 最大値: 480
IdleConnectionTimeout	サーバによって閉じられるまで、IIOp 接続がアイドルであることが許可される最大秒数を指定する。この属性は、開いている接続が多すぎることによって発生するサーバのデッドロックの防止に有用である。	必須: いいえ 単位: 秒 デフォルト値: 60 最小値: 0
MaxMessageSize	メッセージヘッダ内の最大 IIOp メッセージサイズを指定する。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てようとするサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。	必須: いいえ 単位: バイト デフォルト値: 10000000 最小値: 4096 最大値: 2000000000
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ

JDBCConnectionPool

説明

この Bean は、JDBC 接続プールを定義します。

構文

```
<JDBCConnectionPool
  ACLName="String"
  CapacityIncrement="number"
  ConnLeakProfilingEnabled=( "true" | "false" )
  CountOfRefreshFailuresTillDisable="number"
  CountOfTestFailuresTillFlush="number"
  DeploymentOrder="number"
  DriverName="String"
  EnableResourceHealthMonitoring=( "true" | "false" )
  InitialCapacity="number"
  LoginDelaySeconds="number of seconds"
  MaxCapacity="number"
  Name="String"
  Notes="String"
  Password="String"
  PreparedStatementCacheSize="number"
  Properties="java.util.Properties"
  RefreshMinutes="number of minutes"
```

```

ShrinkPeriodMinutes="number of minutes"
ShrinkingEnabled=( "true" | "false" )
SupportsLocalTransaction=( "true" | "false" )
Targets="list of Target names"
TestConnectionsOnRelease=( "true" | "false" )
TestConnectionsOnReserve=( "true" | "false" )
TestTableName="String"
URL="String"
XAPassword="String"
XAPreparedStatementCacheSize="number"
/>

```

親要素

- Domain

属性

表 17-1 JDBCConnectionPool の属性

属性	説明	指定できる値とデフォルト値
ACLName	非推奨 。当該接続プールへのアクセスを制御するための ACL を返す。	Administration Console の フィールド表示 : [ACL 名] 必須 : いいえ
CapacityIncrement	接続プール容量の拡張単位となる増加量。サービス リクエストに提供するための利用可能な物理接続が他にないとき、接続プールはこの数の物理データベース接続を追加作成して、それらを接続プールに追加する。接続プールは MaxCapacity で設定された物理接続の最大数を超えない。	Administration Console の フィールド表示 : [増加容量] 必須 : いいえ デフォルト値 : 1 最小値 : 1 最大値 : 2 ³¹ -1

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnLeakProfilingEnabled	<p>このプロパティによって JDBC 接続リーク プロファイルが有効になる。</p> <p>接続リークは、プールからの接続が <code>close()</code> の呼び出しで明示的にクローズされていないときに発生する。</p> <p>接続リーク プロファイルがアクティブの場合、プールは接続オブジェクトがプールから割り当てられ、クライアントに与えられたときにスタックトレースを格納する。接続リークが検出されたとき (接続オブジェクトのガベージコレクションが行われたとき) に、このスタックトレースが報告される。</p> <p>この機能はリソースを余計に使用し、接続プール処理を遅くする可能性があるため、プロダクション環境では使用しないほうがよい。</p>	<p>必須: いいえ</p> <p>デフォルト値: false</p> <p>セキュアな値: false</p>
CountOfRefreshFailuresTillDisable	<p>切断されたプール接続の置き換えに連続して失敗できる回数を設定する。この回数を超えると、プールが無効になる。ゼロに設定すると、プールは無効にならない。</p>	<p>必須: いいえ</p> <p>デフォルト値: 0</p>
CountOfTestFailurestTillFlush	<p>プール接続のテストに連続して失敗できる回数を設定する。この回数を超えると、プールのすべての接続がクローズされる。ゼロに設定すると、プールの接続はクローズされない。</p>	<p>必須: いいえ</p> <p>デフォルト値: 0</p>

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
DriverName	<p>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。次に例を示す。</p> <p><code>com.pointbase.jdbc.jdbcUniversalDriver</code></p> <p>これは、<code>java.sql.Driver</code> インタフェースを実装するクラスの名前である必要がある。絶対パス名については、使用する JDBC ドライバのマニュアルを参照。</p>	<p>Administration Console のフィールド表示: [ドライバクラス名]</p> <p>必須: いいえ</p>
EnableResourceHealthMonitoring	<p>この XA 接続プールで JTA リソース状態モニタが有効になっている場合に <code>true</code> を返す。</p> <p>このプロパティは XA 接続プールにのみ適用され、XA 以外のドライバを使用する接続プールの場合は無視される。</p>	<p>必須: いいえ</p> <p>デフォルト値: <code>true</code></p>

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
InitialCapacity	接続プールの作成時に作成する物理データベース接続の数。	Administration Console の フィールド表示 : [初期容量] 必須 : いいえ デフォルト値 : 1 最小値 : 0 最大値 : $2^{31}-1$
LoginDelaySeconds	各物理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。	Administration Console の フィールド表示 : [ログイン遅延時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 0 最小値 : 0 最大値 : $2^{31}-1$
MaxCapacity	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。JDBC ドライバおよびデータベース サーバによっては、可能な物理的接続の数が制限されている場合もある。	Administration Console の フィールド表示 : [最大容量] 必須 : いいえ デフォルト値 : 15 最小値 : 1 最大値 : $2^{31}-1$
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ セキュアな値 : false

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
Password	setPassword() で設定されるか、または setProperties() で key=value ペアとして設定されるデータベースパスワードを返す。	Administration Console の フィールド表示 : [パスワード] 必須 : いいえ 暗号化 : はい
PreparedStatement CacheSize	prepared statement キャッシュのサイズを設定する。このキャッシュサイズは、特定の接続から作成される prepared statement の数を表し、後で使用できるようにキャッシュに格納される。prepared statement キャッシュのサイズを 0 に設定する。後で使用するためにキャッシュに格納される prepared statement の数。WebLogic Server では、再ロードせずにキャッシュ内の prepared statement を再利用できるため、サーバのパフォーマンスが向上する。prepared statement キャッシュのサイズを 0 (デフォルト) にすると、キャッシュは使用されない。	Administration Console の フィールド表示 : [Prepared Statement キャッシュ サイズ] 必須 : いいえ デフォルト値 : 5
Properties	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プロパティのリストを返す。	Administration Console の フィールド表示 : [プロパティ (key=value)] 必須 : いいえ
RefreshMinutes	データベース接続テストの周期の分数。RefreshMinutes の間隔が過ぎるたびに、未使用のデータベース接続が、TestTableName を使ってテストされる。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を再確立するために再び開かれる。TestTableName が設定されていない場合、テストは実行されない。	Administration Console の フィールド表示 : [更新間隔] 必須 : いいえ 単位 : 分 デフォルト値 : 0 最小値 : 0 最大値 : 35791394

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
ShrinkPeriodMinutes	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)。接続プールを縮小するには、 <code>ShrinkingEnabled</code> が <code>true</code> に設定されている必要がある。	Administration Console のフィールド表示 : [縮小間隔] 必須 : いいえ 単位 : 分 デフォルト値 : 15 最小値 : 1 最大値 : $2^{31}-1$
ShrinkingEnabled	トラフィックの増加時に作成された接続が未使用であることが検出されたときにプールが <code>InitialCapacity</code> の値まで縮小できるかどうかを示す。	Administration Console のフィールド表示 : [縮小可] 必須 : いいえ デフォルト値 : <code>true</code> セキュアな値 : <code>false</code>
SqlStmtProfilingEnabled	SQL ラウンドトリップ プロファイルに、SQL 文のテキストや実行時間などのメトリックを格納する。	必須 : いいえ デフォルト値 : <code>false</code>
SupportsLocalTransaction	物理データベース接続を作成するための XA ドライバが、グローバルトランザクションを使用しない SQL をサポートしている場合は、 <code>true</code> に設定する。XA ドライバが、グローバルトランザクションを使用しない SQL をサポートしていない場合は、 <code>false</code> (デフォルト) に設定する。 このプロパティは XA ドライバのみを使用する接続プールに適用され、XA 以外のドライバを使用する接続プールでは無視される。	Administration Console のフィールド表示 : [ローカルトランザクションのサポート] 必須 : いいえ デフォルト値 : <code>false</code>
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	Administration Console のフィールド表示 : [対象] 必須 : いいえ

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
TestConnectionsOnRelease		Administration Console の フィールド表示 : [リリースされたときに接続をテスト] 必須 : いいえ デフォルト値 : false
TestConnectionsOnReserve		Administration Console の フィールド表示 : [リザーブされたときに接続をテスト] 必須 : いいえ デフォルト値 : false
TestTableName	<p>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。接続をテストする際に使用されるデフォルト SQL コードは、次の通り。</p> <pre>"select count(*) from TestTableName"</pre> <p>TestTableName が存在し、接続を使用するデータベース ユーザがアクセスできる必要がある。ほとんどのデータベース サーバはこの SQL を最適化して、テーブル スキャンを回避する。それでも、TestTableName を、行が少ない (またはまったくない) テーブルの名前に設定することは有益である。</p> <p>TestTableName が「SQL」で始まる場合、「SQL」より後の文字列は、接続をテストするリテラルの SQL 文として扱われる。</p>	Administration Console の フィールド表示 : [テスト テーブル名] 必須 : いいえ
URL	当該接続プールで接続を作成するためのデータベース URL を返す。	Administration Console の フィールド表示 : [URL] 必須 : いいえ

表 17-1 JDBCConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
XAPassword	<p>設定されている場合、この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、物理データベース接続の作成に使用される。値は、<code>config.xml</code> ファイルに暗号化されて格納される。この値は、<code>Properties</code> で定義されたオープン文字列のパスワード値をオーバーライドする。</p>	<p>Administration Console のフィールド表示 : [オープン文字列のパスワード]</p> <p>必須 : いいえ</p> <p>暗号化 : はい</p>
XAPreparedStatementCacheSize		<p>Administration Console のフィールド表示 : [XAPreparedStatement Cache Size]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 5</p>

JDBCDataSource

説明

この MBean は、トランザクション非対応の JDBC データソースを定義します。

構文

```
<JDBCDataSource
  ConnectionWaitPeriod=number
  DeploymentOrder=number
  JNDIName=String
  Name=String
  Notes=String
  PoolName=String
  RowPrefetchEnabled=( "true" | "false" )
  RowPrefetchSize=number
  StreamChunkSize=number of bytes
  Targets=list of Target names
  WaitForConnectionEnabled=( "true" | "false" )
/>
```

親要素

- Domain

属性

表 18-1 JDBCDataSource の属性

属性	説明	指定できる値とデフォルト値
ConnectionWaitPeriod	<p>プールにフリーなエントリがないときにリクエストを受信した場合に、フリーな接続プールエントリが使用できるようになるまでシステムが待機する時間 (秒単位)。この値は、<code>setWaitForConnectionEnabled (true)</code> が以前に呼び出された場合にのみ使用される。フリーな接続を待機している間は WLS サーバスレッドが 1 つ占有されるので、結果としてシステムのスループットが低下する場合がある。接続プールのエントリ不足に対処するには、プールのサイズを大きくすることをお勧めする。プロダクション環境ではこの方法を使用しないほうがよい。</p>	<p>必須: いいえ デフォルト値: 1 最小値: 1 最大値: 60</p>
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ デフォルト値: 1000 最小値: 0 最大値: $2^{31}-1$</p>

表 18-1 JDBCDataSource の属性 (続き)

属性	説明	指定できる値とデフォルト値
JNDIName	当該データ ソースのバインド先の JNDI パス。 JNDI パスをルックアップするアプリケーションは、このデータ ソースに対応する <code>javax.sql.DataSource</code> インスタンスを取得する。	Administration Console のフィールド表示: [JNDI 名]
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ
PoolName	当該データ ソースに関連付けられる接続プールの名前。 このデータ ソースでアプリケーションから <code>getConnection()</code> を呼び出すと、関連付けられた接続プールから接続が返る。	Administration Console のフィールド表示: [プール名] 必須: いいえ
RowPrefetchEnabled	各 <code>ResultSet</code> について、クライアントと WebLogic Server の間での行のプリフェッチを制御する。外部クライアントが JDBC を使用して WebLogic Server 経由でデータベースにアクセスするとき、行のプリフェッチは、1 回のサーバアクセスでサーバからクライアントに複数の行を取り出すのでパフォーマンスが向上する。クライアントと WebLogic Server が同一の JVM 内にある場合は、WebLogic Server はこの設定を無視し、行のプリフェッチを使用しない。	Administration Console のフィールド表示: [行のプリフェッチを有効化] 必須: いいえ デフォルト値: <code>false</code>

表 18-1 JDBCDataSource の属性 (続き)

属性	説明	指定できる値とデフォルト値
RowPrefetchSize	クライアント用にプリフェッチする結果セットの行数。最適値はクエリの詳細によって異なる。一般に、この数を増やすと、特定の値に達するまでパフォーマンスが向上する。その値に達すると、それ以上数を増やしてもパフォーマンスはそれほど向上しない。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。	Administration Console のフィールド表示: [Row Prefetch サイズ] 必須: いいえ デフォルト値: 48 最小値: 2 最大値: 65536
StreamChunkSize	ストリーム データ型のデータ チャンク サイズ。getBinaryStream() への呼び出し結果などのストリーミング データ型は、必要に応じて WebLogic Server からクライアントに StreamChunkSize サイズのチャンクでプルされる。	Administration Console のフィールド表示: [ストリーム チャンク サイズ] 必須: いいえ 単位: バイト デフォルト値: 256 最小値: 1 最大値: 65536
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	Administration Console のフィールド表示: [対象] 必須: いいえ
Type	MBean の型を返す。	必須: いいえ
WaitForConnectionEnabled	プールの全エントリの使用中にリクエストが行われたときに、フリーな接続プールエントリが使用できるようになるまでシステムが待機する場合に true を返す。サーバスレッドの拘束とパフォーマンスの低下を避けるには、この方法を使用しないほうがよい。	必須: いいえ デフォルト値: false

JDBCDataSourceFactory

説明

この MBean は、アプリケーション スコープ JDBC 接続プールにアクセスするためにアプリケーションで使用するデータソースを作成するためのオブジェクトを表します。

構文

```
<JDBCDataSourceFactory
  DriverClassName="String"
  FactoryName="String"
  Name="String"
  Notes="String"
  Properties="java.util.Map"
  URL="String"
  UserName="String"
/>
```

親要素

- Domain

属性

表 19-1 JDBCDataSourceFactory の属性

属性	説明	指定できる値とデフォルト値
DriverClassName	ドライバの名前。これは記述子に定義されているドライバ名によってオーバーライドされる場合がある。	Administration Console のフィールド表示: [ドライバクラス名] 必須: いいえ
FactoryName	データソースファクトリの名前。これは weblogic-application.xml の connection-factory 要素から参照される。	Administration Console のフィールド表示: [ファクトリ名] 必須: いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
Properties	デフォルトの接続プロパティ。	Administration Console のフィールド表示: [プロパティ] 必須: いいえ

表 19-1 JDBCDataSourceFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
URL	接続 URL。これは記述子に定義されている URL によってオーバーライドされる場合がある。	Administration Console の フィールド表示 : [URL] 必須 : いいえ
UserName	データベース ユーザ名。これは記述子に定義されているユーザ名によってオーバーライドされる場合がある。	Administration Console の フィールド表示 : [ユーザ名] 必須 : いいえ

JDBCMultiPool

説明

この MBean は、JDBC 接続プールのプールである JDBC マルチプールを表します。

構文

```
<JDBCMultiPool
  ACLName=String
  AlgorithmType=( "High-Availability" | "Load-Balancing" )
  ConnectionPoolFailoverCallbackHandler=String
  DeploymentOrder=number
  FailoverRequestIfBusy=( "true" | "false" )
  HealthCheckFrequencySeconds=number
  Name=String
  Notes=String
  PoolList=list of JDBCConnectionPool names
  Targets=list of Target names
/>
```

親要素

- Domain

属性

表 20-1 JDBCMultiPool の属性

属性	説明	指定できる値とデフォルト値
ACLName	非推奨 。当該接続プールへのアクセスを制御するために使うアクセス制御リスト (ACL)。	Administration Console のフィールド表示: [ACL 名] 必須: いいえ
AlgorithmType	<p>当該マルチプールのアルゴリズム タイプ。</p> <p>High-Availability に設定した場合、接続プールは順序付きリストとして設定される。すなわち、アプリケーションがマルチプールに接続を要求するたびに、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。リストの終わりに達した場合には、例外が送出される。</p> <p>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。</p> <p>Load-Balancing に設定した場合、マルチプールは接続要求をそのメンバー プールに均等に分散する。また、このアルゴリズムは高可用性アルゴリズムと同じフェイルオーバー機能を実行する。</p> <p>この属性のデフォルト値は High-Availability である。</p>	<p>Administration Console のフィールド表示: [アルゴリズム タイプ]</p> <p>必須: いいえ</p> <p>デフォルト値: High-Availability</p>

表 20-1 JDBCMultiPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnectionPoolFailoverCallbackHandler	<p>weblogic.jdbc.extensions.ConnectionPoolFailoverCallback インタフェースを実装するアプリケーションクラスの絶対名。</p> <p>これを設定すると、マルチプールがリスト内の次の接続プールにフェイルオーバーする前に、WebLogic Server が指定のコールバックアプリケーションを呼び出して戻り値を待機する。コールバックアプリケーションから返された値に応じて、マルチプールが同じ接続プールを試行するか、次の接続プールにフェイルオーバーするか、または失敗して例外を送出するかが決まる。</p> <p>また、元の接続プールがフェイルバック可能な状態になったときも、マルチプールがコールバックアプリケーションを呼び出す。</p> <p>この属性は、高可用性アルゴリズムのマルチプールにのみ適用される。</p>	<p>必須：いいえ</p>
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須：いいえ</p> <p>デフォルト値：1000</p> <p>最小値：0</p> <p>最大値：2³¹-1</p>

表 20-1 JDBCMultiPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
FailoverRequestfBusy	<p>現在の接続プール内のすべての接続が使用中である場合に、マルチプールが次の接続プールにフェイルオーバーするのを可能にする。</p> <p>この属性を true に設定すると、現在の接続プールにあるすべての接続が使用中の場合に、マルチプール内の代替可能な接続プールにアプリケーションの接続リクエストがルーティングされる。</p> <p>false に設定すると、接続リクエストはフェイルオーバーしない。</p> <p>この設定は、高可用性アルゴリズムで実行されている場合にのみ意味を持つ。</p>	<p>必須: いいえ</p> <p>デフォルト値: false</p>
HealthCheckFrequencySeconds	<p>これまで使用不能で無効と認識されていた接続プールの状態をマルチプールがチェックする頻度。</p>	<p>必須: いいえ</p> <p>デフォルト値: 300</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>
PoolList	<p>マルチ プール内の接続プールのリスト。</p>	<p>Administration Console のフィールド表示: [プールリスト]</p> <p>必須: いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>Administration Console のフィールド表示: [対象]</p> <p>必須: いいえ</p>

JDBCPoolComponent

説明

構文

```
<JDBCPoolComponent
  DeploymentOrder="number"
  Name="String"
  Notes="String"
  Targets="list of Target names"
  URI="String"
/>
```

親要素

- Application

属性

表 21-1 JDBCPoolComponent の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>必須: いいえ</p>
Targets	<p>このデプロイメントの対象を設定する。対象はクラスタかサーバでなければならない。</p>	<p>必須: いいえ</p>
URI	<p>アプリケーション コンポーネントを示す URI を設定する (通常は管理サーバ上にある)。</p>	

JDBCTxDataSource

説明

この MBean は、トランザクション対応の JDBC データソースを定義します。

構文

```
<JDBCTxDataSource
  DeploymentOrder=number
  EnableTwoPhaseCommit=( "true" | "false" )
  JNDIName=String
  Name=String
  Notes=String
  PoolName=String
  RowPrefetchEnabled=( "true" | "false" )
  RowPrefetchSize=number
  StreamChunkSize=number of bytes
  Targets=list of Target names
/>
```

親要素

- Domain

属性

表 22-1 JDBCTxDataSource の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
EnableTwoPhaseCommit	<p>true に設定すると、非 XA JDBC ドライバで、JTA を使用した分散トランザクションへの参加/不参加をエミュレートできる。</p> <p>このオプションは、トランザクションに参加するリソースが JDBC 接続のみであり、XA 対応の JDBC ドライバが存在しない場合に使用する。トランザクションに参加するリソースが複数存在し、それらの JDBC ドライバの 1 つが XA リソースをエミュレートする場合、ヒューリスティック エラーが発生するおそれがある。</p> <p>トランザクション データ ソースが XA 接続プールに関連づけられているか、または分散トランザクションに参加しているリソースが 1 つのみの場合は、この設定は無視される。</p>	<p>Administration Console のフィールド表示: [非 XA ドライバ用に 2 フェーズ コミットをエミュレート]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p>

表 22-1 JDBCTxDataSource の属性 (続き)

属性	説明	指定できる値とデフォルト値
JNDIName	<p>当該トランザクション データ ソースのバインド先の JNDI パス。</p> <p>JNDI パスをルックアップするアプリケーションは、このデータ ソースに対応する <code>javax.sql.DataSource</code> インスタンスを取得する。</p>	<p>Administration Console のフィールド表示: [JNDI 名]</p> <p>必須: いいえ</p>
Name	<p>このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>
PoolName	<p>当該トランザクション データ ソースに関連付けられる JDBC 接続プールの名前。</p> <p>このトランザクション データ ソースでアプリケーションから <code>getConnection()</code> を呼び出すと、関連付けられた接続プールから接続が返る。</p>	<p>Administration Console のフィールド表示: [プール名]</p> <p>必須: いいえ</p>
RowPrefetchEnabled	<p>各 <code>ResultSet</code> について、クライアントと WebLogic Server の間での行のプリフェッチを制御する。外部クライアントが JDBC を使用して WebLogic Server 経由でデータベースにアクセスするとき、行のプリフェッチは、1 回のサーバアクセスでサーバからクライアントに複数の行を取り出すのでパフォーマンスが向上する。クライアントと WebLogic Server が同一の JVM 内にある場合は、WebLogic Server はこの設定を無視し、行のプリフェッチを使用しない。</p>	<p>Administration Console のフィールド表示: [行のプリフェッチを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: <code>false</code></p>

表 22-1 JDBCDataSource の属性 (続き)

属性	説明	指定できる値とデフォルト値
RowPrefetchSize	クライアント用にプリフェッチする結果セットの行数。最適値はクエリの詳細によって異なる。一般に、この数を増やすと、特定の値に達するまでパフォーマンスが向上する。その値に達すると、それ以上数を増やしてもパフォーマンスはそれほど向上しない。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。	Administration Console の フィールド表示: [Row Prefetch サイズ] 必須: いいえ デフォルト値: 48 最小値: 2 最大値: 65536
StreamChunkSize	ストリームデータ型のデータチャンクサイズ。getBinaryStream() への呼び出し結果などのストリーミングデータ型は、必要に応じて WebLogic Server からクライアントに StreamChunkSize サイズのチャンクでプルされる。	Administration Console の フィールド表示: [ストリーム チャンクサイズ] 必須: いいえ 単位: バイト デフォルト値: 256 最小値: 1 最大値: 65536
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	Administration Console の フィールド表示: [対象] 必須: いいえ

JMSBridgeDestination

説明

このクラスは、JMS 製品用のメッセージングブリッジ送り先を表します。

各メッセージングブリッジは、ブリッジングされている2つの送り先で構成されます。1つはメッセージの受信元になるソース送り先、もう1つはメッセージの送信先になる対象送り先です。

構文

```
<JMSBridgeDestination
  AdapterJNDIName="String"
  Classpath="String"
  ConnectionFactoryJNDIName="String"
  ConnectionURL="String"
  DestinationJNDIName="String"
  DestinationType=( "Queue" | "Topic" )
  InitialContextFactory="String"
  Name="String"
  Notes="String"
  UserName="String"
  UserPassword="String"
/>
```

親要素

- Domain

属性

表 23-1 JMSBridgeDestination の属性

属性	説明	指定できる値とデフォルト値
AdapterJNDIName	指定した送り先との通信に使用されるアダプタの JNDI 名を定義する。この名前は、アダプタのデプロイメント記述子ファイルで指定され、WebLogic Server JNDI でアダプタをバインドする WebLogic Server コネクタ コンテナによって使用される。	Administration Console のフィールド表示: [JNDI アダプタ名] デフォルト値: eis.jms.WLSConnectionFactoryJNDIXA
Classpath	ブリッジ送り先の CLASSPATH を定義する。主に、WebLogic JMS の別リリースへの接続に使用される。 接続する送り先がバージョン 6.0 以前の WebLogic Server で動作している場合、古い WebLogic Server 実装のクラス場所を示す CLASSPATH をブリッジ送り先に指定する必要がある。 注意: サードパーティの JMS 製品に接続する場合は、WebLogic Server の CLASSPATH で製品の CLASSPATH をブリッジ送り先に指定する必要がある。	Administration Console のフィールド表示: [アダプタ クラスパス] 必須: いいえ
ConnectionFactoryJNDIName	JMS ブリッジ送り先の接続ファクトリの JNDI 名を定義する。	Administration Console のフィールド表示: [接続ファクトリ JNDI 名]

表 23-1 JMSBridgeDestination の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnectionURL	JMS ブリッジ送り先の接続 URL を定義する。	Administration Console の フィールド表示 : [接続 URL] 必須 : いいえ
DestinationJNDIName	JMS ブリッジ送り先の送り先 JNDI 名を定義する。	Administration Console の フィールド表示 : [送り先 JNDI 名]
DestinationType	JMS ブリッジ送り先の送り先タイプ (キューまたはトピック) を定義する。	Administration Console の フィールド表示 : [送り先タイプ] 必須 : いいえ デフォルト値 : Queue
InitialContextFactory	JMS ブリッジ送り先の初期コンテキストファクトリ名を定義する。	Administration Console の フィールド表示 : [初期コンテキストファクトリ] 必須 : いいえ デフォルト値 : weblogic.jndi.WLInitialContextFactory
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ

表 23-1 JMSBridgeDestination の属性 (続き)

属性	説明	指定できる値とデフォルト値
UserName	ブリッジ送り先にアクセスする場合にアダプタで使用されるユーザ名を定義する (省略可能)。 注意：指定した送り先に対する処理はすべて、このユーザ名と、対応するパスワードを使って行われる。したがって、メッセージングブリッジを機能させるには、ソースおよび対象送り先のユーザ名/パスワードに基底の送り先へのアクセス権限が付与されている必要がある。	Administration Console の フィールド表示：[ユーザ名] 必須：いいえ
UserPassword	ブリッジ送り先にアクセスする場合にアダプタで使用されるユーザパスワードを定義する。	Administration Console の フィールド表示：[ユーザパスワード] 必須：いいえ 暗号化：はい

JMSConnectionFactory

説明

このクラスは、JMS 接続ファクトリを表します。接続ファクトリは、JMS クライアントが JMS 接続を作成することを可能にするオブジェクトです。

構文

```
<JMSConnectionFactory
  AcknowledgePolicy=( "All" | "Previous" )
  AllowCloseInOnMessage=( "true" | "false" )
  ClientId="String"
  DefaultDeliveryMode=( "Persistent" | "Non-Persistent" )
  DefaultPriority="number"
  DefaultRedeliveryDelay="number"
  DefaultTimeToDeliver="number"
  DefaultTimeToLive="number"
  DeploymentOrder="number"
  FlowControlEnabled=( "true" | "false" )
  FlowInterval="number of seconds"
  FlowMaximum="number of messages/second"
  FlowMinimum="number of messages/second"
  FlowSteps="number"
  JNDIName="String"
```

```
LoadBalancingEnabled=( "true" | "false" )
MessagesMaximum=number
Name=String
Notes=String
OverrunPolicy=( "KeepOld" | "KeepNew" )
ServerAffinityEnabled=( "true" | "false" )
Targets=list of Target names
TransactionTimeout=number
UserTransactionsEnabled=( "true" | "false" )
XAConnectionFactoryEnabled=( "true" | "false" )
XAEnabled=( "true" | "false" )
/>
```

親要素

- Domain

属性

表 24-1 JMSConnectionFactory の属性

属性	説明	指定できる値とデフォルト値
AcknowledgePolicy	<p>接続ファクトリの <i>AcknowledgePolicy</i> 属性の値を返す。</p> <p>注意：この属性は、非トランザクションセッション用に</p> <p>CLIENT_ACKNOWLEDGE 確認応答モードを使用する実装にのみ適用される。</p> <p>この属性は、JMS 仕様の変更の問題を回避する。具体的に説明すると、この仕様では確認応答されるメッセージとその前のすべてのメッセージを確認応答できた。しかし、以下に示すとおり、仕様の変更によって、任意のメッセージを確認応答すると受信済みのすべてのメッセージが確認応答されるようになった（確認応答されるメッセージの後に受信したメッセージも含む）。</p> <ul style="list-style-type: none"> • ACKNOWLEDGE_PREVIOUS では、古い振る舞いが保持される（確認応答されるメッセージとその前のすべてのメッセージが確認応答される）。 • ACKNOWLEDGE_ALL では、どのメッセージによって確認応答が発生したのかに関係なく、特定のセッションによって受信したすべてのメッセージが確認応答される。 	<p>Administration Console のフィールド表示：[確認応答ポリシー]</p> <p>必須：いいえ</p> <p>デフォルト値：All</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
AllowCloseInOnMessage	<p>接続ファクトリの <i>AllowCloseInOnMessage</i> 属性の値を返す。</p> <p><i>close()</i> メソッドを <i>onMessage()</i> 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。</p> <ul style="list-style-type: none"> このフラグを有効 (true) にした場合、永続的なブロックングではなく、<i>onMessage()</i> メソッド呼び出し内からの <i>close()</i> 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 <i>AUTO_ACKNOWLEDGE</i> に設定した場合、<i>onMessage()</i> が完了しても現在のメッセージは自動的に通知される。 このフラグを選択しない場合 (false)、<i>onMessage()</i> から呼び出されたときに <i>stop()</i> および <i>close()</i> メソッドがハングする。 <p>この属性は動的であり、いつでも変更できる。ただし、この値を変更しても、既存の接続には効果が及ばない。その効果は、この接続ファクトリで作成された新しい接続だけに及ぶ。</p>	<p>Administration Console のフィールド表示 : [メッセージの短縮を許可]</p> <p>必須 : いいえ デフォルト値 : false</p>
ClientId	<p>この接続ファクトリを使用する恒久サブスクライバのクライアント ID を返す。</p>	<p>Administration Console のフィールド表示 : [クライアント ID]</p> <p>必須 : いいえ</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
DefaultDeliveryMode	<p>接続ファクトリの <i>DefaultDeliveryMode</i> 属性の値を返す。</p> <p>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。このファクトリで作成された接続で生成され、<i>DefaultDeliveryMode</i> が null のすべてのメッセージは、この値を受け取る。</p>	<p>Administration Console のフィールド表示 : [デフォルト配信モード]</p> <p>必須 : いいえ</p> <p>デフォルト値 : Persistent</p>
DefaultPriority	<p>接続ファクトリの <i>defaultPriority</i> 属性を返す。</p> <p>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。このファクトリで作成された接続で生成され、<i>DefaultPriority</i> が -1 のすべてのメッセージは、この値を受け取る。</p>	<p>Administration Console のフィールド表示 : [デフォルト優先順位]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 4</p> <p>最小値 : 0</p> <p>最大値 : 9</p>
DefaultRedeliveryDelay	<p>接続ファクトリの <i>defaultRedeliveryDelay</i> 属性の値を返す。</p> <p>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) を定義する。このファクトリで作成されたコンシューマによって消費され、<i>DefaultRedeliveryDelay</i> が -1 のすべてのメッセージは、この値を使用する。</p>	<p>Administration Console のフィールド表示 : [デフォルト再送遅延]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 0</p> <p>最小値 : 0</p> <p>最大値 : 2⁶³-1</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
DefaultTimeToDeliver	<p>接続ファクトリの <i>DefaultTimeToDeliver</i> 属性の値を返す。</p> <p>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。このファクトリで作成されたプロデューサによって生成され、<i>DefaultTimeToDeliver</i> が -1 のすべてのメッセージは、この値を使用する。</p>	<p>Administration Console のフィールド表示 : [デフォルト配信時間]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 0</p> <p>最小値 : 0</p> <p>最大値 : 2⁶³-1</p>
DefaultTimeToLive	<p>接続ファクトリの <i>DefaultTimeToLive</i> 属性の値を返す。</p> <p>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。このファクトリで作成された接続で生成され、<i>DefaultTimeToLive</i> が -1 のすべてのメッセージは、ミリ秒で表されたこの値を受け取る。</p>	<p>Administration Console のフィールド表示 : [デフォルト生存時間]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 0</p> <p>最小値 : 0</p> <p>最大値 : 2⁶³-1</p>
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須 : いいえ</p> <p>デフォルト値 : 1000</p> <p>最小値 : 0</p> <p>最大値 : 2³¹-1</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
FlowControlEnabled	<p>接続ファクトリの <i>FlowControlEnabled</i> 属性の値を返す。</p> <p>接続ファクトリを使用して作成したプロデューサでフロー制御が許可されるかどうかを指定する。true の場合、JMS サーバが Bbytes/Messages ThresholdHigh に達すると、関連するメッセージプロデューサが低速化される。</p>	<p>Administration Console のフィールド表示 : [フロー制御を有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : true</p>
FlowInterval	<p>接続ファクトリの <i>FlowInterval</i> 属性の値を返す。</p> <p>プロデューサが FlowMaximum のメッセージ数から FlowMinimum に、またはその反対にフローを調整するときの調整期間 (単位は秒)。</p> <p>プロデューサがフロー制御されている場合、<i>FlowInterval</i> 秒の間で <i>FlowMaximum</i> から <i>FlowMinimum</i> へと低速化される。</p>	<p>Administration Console のフィールド表示 : [フロー間隔 (秒)]</p> <p>必須 : いいえ</p> <p>単位 : 秒</p> <p>デフォルト値 : 60</p> <p>最小値 : 0</p> <p>最大値 : $2^{31}-1$</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
FlowMaximum	<p>接続ファクトリの <i>FlowMaximum</i> 属性の値を取得する。</p> <p>しきい値の条件に達したプロデューサに対する秒当たりの最大メッセージ数。プロデューサがフロー制御されている場合、秒あたり <i>FlowMaximum</i> のメッセージ数より高速化できない。</p> <p>しきい値の条件に達したときにプロデューサがフローを制御していない場合、そのプロデューサの初期フロー制限が FlowMaximum に設定される。しきい値の条件に達したときにプロデューサが既にフローを制御している場合 (フロー制限は FlowMaximum 未満)、プロデューサは次にフローが評価されるまで現在のフロー制限で処理を継続する。</p> <p>注意：いったん、しきい値条件への接触を回避してからは、プロデューサはフロー制限を無視できなくなる。このフロー制限が FlowMaximum 未満の場合、プロデューサはフローが評価されるたびにそのフローを徐々に FlowMaximum まで増やす必要がある。プロデューサが FlowMaximum に達すると、そのフロー制限を無視し、そのフローを制限せずに送信できる。</p>	<p>Administration Console のフィールド表示 : [最大フロー]</p> <p>必須 : いいえ</p> <p>単位 : メッセージ / 秒</p> <p>デフォルト値 : 500</p> <p>最小値 : 0</p> <p>最大値 : $2^{31}-1$</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
FlowMinimum	<p>接続ファクトリの <i>FlowMinimum</i> 属性の値を取得する。</p> <p>しきい値の条件に達したプロデューサに対する秒当たりの最小メッセージ数。これは、プロデューサのフロー制御の下限値。つまり、WebLogic JMS は、フロー制限が FlowMinimum に達したプロデューサの処理速度はそれ以上落とさない。</p> <p>プロデューサがフロー制御されている場合、秒あたり <i>FlowMaximum</i> のメッセージ数より低速化できない。</p>	<p>Administration Console のフィールド表示 : [最小フロー]</p> <p>必須 : いいえ</p> <p>単位 : メッセージ / 秒</p> <p>デフォルト値 : 50</p> <p>最小値 : 1</p> <p>最大値 : $2^{31}-1$</p>
FlowSteps	<p>接続ファクトリの <i>FlowSteps</i> 属性の値を取得する。</p> <p>プロデューサが FlowMaximum のメッセージ数から FlowMinimum に、またはその反対にフローを調整するとき使用するステップ数。フロー間隔の調整期間は、フローステップ数に分割される (たとえば 60 秒を 6 ステップで割ると 1 ステップ 10 秒となる)。</p> <p>また、FlowMaximum と FlowMinimum の差をステップに分割することにより、移動 (調整率など) が計算される。各フローステップでは、次のように現在の条件に基づいて必要に応じてフローが上方または下方に調整される。</p> <ul style="list-style-type: none"> • 下方移動 (減衰) は指定した期間 (フロー間隔) および指定したステップ数に対し幾何級数的 (たとえば、100、50、25、12.5)。 • 上方移動は線形。差は単純にステップ数で除算される。 	<p>Administration Console のフィールド表示 : [フロー ステップ]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 10</p> <p>最小値 : 1</p> <p>最大値 : $2^{31}-1$</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
JNDIName	JMS 接続ファクトリの JNDI 名を返す。 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ルックアップに使用される名前。接続ファクトリ名は別途コンフィグレーションされる。	Administration Console の フィールド表示 : [JNDI 名] 必須 : いいえ
LoadBalancingEnabled	接続ファクトリの <i>LoadBalancingEnabled</i> 属性の値を返す。 分散送り先の場合は、接続ファクトリで作成された匿名でないプロデューサが、呼び出しごとに分散送り先内でロードバランシングの対象となるかどうかを指定する。 <ul style="list-style-type: none"> • true の場合、関連付けられたメッセージプロデューサのすべての <code>send()</code> または <code>publish()</code> でロードバランシングが実行される。 • false の場合、関連付けられたメッセージプロデューサの最初の <code>send()</code> または <code>publish()</code> でロードバランシングが実行される。 	Administration Console の フィールド表示 : [ロードバランシングを有効化] 必須 : いいえ デフォルト値 : true

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesMaximum	<p>接続ファクトリの <i>MessagesMaximum</i> 属性の値を返す。</p> <p>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。値 0 は無効で、さまざまな例外の送出を引き起こす。</p> <ul style="list-style-type: none"> マルチキャストセッションの場合、新しいメッセージは OverrunPolicy 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。 <p>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、MessagesMaximum 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。</p>	<p>Administration Console のフィールド表示 : [最大メッセージ数]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 10</p> <p>最小値 : -1</p> <p>最大値 : $2^{31}-1$</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
OverrunPolicy	<p>接続ファクトリの <i>OverrunPolicy</i> 属性の値を返す。</p> <p>超過時のポリシーは、マルチキャストメッセージに適用される。未処理のメッセージ数が <i>MessagesMuximum</i> 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。</p> <ul style="list-style-type: none"> • KeepNew 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。 • KeepOld 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。 <p>メッセージの存在期間は受け取り順序によって決まり、<i>JMSTimestamp</i> 値によって定義されるわけではない。</p>	<p>Administration Console のフィールド表示 : [超過時のポリシー]</p> <p>必須 : いいえ</p> <p>デフォルト値 : KeepOld</p>
ServerAffinityEnabled	<p>接続ファクトリの <i>ServerAffinityEnabled</i> 属性の値を返す。</p> <p>分散送り先の場合、分散送り先内の複数の物理的送り先 (キューまたはトピック) に渡ってコンシューマまたはプロデューサをロード バランシングしている WebLogic Server インスタンスにおいて、同じインスタンス内で実行されている他のすべての物理的送り先間でのロード バランシングが最初に試行されるかどうかを指定する。</p>	<p>Administration Console のフィールド表示 : [サーバアフィニティを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : true</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須 : いいえ</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
TransactionTimeout	<p>接続ファクトリの <i>TransactionTimeout</i> 属性の値を返す。</p> <p>この接続ファクトリを使用して作成された接続のすべてのトランザクションはこのタイムアウト値 (単位は秒) を持つ。</p> <p>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。</p>	<p>Administration Console のフィールド表示: [トランザクションタイムアウト]</p> <p>必須: いいえ</p> <p>デフォルト値: 3600</p> <p>最小値: 0</p> <p>最大値: 2³¹-1</p>
UserTransactionsEnabled	<p><i>UserTransactionsEnabled</i> 属性の値を返す。</p> <p>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。true の場合、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。</p> <p>ただし <i>XAConnectionFactoryEnabled</i> フラグが true の場合、XA 接続ファクトリを返すために <i>UserTransactionsEnabled</i> 属性も true である必要がある。</p> <p>注意: トランザクションセッションは、設定にかかわらず独自の内部トランザクションのために現在のスレッド 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適用される。</p>	<p>Administration Console のフィールド表示: [ユーザ トランザクションを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p>

表 24-1 JMSConnectionFactory の属性 (続き)

属性	説明	指定できる値とデフォルト値
XAConnectionFactoryEnabled	<p>接続ファクトリの <i>XAConnectionFactoryEnabled</i> 属性の値を返す。</p> <p>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。 <i>getXAResource</i> 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。</p> <p>XA 接続ファクトリが返されるようにするためには、 <i>UserTransactionsEnabled</i> フラグも true に設定する必要がある (このフラグはデフォルトで false のため)。</p>	<p>Administration Console のフィールド表示 : [XA コネクションファクトリを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>
XAServerEnabled	<p>接続ファクトリの <i>XAServerEnabled</i> 属性の値を返す。</p> <p>WebLogic Server インスタンス内部から当該接続ファクトリが呼び出されたときに、XA 対応の JMS 接続およびセッションが常に作成されるかどうかを定義する。</p>	<p>Administration Console のフィールド表示 : [サーバサイド XA を有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>

JMSDestinationKey

説明

構文

```
<JMSDestinationKey
  Direction=( "Ascending" | "Descending" )
  KeyType=( "Boolean" | "Byte" | "Short" | "Int" | "Long" | "Float" | "Double"
| "String" )
  Name="String"
  Notes="String"
  Property="String"
/>
```

親要素

- Domain

属性

表 25-1 JMSDestinationKey の属性

属性	説明	指定できる値とデフォルト値
Direction	<p>このキーに対するソート順 (Ascending または Descending) を返す。</p> <p>JMSMessageID に Ascending を選択すると、ソート順は FIFO (先入れ先出し) になる (送り先のデフォルト)。ソート順を LIFO (後入れ先出し) にするには、値を Descending に設定する。</p> <p>この属性は動的にコンフィグレーションできない。</p>	<p>Administration Console のフィールド表示: [ソート順]</p> <p>必須: いいえ</p> <p>デフォルト値: Ascending</p>
KeyType	<p>このキーのプロパティタイプを返す。</p>	<p>Administration Console のフィールド表示: [キータイプ]</p> <p>必須: いいえ</p> <p>デフォルト値: String</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>

表 25-1 JMSDestinationKey の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ
Property	<p>このキーのプロパティ名を返す。</p> <p>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。</p> <p>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。</p> <p>値の範囲 :</p> <p>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とおり。</p> <ul style="list-style-type: none"> • JMSMessageID • JMSTimestamp • JMSCorrelationID • JMSPriority • JMSExpiration • JMSType • JMSRedelivered • JMSDeliveryTime <p>この属性は動的にコンフィグレーションできない。</p>	Administration Console の フィールド表示 : [プロパティ]

JMSDistributedQueue

説明

このクラスは、JMS 分散キューを表します。

構文

```
<JMSDistributedQueue
  DeploymentOrder="number"
  ForwardDelay="number"
  JNDIName="String"
  LoadBalancingPolicy=( "Round-Robin" | "Random" )
  Name="String"
  Notes="String"
  Targets="list of Target names"
/>
```

親要素

- Domain

子要素

- JMSTemplate

属性

表 26-1 JMSDistributedQueue の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ デフォルト値: 1000 最小値: 0 最大値: $2^{31}-1$</p>
ForwardDelay	<p>この分散キューのメッセージ転送遅延を返します。</p> <p>メッセージを持つがコンシューマを持たない分散キューメンバーが、コンシューマを持つ他のキューメンバーにそのメッセージを転送するまでの秒数を定義する。</p> <p>値 -1 は、メッセージが他のキューメンバーに転送されないことを示す。</p>	<p>Administration Console のフィールド表示: [転送の遅延 (秒)]</p> <p>必須: いいえ デフォルト値: -1</p>

表 26-1 JMSDistributedQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
JNDIName	分散送り先の JNDI 名を設定する。 この値は、JNDI ネームスペース内で分散送り先のルックアップに使用される名前。アプリケーションは、JNDI 名を使用して分散送り先をルックアップできる。この属性を指定しない場合、JNDI ネームスペースに送り先がバインドされない。	Administration Console のフィールド表示 : [JNDI 名] 必須 : いいえ
LoadBalancingPolicy	この分散送り先に送信するプロデューサに対するロード バランス ポリシーを返す。 プロデューサが分散送り先のメンバー間でメッセージ負荷のバランスを取るために Round-Robin または Random のいずれのポリシーを使用するかを指定する。	Administration Console のフィールド表示 : [ロード バランス ポリシー] 必須 : いいえ デフォルト値 : Round-Robin
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示 : [メモ] 必須 : いいえ
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須 : いいえ

JMSDistributedTopic

説明

この MBean は、JMS 分散トピックを表します。

構文

```
<JMSDistributedTopic
  DeploymentOrder="number"
  JNDIName="String"
  LoadBalancingPolicy=( "Round-Robin" | "Random" )
  Name="String"
  Notes="String"
  Targets="list of Target names"
/>
```

親要素

- Domain

子要素

- JMSTemplate

属性

表 27-1 JMSDistributedTopic の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
JNDIName	<p>分散送り先の JNDI 名を返す。</p> <p>この値は、JNDI ネームスペース内で分散送り先のルックアップに使用される名前。アプリケーションは、JNDI 名を使用して分散送り先をルックアップできる。この属性を指定しない場合、JNDI ネームスペースに送り先がバインドされない。</p>	<p>Administration Console のフィールド表示: [JNDI 名]</p> <p>必須: いいえ</p>

表 27-1 JMSDistributedTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
LoadBalancingPolicy	この分散送り先に送信するプロデューサに対するロード バランス ポリシーを返す。 プロデューサが分散送り先のメンバー間でメッセージ負荷のバランスを取るために Round-Robin または Random のいずれのポリシーを使用するかを指定する。	Administration Console のフィールド表示:[ロード バランス ポリシー] 必須: いいえ デフォルト値: Round-Robin
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示:[名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示:[メモ] 必須: いいえ
Targets	このデプロイメントの対象を設定する。対象はクラスタかサーバでなければならない。	必須: いいえ

JMSFileStore

説明

このクラスは、ファイルベースの JMS 永続ストアを表します。

構文

```
<JMSFileStore
  Directory="String"
  Name="String"
  Notes="String"
  SynchronousWritePolicy=( "Disabled" | "Cache-Flush" | "Direct-Write" )
/>
```

親要素

- Domain

属性

表 28-1 JMSFileStore の属性

属性	説明	指定できる値とデフォルト値
Directory	ファイルベースの永続ストアが存在するディレクトリを定義する。 この属性は動的にコンフィグレーションできない。	Administration Console の フィールド表示:[ディレクトリ]
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示:[名前] 必須:いいえ

表 28-1 JMSFileStore の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示:[メモ] 必須:いいえ

表 28-1 JMSFileStore の属性 (続き)

属性	説明	指定できる値とデフォルト値
SynchronousWrite Policy	<p>WebLogic JMS ファイル ストアは、3つの同期書き込みポリシー (Disabled、Cache-Flush、および Direct-Write) を提供します。これらのポリシーは、ファイル ストアがデータをディスクに書き込む方法、およびファイル ストアのパフォーマンス、スケーラビリティ、および信頼性に対する影響をコンフィグレーションします。</p> <p>Disabled-- ファイル ストアの書き込みにおいて、オペレーティング システムのキャッシュとファイル システムのオンディスク キャッシュの両方を使用できる。もっとも高速なポリシーだが、信頼性をもっとも劣る。他のポリシーに比べて 100 倍以上高速だが、停電やオペレーティング システムの障害によってメッセージが消失したり重複したりするおそれがある。</p> <p>Cache-Flush-- デフォルトポリシー。すべての書き込みがディスクにフラッシュされるまでトランザクションは完了しない。このポリシーは信頼性とスケーラビリティが高く、同時に処理できるユーザ数も増える。</p> <p>Direct-Write-- ファイル ストアの書き込みが、直接ディスクに書き込まれる。このポリシーは、Solaris、HP、および Windows ではサポートされている。ただし、オンディスク キャッシュをサポートしていないプラットフォームでこのポリシーを設定した場合は、自動的に Cache-Flush ポリシーが適用される。</p> <p>Direct-Write ポリシーの信頼性とパフォーマンスは、次のように、オペレーティング システムおよびハードウェアでオンディスク キャッシュがサポートされているかどうかによって異なる。</p> <ul style="list-style-type: none"> • オンディスク キャッシュを有効にすると、Cache-Flush ポリシーに比べて 2 ~ 5 倍高速になる。ただし、非常に高いスケーラビリティが要求される場合は、多少遅くなるおそれがある。 • オンディスク キャッシュを無効にすると、1 対多の場合は Cache-Flush ポリシーより速いが、それ以外の場合は Cache-Flush ポリシーより遅くなる。 • Direct-Write ポリシーでは、オンディスク キャッシュを有効にするとスケーラビリティが向上し、無効にすると向上しない (Solaris では直接書き込みのオンディスク キャッシュを有効にできない)。 <p>警告 : Solaris および HP とは異なり、Windows 上で Direct-Write ポリシーを使用する場合、トランザクション データはオンディスク キャッシュに残され、すぐにはディスクに書き込まれない。この場合、停電などによってオンディスク キャッシュのデータが消えてメッセージが消失したり重複したりするおそれがあり、トランザクションとしては安全とはいえない。Windows で Direct-Write の信頼性を高めるには、ディスクの書き込みキャッシュをすべて無効にするか、バッテリー バックアップのあるキャッシュを使用する。</p> <p>Windows でハードディスクのオンディスク キャッシュを無効にする手順は次のとおり。</p> <p>[スタート 設定 コントロール パネル システム] を選択し、[ハードウェア] タブで [デバイスマネージャ] ボタンをクリックして [ディスク ドライブ] アイコンをダブルクリックする。次に、ドライブの名前をダブルクリックし、[ディスクのプロパティ] タブの [書き込みキャッシュを有効にする] チェック ボックスのチェックをはずす。この値を変更できないファイル システムもある (たとえば、信頼性のあるキャッシュを備えた RAID システム)。</p> <p>注意 : JMS ファイル ストアが非永続的なメッセージのディスクへのページングのために排他的に使用されている場合、同期書き込みポリシーは無視される。</p>	<p>Administration Console の フィールド表示 : [同期書き込みポリシー]</p> <p>必須 : いいえ</p> <p>デフォルト値 : Cache-Flush</p>

JMSJDBCStore

説明

この MBean は、JDBC ベースの JMS 永続ストアを表します。

構文

```
<JMSJDBCStore
  ConnectionPool="JDBCConnectionPool name"
  Name="String"
  Notes="String"
  PrefixName="String"
/>
```

親要素

- Domain

属性

表 29-1 JMSJDBCStore の属性

属性	説明	指定できる値とデフォルト値
ConnectionPool	この永続ストア用の JDBC 接続プールを取得する。 この属性は動的にコンフィグレーションできない。	Administration Console の フィールド表示: [接続プール] 必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示: [メモ] 必須: いいえ
PrefixName	JDBC 永続ストア内の JMS テーブルのプレフィックス名を取得する。 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。 <ul style="list-style-type: none"> データベース管理システム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。 [schema.[catalog.]]prefix	Administration Console の フィールド表示: [プレフィックス名] 必須: いいえ

JMSQueue

説明

構文

```
<JMSQueue
  BytesMaximum="number"
  BytesPagingEnabled=( "default" | "false" | "true" )
  BytesThresholdHigh="number"
  BytesThresholdLow="number"
  DeliveryModeOverride=( "Persistent" | "Non-Persistent" | "No-Delivery" )
  DestinationKeys="list of JMSDestinationKey names"
  ErrorDestination="JMSDestination name"
  JNDIName="String"
  JNDINameReplicated=( "true" | "false" )
  MessagesMaximum="number"
  MessagesPagingEnabled=( "default" | "false" | "true" )
  MessagesThresholdHigh="number"
  MessagesThresholdLow="number"
  Name="String"
  Notes="String"
  PriorityOverride="number"
  RedeliveryDelayOverride="number"
  RedeliveryLimit="number"
  StoreEnabled=( "default" | "false" | "true" )
  Template="JMSTemplate name"
```

JMSQueue

```
    TimeToDeliverOverride="String"  
    TimeToLiveOverride="number"  
/>
```

親要素

- JMSServer

属性

表 30-1 JMSQueue の属性

属性	説明	指定できる値とデフォルト値
BytesMaximum	<p>BytesMaximum の値を返す。</p> <p>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。</p> <p>注意：分散送り先メンバーに JMS テンプレートが使用される場合、この設定は分散送り先セット全体ではなくその特定のメンバーのみに適用される。</p> <p>値の範囲：>= BytesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示：[最大バイト数]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesPagingEnabled	<p>「true」、「false」、「default」のいずれかを返す。</p> <p>送り先でバイト ページングを有効にするかどうかを指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグを選択しない場合 (false)、この送り先のバイト ページングは明示的に無効化される。 このフラグが選択され (true)、JMS サーバ用のページング ストアがコンフィグレーションされており、かつ BytesThresholdLow と BytesThresholdHigh の属性値がどちらも -1 より大きい場合、この送り先のバイト ページングは有効化される。 この属性が「default」に設定されている場合、この値は JMS テンプレートの値を継承する (JMS テンプレートが指定されている場合)。送り先の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、デフォルト値は false と同じになる。 	<p>Administration Console のフィールド表示: [バイト ページングを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: default</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdHigh	<p>BytesThresholdHigh の値を返す。</p> <p>送り先に保存されているバイト数に基づいてイベントを発生させる上限しきい値。バイト数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている (およびページングストアがコンフィグレーションされている) 場合、送り先レベルのバイト ページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、送り先に対するバイト ページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : <= BytesMaximum; >BytesThresholdLow</p> <p>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。</p> <p>注意 : バイト ページングが有効になっている場合、BytesThresholdHigh を -1 にリセットすることでそれを動的に無効にすることはできない (例外が送出される)。ページングを動的に無効にするには、BytesThresholdHigh を非常に大きな値に設定して、ページングがトリガされないようにする。</p>	<p>Administration Console のフィールド表示 : [最大バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdLow	<p>BytesThresholdLow の値を返す。</p> <p>送り先に保存されているバイト数に基づいてイベントを発生させる下限しきい値。バイト数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている場合、ページングが停止する (ページングが実行されている場合) 。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値の範囲 : < BytesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [最小バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>セキュアな値 : null</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
DeliveryModeOverride	<p>DeliveryModeOverride 値を返す。</p> <p>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。</p> <p>No-Delivery は、配信モードがオーバーライドされないことを示す。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [配信モードオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : No-Delivery</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
DestinationKeys	テンプレートまたは送り先の送り先キーの読み取り専用配列を返す。	Administration Console のフィールド表示 : [送り先キー] 必須 : いいえ
ErrorDestination	再配信の上限に達したメッセージのエラー送り先の値を設定する。 再配信の上限に達したメッセージの送り先の名前を定義する。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、このメッセージはそのまま削除される。 注意 : エラー送り先は、ローカル JMS サーバ上でコンフィグレーションされた送り先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。	Administration Console のフィールド表示 : [エラー送り先] 必須 : いいえ セキュアな値 : null
JNDIName	送り先の JNDI 名を返す。 この値は、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。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通知されないため、送り先名をルックアップおよび使用できない。 この属性は動的にコンフィグレーションできない。	Administration Console のフィールド表示 : [JNDI 名] 必須 : いいえ
JNDINameReplicated	JNDINameReplicated 属性を true に設定すると、送り先の JNDI 名 (存在する場合) はクラスタ全体にレプリケートされる。JNDINameReplicated 属性を false に設定すると、送り先の JNDI 名 (存在する場合) は、その送り先が一部になっているサーバから参照できるだけになる。	必須 : いいえ デフォルト値 : true

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesMaximum	<p>MessagesMaximum の値を返す。</p> <p>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。</p> <p>注意：分散送り先メンバーに JMS テンプレートが使用される場合、この設定は分散送り先セット全体ではなくその特定のメンバーのみに適用される。</p> <p>値の範囲：>= MessagesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示：[最大メッセージ数]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesPagingEnabled	<p>「true」、「false」、「default」のいずれかを返す。</p> <p>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指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグを選択しない場合 (false)、この送り先のメッセージ ページングは明示的に無効化される。 このフラグが選択され (true)、JMS サーバ用のページング ストアがコンフィグレーションされており、かつ MessagesThresholdLow と MessagesThresholdHigh の属性値がどちらも -1 より大きい場合、送り先のメッセージ ページングは有効化される。 この属性が「default」に設定されている場合、この値は JMS テンプレートの値を継承する (JMS テンプレートが指定されている場合)。送り先の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、デフォルト値は false と同じになる。 	<p>Administration Console のフィールド表示: [メッセージ ページングを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: default</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdHigh	<p>MessagesThresholdHigh の値を返す。</p> <p>送り先に保存されているメッセージ数に基づいてイベントを発生させる上限しきい値。メッセージ数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログ メッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>メッセージ ページング - メッセージ ページングが有効化されている (およびページング ストアがコンフィグレーションされている) 場合、送り先レベルのメッセージ ページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、送り先に対するメッセージ ページング、フロー制御、およびしきい値ログ メッセージが無効になる。</p> <p>値の範囲 : <= MessagesMaximum; >MessagesThresholdLow</p> <p>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。</p> <p>注意 : メッセージ ページングが有効になっている場合、MessagesThresholdHigh を -1 にリセットすることでそれを動的に無効にすることはできない (例外が送出される)。ページングを動的に無効にするには、MessagesThresholdHigh を非常に大きな値に設定して、ページングがトリガされないようにする。</p>	<p>Administration Console のフィールド表示 : [最大メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdLow	<p>MessagesThresholdLow の値を返す。</p> <p>送り先に保存されているメッセージ数に基づいてイベントを発生させる下限しきい値。メッセージ数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>メッセージページング - メッセージページングが有効化されている場合、ページングが停止する (ページングが実行されている場合) 。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値の範囲 : < MessagesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [最小メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
PriorityOverride	<p>PriorityOverride 値を返す。</p> <p>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係なく、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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示: [優先順位オーバーライド]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: 9</p>
RedeliveryDelayOverride	<p>ロールバックおよび回復されたメッセージが再配信されるまでの再配信遅延の値 (単位はミリ秒) を返す。</p> <p>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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示: [再配信遅延のオーバーライド]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: 2⁶³-1</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
RedeliveryLimit	<p>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を返す。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、再配信の制限に達したときに以下のことが発生する。</p> <ul style="list-style-type: none"> エラー送り先がコンフィグレーションされているか、またはエラー送り先の割り当てを超過する場合、永続および非永続メッセージは削除される。 エラー送り先がコンフィグレーションされており、エラー送り先の割り当てを超過していない場合、エラーメッセージのログが記録され、メッセージが削除される。しかし、メッセージが永続的である場合、永続ストアに保持される。これにより、永続メッセージは WebLogic Server の再起動時に再配信される。 <p>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。</p> <p>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。</p> <p>注意： RedeliveryLimit の変更は、すでにこの制限に達しているメッセージには適用されない。このようなメッセージが次に再配信される場合、それらは即座にエラー送り先にリダイレクトされる。メッセージが再配信された回数は永続しない。つまり、再起動後は各メッセージの配信試行回数がゼロにリセットされる。</p>	<p>Administration Console のフィールド表示：[再配信の制限]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
StoreEnabled	<p>「true」、「false」、「default」のいずれかを返す。</p> <p>JMS サーバによって指定された永続ストアを送り先が使用するかどうかを指定するフラグ。</p> <ul style="list-style-type: none"> このフラグを true に設定したが、JMS サーバ用の永続ストアが定義されていない場合、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 false に設定した場合、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 default に設定した場合、JMS サーバ用の永続ストアが定義されていれば、送り先はその永続ストアを使用し、永続メッセージをサポートする。 <p>この属性は動的にコンフィグレーションできない。</p>	<p>Administration Console のフィールド表示:[ストアを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: default</p>
Template	<p>送り先の派生元の JMS テンプレートを返す。</p> <p>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。送り先ごとの Template 属性の設定は静的。ただし、動的に変更できる。</p> <p>注意: デフォルト値に設定された属性は、実行時の JMS テンプレートから送り先の値を継承する。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。</p>	<p>Administration Console のフィールド表示:[テンプレート]</p> <p>必須: いいえ</p>

表 30-1 JMSQueue の属性 (続き)

属性	説明	指定できる値とデフォルト値
TimeToDeliverOverride	<p>TimeToDeliver オーバーライド値 (ミリ秒単位またはスケジュール) を返す。</p> <p>プロデューサおよび接続ファクトリによって指定された配信時間とは関係なく、メッセージが生成されてからその対象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。</p> <p>デフォルト値 (-1) は、送り先が配信時間の設定をオーバーライドしないことを示す。TimeToDeliverOverride は、長さまたはスケジュールとして指定できる。</p> <p>注意：TimeToDeliverOverride の変更は将来のメッセージ配信のみに適用され、すでに生成されたメッセージには適用されない。</p>	<p>Administration Console のフィールド表示 : [配信時間のオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p>
TimeToLiveOverride	<p>TimeToLiveOverride 値を返す。</p> <p>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係なく、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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [生存時間のオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

JMSServer

説明

この MBean は、JMS サーバを定義します。

構文

```
<JMSServer
  BytesMaximum="number"
  BytesPagingEnabled=( "true" | "false" )
  BytesThresholdHigh="number"
  BytesThresholdLow="number"
  DeploymentOrder="number"
  MessagesMaximum="number"
  MessagesPagingEnabled=( "true" | "false" )
  MessagesThresholdHigh="number"
  MessagesThresholdLow="number"
  Name="String"
  Notes="String"
  PagingStore="JMSStore name"
  Store="JMSStore name"
  Targets="list of Target names"
  TemporaryTemplate="JMSTemplate name"
/>
```

親要素

- Domain

子要素

- JMSQueue
- JMSTopic
- JMSSessionPool

属性

表 31-1 JMS Server の属性

属性	説明	指定できる値とデフォルト値
BytesMaximum	<p>BytesMaximum の値を返す。</p> <p>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。</p> <p>値の範囲 : \geq BytesThresholdHigh</p> <p>この属性は動的にコンフィグレーションできる。</p>	<p>Administration Console のフィールド表示 : [最大バイト数]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : $2^{63}-1$</p>

表 31-1 JMSServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesPagingEnabled	<p>BytesPagingEnabled が設定されているかどうかをチェックする。</p> <p>JMS サーバでバイト ページングを有効にするかどうかを指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグを選択しない場合 (false)、サーバのバイト ページングは明示的に無効化される。 このフラグが選択され (true)、ページングストアがコンフィグレーションされており、かつ BytesThresholdLow と BytesThresholdHigh の属性値がどちらも -1 より大きい場合、サーバのバイト ページングは有効化される。 BytesThresholdLow 属性または BytesThreshold High 属性のいずれかが未定義であるか、-1 に設定されている場合、このフラグが選択されていてもサーバのバイト ページングは暗黙的に無効化される。 	<p>Administration Console のフィールド表示: [バイト ページングを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p>

表 31-1 JMS Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdHigh	<p>BytesThresholdHigh の値を返す。</p> <p>JMS サーバに保存されているバイト数に基づいてイベントを発生させる上限しきい値。バイト数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている (およびページングストアがコンフィグレーションされている) 場合、サーバのバイト ページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、JMS サーバが防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、JMS サーバに対するサーバのバイト ページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : <= BytesMaximum; >BytesThresholdLow</p> <p>この属性は動的にコンフィグレーションできる。</p> <p>注意 : バイト ページングは、BytesThresholdHigh を -1 に設定することによって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、BytesThresholdHigh を大きい値に設定してページングが発生しないようにする。</p>	<p>Administration Console のフィールド表示 : [最大バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 31-1 JMSServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdLow	<p>BytesThresholdLow の値を返す。</p> <p>JMS サーバに保存されているバイト数に基づいてイベントを発生させる下限しきい値。バイト数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている場合、ページングが停止する (ページングが実行されている場合) 。</p> <p>フロー制御 - フロー制御が有効化されている場合、JMS サーバが防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値 -1 を指定すると、JMS サーバに対するサーバのバイト ページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : < BytesThresholdHigh</p> <p>この属性は動的にコンフィグレーションできる。</p>	<p>Administration Console のフィールド表示 : [最小バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 31-1 JMS Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
MessagesMaximum	<p>MessagesMaximum の値を返す。</p> <p>JMS サーバに保存可能な最大メッセージ数。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。</p> <p>この属性は動的にコンフィグレーションできる。</p> <p>値の範囲: \geq MessagesThresholdHigh</p>	<p>Administration Console のフィールド表示: [最大メッセージ数]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: $2^{63}-1$</p>

表 31-1 JMSServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesPagingEnabled	<p>MessagesPagingEnabled が設定されているかどうかをチェックする。</p> <p>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指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグを選択しない場合 (false)、サーバのメッセージ ページングは明示的に無効化される。 このフラグが選択され (true)、ページングストアがコンフィグレーションされており、かつ MessagesThresholdLow と MessagesThresholdHigh の属性値がどちらも -1 より大きい場合、サーバのメッセージ ページングは有効化される。 MessagesThresholdLow 属性または MessagesThreshold High 属性のいずれかが未定義であるか、-1 に設定されている場合、このフラグが選択されていてもサーバのメッセージ ページングは暗黙的に無効化される。 	<p>Administration Console のフィールド表示: [メッセージ ページングを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p>

表 31-1 JMS Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdHigh	<p>MessagesThresholdHigh の値を返す。</p> <p>JMS サーバに保存されているメッセージ数に基づいてイベントを発生させる上限しきい値。メッセージ数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>メッセージページング - メッセージページングが有効化されている (およびページングストアがコンフィグレーションされている) 場合、サーバメッセージページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、JMS サーバが防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、JMS サーバに対するサーバのメッセージページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : <= MessagesMaximum; >MessagesThresholdLow</p> <p>この属性は動的にコンフィグレーションできる。</p> <p>注意 : メッセージページングは、MessagesThresholdHigh を -1 に設定することによって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、MessagesThresholdHigh を大きい値に設定してページングが発生しないようにする。</p>	<p>Administration Console のフィールド表示 : [最大メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 31-1 JMSServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdLow	<p>MessagesThresholdLow の値を返す。</p> <p>JMS サーバに保存されているメッセージ数に基づいてイベントを発生させる下限しきい値。メッセージ数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>メッセージページング - メッセージページングが有効化されている場合、ページングが停止する (ページングが実行されている場合)。</p> <p>フロー制御 - フロー制御が有効化されている場合、JMS サーバが防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値 -1 を指定すると、JMS サーバに対するサーバのメッセージページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : < MessagesThresholdHigh</p> <p>この属性は動的にコンフィグレーションできる。</p>	<p>Administration Console のフィールド表示 : [最小メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>

表 31-1 JMS Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
PagingStore	<p>JMS サーバ用のページングストアを取得する。</p> <p>非永続メッセージをページングする、JMS サーバ用の永続ストアの名前。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用と同じストアであってはならない。</p> <p>ページングストアは、1つのJMSサーバによってのみ使用される。値が指定されない場合、メッセージページングはサポートされない。ページングストアが指定されない場合は、サーバでも送り先でもメッセージページングはサポートされない。</p> <p>注意：JDBCストアはページングストアとして使用しないほうがよい。トラフィックが多いことと以後のパフォーマンスが低下することから、そうしたコンフィグレーションは望ましくない。</p>	<p>Administration Console のフィールド表示：[ページングストア]</p> <p>必須：いいえ</p>
Store	<p>JMS サーバ用の永続ストアを取得する。</p> <p>永続ストアは、1つのJMSサーバによってのみ使用される。値が指定されない場合、永続メッセージングはサポートされない。永続ストアが指定されない場合は、JMS サーバ上の送り先で永続メッセージも恒久サブスクライバもサポートされない。</p>	<p>Administration Console のフィールド表示：[ストア]</p> <p>必須：いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>Administration Console のフィールド表示：[対象]</p> <p>必須：いいえ</p>

表 31-1 JMSServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
TemporaryTemplate	<p>JMS サーバの一時的なテンプレートを返す。</p> <p>一時的キューおよび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される既存の JMS テンプレートの名前。一時的送り先の属性値は 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、Store 属性の値は無視される。</p> <p>注意：この属性が設定されていない場合、一時的な送り先 (キューまたはトピック) の作成は失敗する。</p>	<p>Administration Console のフィールド表示 : [一時的なテンプレート]</p> <p>必須 : いいえ</p>

JMSSessionPool

説明

この MBean は、JMS セッション プールを表します。

構文

```
<JMSSessionPool
  AcknowledgeMode=( "Auto" | "Client" | "Dups-Ok" | "None" )
  ConnectionFactory="String"
  ListenerClass="String"
  Name="String"
  Notes="String"
  SessionsMaximum="number"
  Transacted=( "true" | "false" )
/>
```

親要素

- JMS Server

属性

表 32-1 JMSSESSIONPOOL の属性

属性	説明	指定できる値とデフォルト値
AcknowledgeMode	<p>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確認応答モードを返す。</p> <p>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確認応答されるので、このフィールドは無視される。</p> <p>この属性は動的にコンフィグレーションできない。</p>	<p>Administration Console のフィールド表示: [確認応答モード]</p> <p>必須: いいえ</p> <p>デフォルト値: Auto</p>
ConnectionFactory	<p>セッションプール用の接続ファクトリの JNDI 名を返す。</p>	<p>Administration Console のフィールド表示: [接続ファクトリ]</p> <p>必須: いいえ</p>
ListenerClass	<p>セッションプール用のリスナクラスの名前を返す。</p>	<p>Administration Console のフィールド表示: [リスナクラス]</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>

表 32-1 JMSSessionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
SessionsMaximum	<p>セッション プール用のセッションの最大数を返す。</p> <p>この値は、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。</p> <p>この属性は動的にコンフィグレーションできるが、セッション プールを再起動しないと有効にならない。</p>	<p>Administration Console のフィールド表示 : [最大セッション]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : $2^{63}-1$</p>
Transacted	<p>セッション プールがトランザクション処理されるかどうかをチェックする。</p> <p>このフラグは、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。</p> <p>この属性は動的にコンフィグレーションできない。</p>	<p>Administration Console のフィールド表示 : [トランザクション化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>

JMSStore

説明

JMS 永続ストア。これは永続的データを格納するための物理的なりポジトリです。

構文

```
<JMSStore  
  Name="String"  
  Notes="String"  
>
```

親要素

- Domain

属性

表 33-1 JMSStore の属性

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示:[名前] 必須:いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須:いいえ

JMSTemplate

説明

JMS 送り先テンプレート。

構文

```
<JMSTemplate
  BytesMaximum="number"
  BytesPagingEnabled=( "true" | "false" )
  BytesThresholdHigh="number"
  BytesThresholdLow="number"
  DeliveryModeOverride=( "Persistent" | "Non-Persistent" | "No-Delivery" )
  DestinationKeys="list of JMSDestinationKey names"
  ErrorDestination="JMSDestination name"
  MessagesMaximum="number"
  MessagesPagingEnabled=( "true" | "false" )
  MessagesThresholdHigh="number"
  MessagesThresholdLow="number"
  Name="String"
  Notes="String"
  PriorityOverride="number"
  RedeliveryDelayOverride="number"
  RedeliveryLimit="number"
  TimeToDeliverOverride="String"
```

```
    TimeToLiveOverride="number"  
/>
```

親要素

- Domain
- JMSDistributedQueue
- JMSDistributedTopic

属性

表 34-1 JMSTemplate の属性

属性	説明	指定できる値とデフォルト値
BytesMaximum	<p>BytesMaximum の値を返す。</p> <p>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。</p> <p>注意：分散送り先メンバーに JMS テンプレートが使用される場合、この設定は分散送り先セット全体ではなくその特定のメンバーのみに適用される。</p> <p>値の範囲：>= BytesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示：[最大バイト数]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesPagingEnabled	<p>BytesPagingEnabled が設定されているかどうかをチェックする。</p> <p>テンプレートでバイト ページングを有効にするかどうかを指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグが選択されていない場合 (false)、送り先の設定によってテンプレートがオーバーライドされない限り、テンプレートの送り先のバイト ページングは無効化される。 このフラグが選択され (true)、JMS サーバ用のページング ストアがコンフィグレーションされており、かつ BytesThresholdLow と BytesThresholdHigh の属性値がどちらも -1 より大きい場合、送り先の設定によってテンプレートがオーバーライドされない限り、テンプレートの送り先のバイト ページングは有効化される。 <p>注意：値が定義されていない場合、この設定はデフォルトで「false」となり、送り先の設定によってテンプレートがオーバーライドされない限りテンプレートの送り先のバイト ページングは無効化される。</p>	<p>Administration Console のフィールド表示:[バイト ページングを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdHigh	<p>BytesThresholdHigh の値を返す。</p> <p>送り先に保存されているバイト数に基づいてイベントを発生させる上限しきい値。バイト数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている (およびページングストアがコンフィグレーションされている) 場合、送り先レベルのバイト ページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、送り先に対するバイト ページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : <= BytesMaximum; >BytesThresholdLow</p> <p>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。</p> <p>注意 : バイト ページングが有効になっている場合、BytesThresholdHigh を -1 にリセットすることでそれを動的に無効にすることはできない (例外が送出される)。ページングを動的に無効にするには、BytesThresholdHigh を非常に大きな値に設定して、ページングがトリガされないようにする。</p>	<p>Administration Console のフィールド表示 : [最大バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdLow	<p>BytesThresholdLow の値を返す。</p> <p>送り先に保存されているバイト数に基づいてイベントを発生させる下限しきい値。バイト数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている場合、ページングが停止する (ページングが実行されている場合) 。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値の範囲 : < BytesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [最小バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
DeliveryModeOverride	<p>DeliveryModeOverride 値を返す。</p> <p>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。</p> <p>No-Delivery は、配信モードがオーバーライドされないことを示す。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [配信モードオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : No-Delivery</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
DestinationKeys	<p>テンプレートまたは送り先の送り先キーの読み取り専用配列を返す。</p>	<p>Administration Console のフィールド表示:[送り先キー]</p> <p>必須: いいえ</p>
ErrorDestination	<p>再配信の上限に達したメッセージのエラー送り先の値を設定する。</p> <p>再配信の上限に達したメッセージの送り先の名前を定義する。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、このメッセージはそのまま削除される。</p> <p>注意: エラー送り先は、ローカル JMS サーバ上でコンフィグレーションされた送り先でなければならない。</p> <p>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。</p>	<p>Administration Console のフィールド表示:[エラー送り先]</p> <p>必須: いいえ</p> <p>セキュアな値: null</p>
MessagesMaximum	<p>MessagesMaximum の値を返す。</p> <p>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。</p> <p>注意: 分散送り先メンバーに JMS テンプレートが使用される場合、この設定は分散送り先セット全体ではなくその特定のメンバーのみに適用される。</p> <p>値の範囲: \geq MessagesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示:[最大メッセージ数]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: $2^{63}-1$</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesPagingEnabled	<p>MessagesPagingEnabled が設定されているかどうかをチェックする。 テンプレートでメッセージ ページングを有効にするかどうかを指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグが選択されていない場合 (false)、送り先の設定によってテンプレートがオーバーライドされない限り、テンプレートの送り先のメッセージ ページングは無効化される。 このフラグが選択され (true)、JMS サーバ用のページング ストアがコンフィグレーションされており、かつ MessagesThresholdLow と MessagesThresholdHigh の属性値がどちらも -1 より大きい場合、送り先の設定によってテンプレートがオーバーライドされない限り、テンプレートの送り先のメッセージ ページングは有効化される。 <p>注意：値が定義されていない場合、この設定はデフォルトで「false」となり、送り先の設定によってテンプレートがオーバーライドされない限りテンプレートの送り先のメッセージ ページングは無効化される。</p>	<p>Administration Console のフィールド表示：[メッセージ ページングを有効化]</p> <p>必須：いいえ</p> <p>デフォルト値：false</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdHigh	<p>MessagesThresholdHigh の値を返す。</p> <p>送り先に保存されているメッセージ数に基づいてイベントを発生させる上限しきい値。メッセージ数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>メッセージペー징ング - メッセージペー징ングが有効化されている (およびペー징ングストアがコンフィグレーションされている) 場合、送り先レベルのメッセージペー징ングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、送り先に対するメッセージペー징ング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : <= MessagesMaximum; >MessagesThresholdLow</p> <p>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。</p> <p>注意 : メッセージペー징ングが有効になっている場合、MessagesThresholdHigh を -1 にリセットすることでそれを動的に無効にすることはできない (例外が送出される)。ペー징ングを動的に無効にするには、MessagesThresholdHigh を非常に大きな値に設定して、ペー징ングがトリガされないようにする。</p>	<p>Administration Console のフィールド表示 : [最大メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdLow w	<p>MessagesThresholdLow の値を返す。</p> <p>送り先に保存されているメッセージ数に基づいてイベントを発生させる下限しきい値。メッセージ数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>メッセージページング - メッセージページングが有効化されている場合、ページングが停止する (ページングが実行されている場合)。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値の範囲 : < MessagesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [最小メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
PriorityOverride	<p>PriorityOverride 値を返す。</p> <p>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係なく、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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示: [優先順位オーバーライド]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: 9</p>
RedeliveryDelayOverride	<p>ロールバックおよび回復されたメッセージが再配信されるまでの再配信遅延の値 (単位はミリ秒) を返す。</p> <p>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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示: [再配信遅延のオーバーライド]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: 2⁶³-1</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
RedeliveryLimit	<p>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を返す。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、再配信の制限に達したときに以下のことが発生する。</p> <ul style="list-style-type: none"> エラー送り先がコンフィグレーションされているか、またはエラー送り先の割り当てを超過する場合、永続および非永続メッセージは削除される。 エラー送り先がコンフィグレーションされており、エラー送り先の割り当てを超過していない場合、エラーメッセージのログが記録され、メッセージが削除される。しかし、メッセージが永続的である場合、永続ストアに保持される。これにより、永続メッセージは WebLogic Server の再起動時に再配信される。 <p>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。</p> <p>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。</p> <p>注意： RedeliveryLimit の変更は、すでにこの制限に達しているメッセージには適用されない。このようなメッセージが次に再配信される場合、それらは即座にエラー送り先にリダイレクトされる。メッセージが再配信された回数は永続しない。つまり、再起動後は各メッセージの配信試行回数がゼロにリセットされる。</p>	<p>Administration Console のフィールド表示: [再配信の制限]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: 2⁶³-1</p>

表 34-1 JMSTemplate の属性 (続き)

属性	説明	指定できる値とデフォルト値
TimeToDeliverOverride	<p>TimeToDeliver オーバーライド値 (ミリ秒単位またはスケジュール) を返す。</p> <p>プロデューサおよび接続ファクトリによって指定された配信時間とは関係なく、メッセージが生成されてからその対象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。</p> <p>デフォルト値 (-1) は、送り先が配信時間の設定をオーバーライドしないことを示す。TimeToDeliverOverride は、長さまたはスケジュールとして指定できる。</p> <p>注意：TimeToDeliverOverride の変更は将来のメッセージ配信のみに適用され、すでに生成されたメッセージには適用されない。</p>	<p>Administration Console のフィールド表示：[配信時間のオーバーライド]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p>
TimeToLiveOverride	<p>TimeToLiveOverride 値を返す。</p> <p>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係なく、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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示：[生存時間のオーバーライド]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

JMSTopic

説明

構文

```
<JMSTopic
  BytesMaximum="number"
  BytesPagingEnabled=( "default" | "false" | "true" )
  BytesThresholdHigh="number"
  BytesThresholdLow="number"
  DeliveryModeOverride=( "Persistent" | "Non-Persistent" | "No-Delivery" )
  DestinationKeys="list of JMSDestinationKey names"
  ErrorDestination="JMSDestination name"
  JNDIName="String"
  JNDINameReplicated=( "true" | "false" )
  MessagesMaximum="number"
  MessagesPagingEnabled=( "default" | "false" | "true" )
  MessagesThresholdHigh="number"
  MessagesThresholdLow="number"
  MulticastAddress="String"
  MulticastPort="number"
  MulticastTTL="number"
  Name="String"
  Notes="String"
  PriorityOverride="number"
  RedeliveryDelayOverride="number"
```

JMSTopic

```
RedeliveryLimit=number
StoreEnabled=( "default" | "false" | "true" )
Template=JMSTemplate name
TimeToDeliverOverride=String
TimeToLiveOverride=number
/>
```

親要素

- JMS Server

属性

表 35-1 JMS Topic の属性

属性	説明	指定できる値とデフォルト値
BytesMaximum	<p>BytesMaximum の値を返す。</p> <p>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。</p> <p>注意：分散送り先メンバーに JMS テンプレートが使用される場合、この設定は分散送り先セット全体ではなくその特定のメンバーのみに適用される。</p> <p>値の範囲：>= BytesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示：[最大バイト数]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesPagingEnabled	<p>「true」、「false」、「default」のいずれかを返す。</p> <p>送り先でバイト ページングを有効にするかどうかを指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグを選択しない場合 (false)、この送り先のバイト ページングは明示的に無効化される。 このフラグが選択され (true)、JMS サーバ用のページング ストアがコンフィグレーションされており、かつ BytesThresholdLow と BytesThresholdHigh の属性値がどちらも -1 より大きい場合、この送り先のバイト ページングは有効化される。 この属性が「default」に設定されている場合、この値は JMS テンプレートの値を継承する (JMS テンプレートが指定されている場合)。送り先の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、デフォルト値は false と同じになる。 	<p>Administration Console のフィールド表示 : [バイト ページングを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : default</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdHigh	<p>BytesThresholdHigh の値を返す。</p> <p>送り先に保存されているバイト数に基づいてイベントを発生させる上限しきい値。バイト数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている (およびページングストアがコンフィグレーションされている) 場合、送り先レベルのバイト ページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、送り先に対するバイト ページング、フロー制御、およびしきい値ログメッセージが無効になる。</p> <p>値の範囲 : <= BytesMaximum; >BytesThresholdLow</p> <p>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。</p> <p>注意 : バイト ページングが有効になっている場合、BytesThresholdHigh を -1 にリセットすることでそれを動的に無効にすることはできない (例外が送出される)。ページングを動的に無効にするには、BytesThresholdHigh を非常に大きな値に設定して、ページングがトリガされないようにする。</p>	<p>Administration Console のフィールド表示 : [最大バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
BytesThresholdLow	<p>BytesThresholdLow の値を返す。</p> <p>送り先に保存されているバイト数に基づいてイベントを発生させる下限しきい値。バイト数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>バイト ページング - バイト ページングが有効化されている場合、ページングが停止する (ページングが実行されている場合) 。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値の範囲 : < BytesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [最小バイトしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
DeliveryModeOverride	<p>DeliveryModeOverride 値を返す。</p> <p>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。</p> <p>No-Delivery は、配信モードがオーバーライドされないことを示す。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [配信モードオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : No-Delivery</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
DestinationKeys	テンプレートまたは送り先の送り先キーの読み取り専用配列を返す。	Administration Console のフィールド表示 : [送り先キー] 必須 : いいえ
ErrorDestination	再配信の上限に達したメッセージのエラー送り先の値を設定する。 再配信の上限に達したメッセージの送り先の名前を定義する。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、このメッセージはそのまま削除される。 注意 : エラー送り先は、ローカル JMS サーバ上でコンフィグレーションされた送り先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。	Administration Console のフィールド表示 : [エラー送り先] 必須 : いいえ セキュアな値 : null
JNDIName	送り先の JNDI 名を返す。 この値は、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。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通知されないため、送り先名をルックアップおよび使用できない。 この属性は動的にコンフィグレーションできない。	Administration Console のフィールド表示 : [JNDI 名] 必須 : いいえ
JNDINameReplicated	JNDINameReplicated 属性を true に設定すると、送り先の JNDI 名 (存在する場合) はクラスタ全体にレプリケートされる。JNDINameReplicated 属性を false に設定すると、送り先の JNDI 名 (存在する場合) は、その送り先が一部になっているサーバから参照できるだけになる。	必須 : いいえ デフォルト値 : true

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesMaximum	<p>MessagesMaximum の値を返す。</p> <p>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。</p> <p>注意：分散送り先メンバーに JMS テンプレートが使用される場合、この設定は分散送り先セット全体ではなくその特定のメンバーのみに適用される。</p> <p>値の範囲：>= MessagesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示：[最大メッセージ数]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesPagingEnabled	<p>「true」、「false」、「default」のいずれかを返す。</p> <p>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指定するためのフラグ。</p> <ul style="list-style-type: none"> このフラグを選択しない場合 (false)、この送り先のメッセージ ページングは明示的に無効化される。 このフラグが選択され (true)、JMS サーバ用のページング ストアがコンフィグレーションされており、かつ MessagesThresholdLow と MessagesThresholdHigh の属性値がどちらも -1 より大きい場合、送り先のメッセージ ページングは有効化される。 この属性が「default」に設定されている場合、この値は JMS テンプレートの値を継承する (JMS テンプレートが指定されている場合)。送り先の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、デフォルト値は false と同じになる。 	<p>Administration Console のフィールド表示: [メッセージ ページングを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: default</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdHigh	<p>MessagesThresholdHigh の値を返す。</p> <p>送り先に保存されているメッセージ数に基づいてイベントを発生させる上限しきい値。メッセージ数がこのしきい値を超えた場合、発生するイベントは以下のとおり。</p> <p>ログ メッセージ - 最大しきい値条件を示すメッセージがサーバに記録される。</p> <p>メッセージ ページング - メッセージ ページングが有効化されている (およびページング ストアがコンフィグレーションされている) 場合、送り先レベルのメッセージ ページングが開始される。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態に入り、プロデューサに対してメッセージフローを減らすよう指示する。</p> <p>値 -1 を指定すると、送り先に対するメッセージ ページング、フロー制御、およびしきい値ログ メッセージが無効になる。</p> <p>値の範囲 : <= MessagesMaximum; >MessagesThresholdLow</p> <p>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。</p> <p>注意 : メッセージ ページングが有効になっている場合、MessagesThresholdHigh を -1 にリセットすることでそれを動的に無効にすることはできない (例外が送出される)。ページングを動的に無効にするには、MessagesThresholdHigh を非常に大きな値に設定して、ページングがトリガされないようにする。</p>	<p>Administration Console のフィールド表示 : [最大メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
MessagesThresholdLow	<p>MessagesThresholdLow の値を返す。</p> <p>送り先に保存されているメッセージ数に基づいてイベントを発生させる下限しきい値。メッセージ数がこのしきい値を下回った場合、発生するイベントは以下のとおり。</p> <p>ログメッセージ - しきい値条件が解除されたことを示すメッセージがサーバに記録される。</p> <p>メッセージページング - メッセージページングが有効化されている場合、ページングが停止する (ページングが実行されている場合) 。</p> <p>フロー制御 - フロー制御が有効化されている場合、送り先が防御状態を解除して、プロデューサに対してメッセージフローを増やすよう指示する。</p> <p>値の範囲 : < MessagesThresholdHigh</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [最小メッセージしきい値]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>
MulticastAddress	<p>送り先によって使用されるマルチキャストアドレスを返す。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [マルチキャストアドレス]</p> <p>必須 : いいえ</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
MulticastPort	<p>送り先用のマルチキャスト ポートを取得する。</p> <p>これは、マルチキャスト用に使用される 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。</p>	<p>Administration Console のフィールド表示 : [マルチキャスト ポート]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 6001</p> <p>最小値 : 1</p> <p>最大値 : 65535</p>
MulticastTTL	<p>マルチキャスト メッセージの到達範囲を定義するネットワーク ホップ数を返す。</p> <p>これはマルチキャストの存続期間の値で、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 1 は、メッセージがルータを超えず、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 の値に依存しない。</p>	<p>Administration Console のフィールド表示 : [マルチキャスト生存時間]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 1</p> <p>最小値 : 0</p> <p>最大値 : 255</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
PriorityOverride	<p>PriorityOverride 値を返す。</p> <p>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係なく、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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [優先順位オーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 9</p>
RedeliveryDelayOverride	<p>ロールバックおよび回復されたメッセージが再配信されるまでの再配信遅延の値 (単位はミリ秒) を返す。</p> <p>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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [再配信遅延のオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
RedeliveryLimit	<p>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を返す。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、再配信の制限に達したときに以下のことが発生する。</p> <ul style="list-style-type: none"> エラー送り先がコンフィグレーションされているか、またはエラー送り先の割り当てを超過する場合、永続および非永続メッセージは削除される。 エラー送り先がコンフィグレーションされており、エラー送り先の割り当てを超過していない場合、エラーメッセージのログが記録され、メッセージが削除される。しかし、メッセージが永続的である場合、永続ストアに保持される。これにより、永続メッセージは WebLogic Server の再起動時に再配信される。 <p>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。</p> <p>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。</p> <p>注意： RedeliveryLimit の変更は、すでにこの制限に達しているメッセージには適用されない。このようなメッセージが次に再配信される場合、それらは即座にエラー送り先にリダイレクトされる。メッセージが再配信された回数は永続しない。つまり、再起動後は各メッセージの配信試行回数がゼロにリセットされる。</p>	<p>Administration Console のフィールド表示：[再配信の制限]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p> <p>最小値：-1</p> <p>最大値：2⁶³-1</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
StoreEnabled	<p>「true」、「false」、「default」のいずれかを返す。</p> <p>JMS サーバによって指定された永続ストアを送り先が使用するかどうかを指定するフラグ。</p> <ul style="list-style-type: none"> このフラグを true に設定したが、JMS サーバ用の永続ストアが定義されていない場合、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 false に設定した場合、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 default に設定した場合、JMS サーバ用の永続ストアが定義されていれば、送り先はその永続ストアを使用し、永続メッセージをサポートする。 <p>この属性は動的にコンフィグレーションできない。</p>	<p>Administration Console のフィールド表示 : [ストアを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : default</p>
Template	<p>送り先の派生元の JMS テンプレートを返す。</p> <p>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。送り先ごとの Template 属性の設定は静的。ただし、動的に変更できる。</p> <p>注意 : デフォルト値に設定された属性は、実行時の JMS テンプレートから送り先の値を継承する。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。</p>	<p>Administration Console のフィールド表示 : [テンプレート]</p> <p>必須 : いいえ</p>

表 35-1 JMSTopic の属性 (続き)

属性	説明	指定できる値とデフォルト値
TimeToDeliverOverride	<p>TimeToDeliver オーバーライド値 (ミリ秒単位またはスケジュール) を返す。</p> <p>プロデューサおよび接続ファクトリによって指定された配信時間とは関係なく、メッセージが生成されてからその対象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。</p> <p>デフォルト値 (-1) は、送り先が配信時間の設定をオーバーライドしないことを示す。TimeToDeliverOverride は、長さまたはスケジュールとして指定できる。</p> <p>注意 : TimeToDeliverOverride の変更は将来のメッセージ配信のみに適用され、すでに生成されたメッセージには適用されない。</p>	<p>Administration Console のフィールド表示 : [配信時間のオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p>
TimeToLiveOverride	<p>TimeToLiveOverride 値を返す。</p> <p>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係なく、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。</p> <p>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。</p> <p>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。</p>	<p>Administration Console のフィールド表示 : [生存時間のオーバーライド]</p> <p>必須 : いいえ</p> <p>デフォルト値 : -1</p> <p>最小値 : -1</p> <p>最大値 : 2⁶³-1</p>

JoltConnectionPool

説明

この Bean は、Jolt 接続プールを定義します。

構文

```
<JoltConnectionPool
  ApplicationPassword="String"
  DeploymentOrder="number"
  FailoverAddresses="list of Strings"
  MaximumPoolSize="number"
  MinimumPoolSize="number"
  Name="String"
  Notes="String"
  PrimaryAddresses="list of Strings"
  RecvTimeout="number"
  SecurityContextEnabled=( "true" | "false" )
  Targets="list of Target names"
  UserName="String"
  UserPassword="String"
  UserRole="String"
/>
```

親要素

- Domain

属性

表 36-1 JoltConnectionPool の属性

属性	説明	指定できる値とデフォルト値
ApplicationPassword	この接続プールアプリケーションパスワードを定義する。	Administration Console のフィールド表示: [アプリケーションパスワード] 必須: いいえ 暗号化: はい セキュアな値: null
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	必須: いいえ デフォルト値: 1000 最小値: 0 最大値: $2^{31}-1$

表 36-1 JoltConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
FailoverAddresses	プライマリ アドレスに定義されている接続が確立できないか失敗した場合に使用される、アドレスのリストを定義する。	Administration Console のフィールド表示: [フェイルオーバーアドレス] 必須: いいえ
MaximumPoolSize	Jolt 接続プールから開始できる接続の最大数を定義する。	Administration Console のフィールド表示: [最大プールサイズ] 必須: いいえ デフォルト値: 1
MinimumPoolSize	WebLogic Server が起動したときに、Jolt 接続プールに追加する接続の最小数を定義する。	Administration Console のフィールド表示: [最小プールサイズ] 必須: いいえ デフォルト値: 0
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ
PrimaryAddresses	Jolt 接続プールと Tuxedo との接続を確立するために使用するプライマリ アドレスのリストを定義する。	Administration Console のフィールド表示: [プライマリアドレス] 必須: いいえ
RecvTimeout	タイムアウトするまでクライアントが応答を待機する時間を定義する。	Administration Console のフィールド表示: [タイムアウト] 必須: いいえ デフォルト値: 0

表 36-1 JoltConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
SecurityContextEnabled	セキュリティ コンテキストがこのプールで有効かどうかを示す。	Administration Console の フィールド表示: [セキュリティ コンテキストを有効化] 必須: いいえ デフォルト値: false
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須: いいえ
UserName	この接続プールのユーザ名を定義する。	Administration Console の フィールド表示: [ユーザ名] 必須: いいえ
UserPassword	この接続プールのユーザ パスワードを定義する。	Administration Console の フィールド表示: [ユーザ パスワード] 必須: いいえ 暗号化: はい
UserRole	この接続プールのユーザ ロールを定義する。	Administration Console の フィールド表示: [ユーザ ロール] 必須: いいえ

JTA

説明

このインタフェースは、JTA コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、ドメイン レベルの JTA コンフィグレーションに適用されます。

構文

```
<JTA
  AbandonTimeoutSeconds="number"
  BeforeCompletionIterationLimit="number"
  CheckpointIntervalSeconds="number"
  ForgetHeuristics=( "true" | "false" )
  MaxResourceRequestsOnServer="number"
  MaxResourceUnavailableMillis="number"
  MaxTransactions="number"
  MaxUniqueNameStatistics="number"
  MaxXACallMillis="number"
  Name="String"
  Notes="String"
  TimeoutSeconds="number"
/>
```

親要素

- Domain
- Server

属性

表 37-1 JTA の属性

属性	説明	指定できる値とデフォルト値
AbandonTimeoutSeconds	トランザクション破棄タイムアウトの秒数を返す。	Administration Console のフィールド表示: [トランザクションを保持する最長時間] 必須: いいえ デフォルト値: 86400 最小値: 1 最大値: $2^{31}-1$
BeforeCompletionIterationLimit	トランザクション マネージャが <code>beforeCompletion</code> 同期化コールバックを実行するサイクルの最大数。同期化オブジェクトは <code>beforeCompletion</code> が既に呼び出されたオブジェクトであっても、 <code>beforeCompletion</code> で別のオブジェクトを登録できる。たとえば、EJB は、その <code>ejbStore()</code> メソッドで別の EJB を呼び出せる。これに対応するために、トランザクション マネージャはすべての同期化オブジェクトを呼び出し、新しいオブジェクトが登録されている場合、このサイクルを繰り返す。この値で、発生するサイクルの回数を制限する。	Administration Console のフィールド表示: [beforeCompletion の反復上限] 必須: いいえ デフォルト値: 10 最小値: 1 最大値: $2^{31}-1$

表 37-1 JTA の属性 (続き)

属性	説明	指定できる値とデフォルト値
CheckpointIntervalSeconds	トランザクション マネージャが、新しいトランザクション ログ ファイルを作成し、すべての古いトランザクション ログ ファイルをチェックしてそれらを削除できるかどうかを確認する間隔。デフォルトは 300 秒 (5 分)、最小値は 10 秒、最大値は 1800 秒 (30 分)。	Administration Console の フィールド表示: [チェックポイント間隔の秒数] 必須: いいえ デフォルト値: 300 最小値: 10 最大値: 1800
ForgetHeuristics	トランザクション マネージャがトランザクションのヒューリスティックな終了に対して自動的に XAResource 無視処理を実行するかどうかを示す。	Administration Console の フィールド表示: [ヒューリスティックを無視] 必須: いいえ デフォルト値: true
MaxResourceRequestsOnServer	サーバごとの、リソースに対して同時に発行できるリクエストの最大数。	必須: いいえ デフォルト値: 50 最小値: 10 最大値: $2^{31}-1$
MaxResourceUnavailableMillis	リソースが応答なしと宣言されている最長時間 (ミリ秒単位)。この時間を過ぎると、リソース プロバイダにより明示的に再登録されなくても、リソースは再び利用可能と宣言される。	必須: いいえ デフォルト値: 1800000 最小値: 0 最大値: $2^{63}-1$
MaxTransactions	サーバで許可される同時進行トランザクションの最大数。	Administration Console の フィールド表示: [最大トランザクション数] 必須: いいえ デフォルト値: 10000 最小値: 1 最大値: $2^{31}-1$

表 37-1 JTA の属性 (続き)

属性	説明	指定できる値とデフォルト値
MaxUniqueNameStatistics	保持する統計の対象となるユニークなトランザクション名の最大数。トランザクション名は、通常、ビジネストランザクションのカテゴリを表す(「送金」など)。	Administration Console のフィールド表示: [ユニーク名の最大数] 必須: いいえ デフォルト値: 1000 最小値: 0 最大値: $2^{31}-1$
MaxXACallMillis	リソースに対する XA 呼び出しに許される最長時間。リソースに対する特定の XA 呼び出しがこの上限を超えると、リソースは利用不可能と宣言される。	必須: いいえ デフォルト値: 120000 最小値: 0 最大値: 263-1
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
TimeoutSeconds	トランザクション タイムアウトの秒数を返す。	Administration Console のフィールド表示: [タイムアウト秒数] 必須: いいえ デフォルト値: 30 最小値: 1 最大値: $2^{31}-1$

JTAMigratableTarget

説明

JTA 回復マネージャを移行マネージャに登録するために内部的に使用される対象。

構文

```
<JTAMigratableTarget  
  HostingServer="Server name"  
  Name="String"  
  Notes="String"  
>
```

親要素

- Server

属性

表 38-1 JTAMigratableTarget の属性

属性	説明	指定できる値とデフォルト値
HostingServer	移行可能な対象を現在ホストしているサーバを返す。	必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ

JTARecoveryService

説明

プライマリ サーバのバックアップ サーバへの通知用に JTA 回復サービスによって使用されるプライベート MBean。

構文

```
<JTARecoveryService  
  Name="String"  
  Notes="String"  
>
```

親要素

- Server

属性

表 39-1 JTARecoveryService の属性

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ

Log

説明

ログメッセージの格納時に **WebLogic Server** で使用される格納場所、ファイルローテーションの基準、およびファイル数をコンフィグレーションします。このクラスの方法は、サーバ ログ ファイルおよびドメイン ログ ファイルの両方をコンフィグレーションします。

構文

```
<Log
  FileCount="number"
  FileMinSize="number of kilobytes"
  FileName="String"
  FileTimeSpan="number of hours"
  Name="String"
  Notes="String"
  NumberOfFilesLimited=( "true" | "false" )
  RotationTime="String"
  RotationType=( "bySize" | "byTime" | "none" )
/>
```

親要素

- Server

属性

表 40-1 Log の属性

属性	説明	指定できる値とデフォルト値
FileCount	ログをローテーションする場合にサーバが作成するログファイルの最大数。 <code>isNumberOfFilesLimited</code> が <code>true</code> であり、かつ、 <code>setRotationType</code> が <code>Size</code> または <code>Time</code> のいずれかである場合にのみ有効。	Administration Console の フィールド表示: [ファイル数] 必須: いいえ デフォルト値: 7
FileMinSize	サーバがログメッセージを独立したファイルに移動することになるサイズ (1 ~ 65535 KB)。ログファイルが指定された最小サイズに達した後、サーバが次にファイルサイズをチェックしたときに、現在のログファイル名が <code>FileName.n</code> に変更され、それ以降のメッセージを格納する新しいファイルが作成される。このフィールドは、[ローテーションタイプ] が [サイズ] に設定されている場合のみ意味を持つ。	Administration Console の フィールド表示: [最小ファイルサイズ] 必須: いいえ 単位: KB デフォルト値: 500 最小値: 1 最大値: 65535
FileName	現在のログメッセージを格納するファイルの名前。相対パス名は、サーバのルートディレクトリを基準とする。	Administration Console の フィールド表示: [ファイル名] 必須: いいえ デフォルト値: <code>weblogic.log</code>

表 40-1 Log の属性 (続き)

属性	説明	指定できる値とデフォルト値
FileTimeSpan	古いログメッセージが別のログファイルに移される間隔 (単位は時間) 。この値は、時間ベースのローテーション間隔を使用する場合にのみ意味を持つ。	Administration Console のフィールド表示 : [ファイルローテーション間隔] 必須 : いいえ 単位 : 時間 デフォルト値 : 24 最小値 : 1
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
NumberOfFilesLimited	古いメッセージを保存するためにサーバが作成するファイルの数を、FileCount で指定された数に制限する。この上限に達した後は、サーバによりもっとも古いログファイルが削除され、最新のサフィックスの付いた新しいログファイルが作成される。 このオプションを無効にした場合、新しいファイルが無限に作成される。このため、必要に応じてこれらのファイルを削除する必要がある。 この値は、ファイルのローテーションタイプに SIZE または TIME を指定した場合にのみ意味を持つ。	Administration Console のフィールド表示 : [ファイル数の制限] 必須 : いいえ デフォルト値 : false セキュアな値 : false

表 40-1 Log の属性 (続き)

属性	説明	指定できる値とデフォルト値
RotationTime	時間ベースのローテーションの開始時間を指定する。	Administration Console の フィールド表示 : [ローテーション時間] 必須 : いいえ デフォルト値 : 00:00
RotationType	古いログメッセージを別のログファイルに移すための条件。 <ul style="list-style-type: none"> • NONE。メッセージは 1 つのファイルに蓄積される。サイズが大きくなった場合、ファイルの内容を消去する必要がある。 • SIZE。ログファイルが FileMinSize に指定したサイズに達すると、ファイル名が FileName.n に変更される。 • TIME。TimeSpan に指定した間隔で、ファイル名が FileName.n に変更される。 ファイル名が変更されると、以後のメッセージは FileName に指定した名前の新しいファイルに蓄積される。	Administration Console の フィールド表示 : [ローテーションタイプ] 必須 : いいえ デフォルト値 : none

Machine

説明

この Bean はサーバが起動されるマシンを表します。サーバは `ServerMBean.setMachine()` を呼び出すことで、マシンにバインドされます。一般的に 1 つの `MachineMBean` が 1 つの物理的なマシンを表し、1 つの物理的なマシンが 1 つの `MachineMBean` を表しますが、複数の `MachineMBean` で表す 1 つのマルチホーム マシンを持つことも可能です。唯一の制限事項は、それぞれの `MachineMBean` がオーバーラップしないアドレスでコンフィグレーションされることです。1 つのコンフィグレーションには、論理名でロックアップされる 1 つまたは複数の `MachineMBean` がある可能性があります。

構文

```
<Machine
  Addresses="list of Strings"
  Name="String"
  Notes="String"
/>
```

親要素

- Domain

子要素

- NodeManager

属性

表 41-1 Machine の属性

属性	説明	指定できる値とデフォルト値
Addresses	このマシンが認知されるアドレスを返す。ホスト名またはリテラル IP アドレス。	Administration Console のフィールド表示: [アドレス] 必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ

MailSession

説明

MailSession リソース マネージャ接続ファクトリ用の MBean です。

構文

```
<MailSession
  DeploymentOrder=number"
  JNDIName=String"
  Name=String"
  Notes=String"
  Properties=java.util.Properties"
  Targets=list of Target names"
/>
```

親要素

- Domain

属性

表 42-1 MailSession の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須：いいえ</p> <p>デフォルト値：1000</p> <p>最小値：0</p> <p>最大値：$2^{31}-1$</p>
JNDIName		<p>Administration Console のフィールド表示：[JNDI 名]</p> <p>必須：いいえ</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示：[名前]</p> <p>必須：いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示：[メモ]</p> <p>必須：いいえ</p>

表 42-1 MailSession の属性 (続き)

属性	説明	指定できる値とデフォルト値
Properties		Administration Console の フィールド表示 : [プロパティ (key=value)] 必須 : いいえ
Targets	現在のドメイン内で、この項目のデブ ロイ先となる対象。	必須 : いいえ

MessagingBridge

説明

このクラスは、2つのメッセージング製品間の転送メカニズムをコンフィグレーションできるようにするメッセージングブリッジを表します。これにより、別々に実装した **WebLogic JMS** 間や、**WebLogic JMS** と他のメッセージング製品の間の相互運用性が提供されます。

WebLogic JMS とサードパーティの **JMS** 製品では、コンフィグレーションされているソース送り先および対象送り先との通信に **WebLogic Server** が提供するリソースアダプタを使用します。**JMS** 以外のメッセージング製品の場合、**JMS** 以外のソースまたは対象送り先にアクセスするためには、サードパーティ **OEM** ベンダからのカスタマイズされたアダプタが必要です。

構文

```
<MessagingBridge
  AsyncEnabled=( "true" | "false" )
  BatchInterval="number"
  BatchSize="number"
  DeploymentOrder="number"
  DurabilityEnabled=( "true" | "false" )
  IdleTimeMaximum="number"
```

MessagingBridge

```
Name="String"
Notes="String"
QOSDegradationAllowed=( "true" | "false" )
QualityOfService=( "Exactly-once" | "Atmost-once" | "Duplicate-okay" )
ReconnectDelayIncrease="number"
ReconnectDelayMaximum="number"
ReconnectDelayMinimum="number"
Selector="String"
SourceDestination="BridgeDestinationCommon name"
Started=( "true" | "false" )
TargetDestination="BridgeDestinationCommon name"
Targets="list of Target names"
TransactionTimeout="number"
/>
```

親要素

- Domain

属性

表 43-1 MessagingBridge の属性

属性	説明	指定できる値とデフォルト値
AsyncEnabled	<p>メッセージングブリッジが非同期メッセージングモードで動作するかどうかを示す。</p> <p>注意：この属性は、ソース送り先で非同期受信がサポートされているメッセージングブリッジのみに適用される。</p> <p>非同期モード (true) で動作するメッセージングブリッジは、ソース送り先によって駆動される。メッセージングブリッジはメッセージをリスンし、到着したものを転送する。値が false に設定されていると、ソースで非同期受信がサポートされていても、ブリッジは強制的に同期モードで動作する。</p> <p>注意：QOS が <i>Exactly-once</i> に設定されているメッセージングブリッジが非同期モードで動作するには、ソース送り先で MDBTransaction インタフェースがサポートされていなければならない。ソース送り先で MDBTransaction がサポートされていないことが検出されると、ブリッジは自動的に同期モードに切り替わる。</p>	<p>Administration Console のフィールド表示：[非同期モードを有効化]</p> <p>必須：いいえ</p> <p>デフォルト値：true</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
<p>BatchInterval</p>	<p>1 回のトランザクションで、バッチ サイズに到達したかどうかに関係なく、メッセージのバッチを送信するまでにブリッジが待機する最大時間 (単位：ミリ秒) を定義する。デフォルト値の -1 は、トランザクションが完了する前に、メッセージ数がバッチ サイズに達するまで、ブリッジが待機することを示す。</p> <p>注意：この属性は同期モードで動作し、QOS で 2 フェーズ トランザクションが必要になるブリッジのみに適用される。</p>	<p>Administration Console のフィールド表示：[バッチ間隔 (ミリ秒)]</p> <p>必須：いいえ</p> <p>デフォルト値：-1</p>
<p>BatchSize</p>	<p>1 回のトランザクションで処理されるメッセージ数を指定する。</p> <p>注意：この属性は同期モードで動作し、QOS で 2 フェーズ トランザクションが必要になるブリッジのみに適用される。</p>	<p>Administration Console のフィールド表示：[バッチ サイズ]</p> <p>必須：いいえ</p> <p>デフォルト値：10</p> <p>最小値：0</p>
<p>DeploymentOrder</p>	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須：いいえ</p> <p>デフォルト値：1000</p> <p>最小値：0</p> <p>最大値：2³¹-1</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
DurabilityEnabled	<p>メッセージングブリッジが永続メッセージを許可するかどうかを示す。これは恒久サブスクリプションを使用するソース送り先にのみ適用される。この機能により、ソース JMS 実装では、ブリッジが実行されていないときでも送られてきたメッセージを保存できる。ブリッジは、再起動時にこれらのメッセージをターゲット送り先に転送する。管理者側で永続性を無効にする選択を行える。</p>	<p>Administration Console のフィールド表示: [永続性を有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: true</p>
IdleTimeMaximum	<p>メッセージングブリッジの最大アイドル時間 (秒単位) を定義する。</p> <p>ブリッジが非同期モードで動作する場合、最大アイドル時間では、ブリッジがソースへの接続状態をチェックするまでの最長のアイドル時間を定義する。</p> <p>ブリッジが同期モードで動作する場合、最大アイドル時間では、関与しているトランザクションがない場合に、ブリッジが受信呼び出しに対してブロックできる時間の長さを定義する。</p>	<p>Administration Console のフィールド表示: [最大待機時間 (秒)]</p> <p>必須: いいえ</p> <p>デフォルト値: 60</p> <p>最小値: 0</p> <p>最大値: 2⁶³-1</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
QOSDegradationAllowed	<p>コンフィグレーションした QOS が使用できないときに、ブリッジで QOS の低下を許容するかどうかを示す。</p>	<p>Administration Console のフィールド表示 : [QOS デegradation]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>
QualityOfService	<p>以下のいずれかの値から、メッセージングブリッジの QOS (サービスの品質) を定義する。</p> <p>Exactly-once</p> <p>ソース送り先の各メッセージが、正確に 1 回だけターゲットに転送される。ブリッジで提供できる最高の QOS。</p> <p>Atmost-once</p> <p>ソースの 1 つのメッセージが、1 回だけターゲットに転送されるが、転送中に失われる可能性がある。</p> <p>Duplicate-okay</p> <p>ソースのメッセージが失われることはないが、一部のメッセージが複数回ターゲットに現れる可能性がある。</p>	<p>Administration Console のフィールド表示 : [サービスの品質]</p> <p>必須 : いいえ</p> <p>デフォルト値 : Exactly-once</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
ReconnectDelayIncrease	<p>メッセージングブリッジが、再接続の失敗から、さらにその次の試行を行うまでの待ち時間の延長分 (単位 : 秒) を定義する。</p> <p>この属性は <code>ReconnectDelayMinimum</code> 属性および <code>ReconnectDelayMaximum</code> 属性と共に機能する。送り先への接続が失敗すると、ブリッジはまず <code>ReconnectDelayMinimum</code> によって定義された秒数だけ待機する。再接続の試行が失敗するたびに、ブリッジは <code>ReconnectDelayIncrease</code> で定義された秒数だけ、待ち時間を延長する。最大遅延時間は、<code>ReconnectDelayMaximum</code> で定義される。待ち時間が最大値に到達すると、ブリッジはそれ以上待ち時間を延長しない。</p> <p>ブリッジから送り先への接続が成功すると、待ち時間は <code>ReconnectDelayMinimum</code> で定義された最小値にリセットされる。</p>	<p>Administration Console のフィールド表示 : [増加遅延 (秒)]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 5</p> <p>最小値 : 0</p> <p>最大値 : $2^{63}-1$</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
ReconnectDelayMaximum	<p>メッセージングブリッジが、ソースまたはターゲットへの再接続に失敗してから、さらにその次の試行を行うまでの最大の待ち時間 (単位 : 秒) を定義する。</p> <p>この属性は <code>ReconnectDelayMinimum</code> 属性および <code>ReconnectDelayIncrease</code> 属性と共に機能する。送り先への接続が失敗すると、ブリッジはまず <code>ReconnectDelayMinimum</code> によって定義された秒数だけ待機する。再接続の試行が失敗するたびに、ブリッジは <code>ReconnectDelayIncrease</code> で定義された秒数だけ、待ち時間を延長する。最大遅延時間は、<code>ReconnectDelayMaximum</code> で定義される。待ち時間が最大値に到達すると、ブリッジはそれ以上待ち時間を延長しない。</p> <p>ブリッジから送り先への接続が成功すると、待ち時間は <code>ReconnectDelayMinimum</code> で定義された初期値にリセットされる。</p>	<p>Administration Console のフィールド表示 : [最大遅延 (秒)]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 60</p> <p>最小値 : 0</p> <p>最大値 : $2^{63}-1$</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
ReconnectDelayMinimum	<p>メッセージングブリッジが、エラーの後でソース送り先またはターゲット送り先に再接続しようとするまでの最小の待ち時間 (単位 : 秒) を定義する。</p> <p>この属性は ReconnectDelayMaximum 属性および ReconnectDelayIncrease 属性と共に機能する。送り先への接続が失敗すると、ブリッジはまず ReconnectDelayMinimum によって定義された秒数だけ待機する。</p> <p>2 回目の試行も失敗した場合は、待ち時間を ReconnectDelayIncrease によって定義された秒数だけ延長する。最大遅延時間は、ReconnectDelayMaximum で定義される。待ち時間が最大値に到達すると、ブリッジはそれ以上待ち時間を延長しない。</p> <p>ブリッジから送り先への接続が成功すると、待ち時間は ReconnectDelayMinimum で定義された初期値にリセットされる。</p>	<p>Administration Console のフィールド表示 : [最小遅延 (秒)]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 15</p> <p>最小値 : 0</p> <p>最大値 : 2⁶³-1</p>
Selector	<p>メッセージングブリッジのメッセージセクタを定義する。</p> <p>メッセージセクタを使うと、メッセージングブリッジ経由で送信されるメッセージをフィルタ処理できる。選択条件に一致するメッセージのみが、メッセージングブリッジを経由して送信される。キューの場合、選択条件に一致しないメッセージは後に残され、キュー内で蓄積される。トピックの場合、接続条件に一致しないメッセージは破棄される。</p>	<p>Administration Console のフィールド表示 : [セクタ]</p> <p>必須 : いいえ</p>

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
SourceDestination	<p>メッセージングブリッジのソース送り先を定義する。</p> <p>この属性は、ブリッジによるメッセージの読み込み元になるソース送り先を定義する、<code>BridgeDestinationMBean</code> または <code>JMSBridgeDestinationMBean</code> のいずれかのインスタンスでなければならない。</p>	Administration Console のフィールド表示: [ソース送り先]
Started	<p>メッセージングブリッジの初期状態 (ブリッジを起動した時の状態) を定義する。値が <code>true</code> であれば、ブリッジは動作中の状態である。値が <code>false</code> であれば、ブリッジは一時的に停止されている。</p> <p>注意: これは、ブリッジの実行時の状態ではない。</p>	Administration Console のフィールド表示: [起動する] 必須: いいえ デフォルト値: <code>true</code>
TargetDestination	<p>メッセージングブリッジのターゲット送り先を定義する。</p> <p>この属性は、ソース送り先から受信したメッセージの、ブリッジによる送信先になるターゲット送り先を定義する、<code>BridgeDestinationMBean</code> または <code>JMSBridgeDestinationMBean</code> のいずれかのインスタンスでなければならない。</p>	Administration Console のフィールド表示: [対象送り先]
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須: いいえ

表 43-1 MessagingBridge の属性 (続き)

属性	説明	指定できる値とデフォルト値
TransactionTimeout	<p>各トランザクションがタイムアウトするまでに、トランザクションマネージャが待機する時間 (単位 : 秒) を定義する。</p> <p>トランザクションタイムアウトは、ブリッジの QOS でトランザクションが必要になる場合に使用される。ブリッジの QOS が <i>Exactly-once</i> にコンフィグレーションされている場合、送信と受信は 1 回のトランザクションで完了する。</p>	<p>Administration Console のフィールド表示 : [トランザクションタイムアウト]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 30</p> <p>最小値 : 0</p> <p>最大値 : $2^{31}-1$</p>

MigratableRMIService

説明

移行可能サービスのテスト サービス。

構文

```
<MigratableRMIService
  DeploymentOrder=number
  Name=String
  Notes=String
  Targets=list of Target names
/>
```

親要素

- Domain

属性

表 44-1 MigratableRMIService の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須：いいえ</p> <p>デフォルト値：1000</p> <p>最小値：0</p> <p>最大値：$2^{31}-1$</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>必須：いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>必須：いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須：いいえ</p>

MigratableTarget

説明

一度にクラスタの1つのサーバでアクティブになるサービスに適した対象。

構文

```
<MigratableTarget  
  HostingServer="Server name"  
  Name="String"  
  Notes="String"  
>
```

親要素

- Domain

属性

表 45-1 MigratableTarget の属性

属性	説明	指定できる値とデフォルト値
HostingServer	移行可能な対象を現在ホストしているサーバを返す。	必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ

NetworkChannel

説明

この MBean では、ネットワーク チャネルを定義します。ネットワーク チャネルは、サーバのデフォルトのリスン ポート以外に追加ポートをコンフィグレーションする場合に使用します。ネットワーク チャネルでは、IIOP はサポートされません。

ネットワーク チャネルは、複数のクラスタおよびサーバで指定できます。クラスタで指定したチャネルは、そのクラスタのメンバーであるすべてのサーバに適用されます。サーバは、複数のチャネルをサポートできます。**NetworkAccessPointMBean** を使用すると、サーバでネットワーク チャネルをきめ細かくチューニングできます。また、**NetworkAccessPointMBean** では、サーバで特定のチャネルに使用されるリスンアドレスおよび外部 DNS 名も設定できます。

サーバには、**ServerMBean ListenPort**、**ServerMBean AdministrationPort**、および **SSLMBean ListenPort** という 3 つのデフォルト リスン ポートがあります。デフォルト リスン ポートは、重みが 50 の暗黙的チャネルで構成されます。

また、ネットワーク チャネルでは、サーバ間接続の作成も定義します。あるサーバが別のサーバへの新しい接続を開始する際は、必要なプロトコルをサポートする、重みが最も高い共通の (同じ名前の) チャネルが接続先ポートの決定に使用されます。

構文

```
<NetworkChannel
  AcceptBacklog=number
  COMEnabled=( "true" | "false" )
  ChannelWeight=number
  ClusterAddress=String
  CompleteCOMMessageTimeout=number of seconds
  CompleteHTTPMessageTimeout=number of seconds
  CompleteT3MessageTimeout=number of seconds
  DeploymentOrder=number
  Description=String
  HTTPEnabled=( "true" | "false" )
  HTTPSEnabled=( "true" | "false" )
  ListenPort=number
  ListenPortEnabled=( "true" | "false" )
  LoginTimeoutMillis=number of milliseconds
  LoginTimeoutMillisSSL=number of milliseconds
  MaxCOMMessageSize=number of bytes
  MaxHTTPMessageSize=number of bytes
  MaxT3MessageSize=number of bytes
  Name=String
  Notes=String
  OutgoingEnabled=( "true" | "false" )
  SSLListenPort=number
  SSLListenPortEnabled=( "true" | "false" )
  T3Enabled=( "true" | "false" )
  T3SEnabled=( "true" | "false" )
  Targets=list of Target names
  TunnelingClientPingSecs=number of seconds
  TunnelingClientTimeoutSecs=number of seconds
  TunnelingEnabled=( "true" | "false" )
/>
```

親要素

- Domain

属性

表 46-1 NetworkChannel の属性

属性	説明	指定できる値とデフォルト値
AcceptBacklog	リスンポート上の接続リクエストの受け入れられるバックログ。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。バックログを 0 に設定すると、一部の OS 上で着信接続の受け付けを防ぐことができる。	Administration Console のフィールド表示: [バックログを受け入れ] 必須: いいえ デフォルト値: 50 最小値: 0
COMEnabled	プレーンテキスト (非 SSL) の COM トラフィックを有効化するかどうかを示す。	Administration Console のフィールド表示: [COM を有効化] 必須: いいえ デフォルト値: false
ChannelWeight	サーバ間接続を作成するときこのチャンネルに割り当てる重み。	Administration Console のフィールド表示: [チャンネルの重み] 必須: いいえ デフォルト値: 50 最小値: 1 最大値: 100
ClusterAddress	このチャンネルのクラスタアドレス。これを設定しない場合は、代わりにクラスタのコンフィグレーションで指定したクラスタアドレスが使用される。	Administration Console のフィールド表示: [クラスタアドレス] 必須: いいえ

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
CompleteCOMMessageTimeout	完全な COM メッセージの受信を待機する期間の最大秒数。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console の フィールド表示 : [COM メッセージのタイムアウトまでの時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 セキュアな値 : 60 最小値 : 0 最大値 : 480
CompleteHTTPMessageTimeout	完全な HTTP メッセージの受信を待機する期間の最大秒数。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console の フィールド表示 : [HTTP メッセージのタイムアウトまでの時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 セキュアな値 : 60 最小値 : 0 最大値 : 480
CompleteT3MessageTimeout	完全な T3 メッセージの受信を待機する期間の最大秒数。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console の フィールド表示 : [T3 メッセージのタイムアウトまでの時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 セキュアな値 : 60 最小値 : 0 最大値 : 480

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
Description	<p>コンソールで表示するための、このチャンネルの短い任意の説明。長文の説明には、Notes フィールドを使用する。</p>	<p>Administration Console のフィールド表示: [記述]</p> <p>必須: いいえ</p>
HTTPEnabled	<p>プレーンテキスト (非 SSL) の HTTP トラフィックを有効化するかどうか。</p>	<p>Administration Console のフィールド表示: [HTTP を有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p> <p>セキュアな値: false</p>
HTTPSEnabled	<p>セキュア (SSL) HTTP トラフィックを有効化するかどうか。</p>	<p>Administration Console のフィールド表示: [HTTPS を有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p> <p>セキュアな値: true</p>

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
ListenPort	チャンネルのプレーンテキスト (非 SSL) のリスンポート。個々のサーバでこの値をオーバーライドできるが、ここで無効化した場合にはポートを有効化できない。また、ここで有効化した場合は、ポートを無効化できない。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [リスンポート] 必須 : いいえ デフォルト値 : 8001 最小値 : 1 最大値 : 65534
ListenPortEnabled	プレーンテキストのポートをチャンネル用に有効化するかどうか。	Administration Console のフィールド表示 : [リスンポートを有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : false
LoginTimeoutMillis	サーバのログインタイムアウト (単位 : ミリ秒) 。この値は 0 以上とする必要がある。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [ログインタイムアウト] 必須 : いいえ 単位 : ミリ秒 デフォルト値 : 5000 セキュアな値 : 5000 最小値 : 0 最大値 : 100000

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
LoginTimeoutMillisSSL	SSL ログイン シーケンスで許容される持続時間。この期間を超えると、ログインはタイムアウトとなる。0 を指定すると無効になる。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [SSL ログインタイムアウト] 必須 : いいえ 単位 : ミリ秒 デフォルト値 : 25000 セキュアな値 : 25000 最小値 : 0 最大値 : $2^{31}-1$
MaxCOMMessageSize	メッセージヘッダで許容される最大の COM メッセージ サイズ。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [最大 COM メッセージ サイズ] 必須 : いいえ 単位 : バイト デフォルト値 : 10000000 セキュアな値 : 10000000 最小値 : 4096 最大値 : 2000000000
MaxHTTPMessageSize	メッセージヘッダで許容される最大の HTTP メッセージ サイズ。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。個々のサーバでこの値をオーバーライドするには、 NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [最大 HTTP メッセージ サイズ] 必須 : いいえ 単位 : バイト デフォルト値 : 10000000 セキュアな値 : 10000000 最小値 : 4096 最大値 : 2000000000

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
MaxT3MessageSize	メッセージヘッダで許容される最大の T3 メッセージサイズ。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。個々のサーバでこの値をオーバーライドするには、 <code>NetworkAccessPointMBean</code> を使用する。	Administration Console のフィールド表示:[最大 T3 メッセージサイズ] 必須: いいえ 単位: バイト デフォルト値: 10000000 セキュアな値: 10000000 最小値: 4096 最大値: 2000000000
Name	チャンネルの名前。この名前は「.WL」で開始してはならない。	Administration Console のフィールド表示:[名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示:[メモ] 必須: いいえ
OutgoingEnabled	新規のサーバ間接続で、初期化時にこのチャンネルを考慮するかどうか。	Administration Console のフィールド表示:[送信を有効化] 必須: いいえ デフォルト値: true セキュアな値: true

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
SSLListenPort	チャンネルの SSL リスン ポート。個々のサーバでこの値をオーバーライドできるが、ここで無効化した場合にはポートを有効化できない。また、ここで有効化した場合は、ポートを無効化できない。SSL をコンフィグレーションしなければ、このポートは機能しない。個々のサーバでこの値をオーバーライドするには、NetworkAccessPointMBean を使用する。	Administration Console の フィールド表示 : [SSL リスン ポート] 必須 : いいえ デフォルト値 : 8002 最小値 : 1 最大値 : 65534
SSLListenPortEnabled	SSL ポートをチャンネル用に有効化するかどうか。この設定に加えて SSL のコンフィグレーションと有効化を行わなければ、SSL ポートは機能しない。	Administration Console の フィールド表示 : [SSL リスン ポートを有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : true
T3Enabled	プレーンテキスト (非 SSL) の T3 トラフィックを有効化するかどうか。デフォルトのチャンネルでは T3 トラフィックを無効化できない。	Administration Console の フィールド表示 : [T3 を有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : false
T3SEnabled	セキュア T3 トラフィックを有効化するかどうか。デフォルトのチャンネルでは T3 トラフィックを無効化できない。	Administration Console の フィールド表示 : [T3S を有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : false
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須 : いいえ

表 46-1 NetworkChannel の属性 (続き)

属性	説明	指定できる値とデフォルト値
TunnelingClientPingSeconds	HTTP トンネリング クライアントがまだ有効かどうかを確認するために ping を実行する間隔 (単位 : 秒) 。個々のサーバでこの値をオーバーライドするには、NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [トンネリング クライアント Ping] 必須 : いいえ 単位 : 秒 デフォルト値 : 45
TunnelingClientTimeoutSecs	見つからない HTTP トンネリング クライアントを応答なしとみなすまでの期間 (単位 : 秒) 。個々のサーバでこの値をオーバーライドするには、NetworkAccessPointMBean を使用する。	Administration Console のフィールド表示 : [トンネリング クライアント タイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 40 セキュアな値 : 40
TunnelingEnabled	HTTP によるトンネリングを有効化する。	Administration Console のフィールド表示 : [トンネリングを有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : false

NodeManager

説明

この Bean は、マシンに関連付けられたノード マネージャを表します。

構文

```
<NodeManager
  Certificate="String"
  CertificatePassword="String"
  CertificateType="String"
  ListenAddress="String"
  ListenPort="number"
  Name="String"
  Notes="String"
  TrustedCertsFile="String"
/>
```

親要素

- Machine
- UnixMachine

属性

表 47-1 NodeManager の属性

属性	説明	指定できる値とデフォルト値
Certificate	ノードマネージャでセキュアな通信を実現するために使用する証明書ファイル。パスは、管理サーバのルートディレクトリを基準とした相対パス。	Administration Console のフィールド表示 : [証明書] 必須 : いいえ デフォルト値 : demo.crt
CertificatePassword	ノードマネージャでセキュアな通信を実現するために使用する証明書パスワード。	Administration Console のフィールド表示 : [証明書パスワード] 必須 : いいえ 暗号化 : はい デフォルト値 : password
CertificateType	ノードマネージャでセキュアな通信を実現するために使用する証明書のタイプ。	必須 : いいえ デフォルト値 : RSA

表 47-1 NodeManager の属性 (続き)

属性	説明	指定できる値とデフォルト値
ListenAddress	ノードマネージャが接続をリスンするために使用するアドレス。	Administration Console のフィールド表示: [リスンアドレス] 必須: いいえ デフォルト値: localhost セキュアな値: "127.0.0.1"
ListenPort	ノードマネージャのリスンポートを返す。	Administration Console のフィールド表示: [リスンポート] 必須: いいえ デフォルト値: 5555 最小値: 0 最大値: 65534
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
TrustedCertsFile	ノードマネージャでセキュアな通信を実現するために使用する信頼された証明書ファイル。パスは、管理サーバのルートディレクトリを基準とした相対パス。	Administration Console のフィールド表示: [信頼性のある証明書ファイル] 必須: いいえ デフォルト値: trusted.crt

Realm

説明

構文

```
<Realm
  CachingRealm="CachingRealm name"
  EnumerationAllowed=( "true" | "false" )
  Name="String"
  Notes="String"
  ResultsBatchSize="number"
/>
```

親要素

- Domain

属性

表 48-1 Realm の属性

属性	説明	指定できる値とデフォルト値
CachingRealm	<p>ファイル レalm 以外のレalmを使用する場合、そのレalmは CachingRealm MBean にレalm名を付加して指定する。その CachingRealm MBean は、Realm MBean に付加される。</p> <p>属性に値が指定されている場合は、代替セキュリティ レalmが使用される。属性が null の場合は、ファイル レalmのみ使用できる。</p>	<p>Administration Console のフィールド表示: [キャッシングレalm]</p> <p>必須: いいえ</p>
EnumerationAllowed	<p>ユーザ、グループ、およびメンバーシップを列挙して、サービス拒否攻撃を防止できるかどうかを指定する (ユーザまたはグループが多数の場合)。</p>	<p>必須: いいえ</p> <p>デフォルト値: true</p> <p>セキュアな値: false</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>
ResultsBatchSize	<p>ユーザ グループ、および ACL を返すためのバッチ サイズ (rpc ごとに返すユーザ、グループ、および ACL の数) を指定する。その目的は、ユーザ、グループ、または ACL ごとに rpc が行われたり、1 回の非常に大規模な rpc でメモリ不足になったりするのを防止することにある。</p>	<p>Administration Console のフィールド表示: [結果バッチサイズ]</p> <p>必須: いいえ</p> <p>デフォルト値: 200</p> <p>最小値: 0</p>

RMCFactory

説明

RMCFactoryMBean は、J2EE アプリケーションの J2EE リソース マネージャの接続ファクトリを表します。リソース マネージャの接続ファクトリの MBean は、このインタフェースを拡張します (たとえば、MailSessionMBean)。

構文

```
<RMCFactory
  DeploymentOrder="number"
  JNDIName="String"
  Name="String"
  Notes="String"
  Targets="list of Target names"
/>
```

親要素

- Domain

属性

表 49-1 RMCFactory の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
JNDIName		必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須: いいえ

Security

説明

WebLogic ドメインのセキュリティ プロパティを指定します。

構文

```
<Security
  AuditProviderClassName="String"
  CompatibilityMode=( "true" | "false" )
  ConnectionFilter="String"
  ConnectionFilterRules="list of Strings"
  ConnectionLoggerEnabled=( "true" | "false" )
  GuestDisabled=( "true" | "false" )
  Name="String"
  Notes="String"
  PasswordPolicy="PasswordPolicy name"
  Realm="Realm name"
  SystemUser="String"
/>
```

親要素

- Domain

属性

表 50-1 Security の属性

属性	説明	指定できる値とデフォルト値
AuditProviderClassName		Administration Console のフィールド表示: [監査プロバイダクラス] 必須: いいえ
CompatibilityMode	WebLogic Server 6.x のセキュリティコンフィグレーションからの移行を有効にするかどうかを指定する。	必須: いいえ デフォルト値: false
ConnectionFactory	接続フィルタを実装する Java クラスの名前。接続フィルタは <code>weblogic.security.net.ConnectionFilter</code> インタフェースの実装でなければならない。WebLogic Server ではデフォルトの実装が用意されている。	Administration Console のフィールド表示: [接続フィルタ] 必須: いいえ
ConnectionFactoryRules		Administration Console のフィールド表示: [接続フィルタ ルール] 必須: いいえ

表 50-1 Security の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnectionLoggerEnabled	受け付けた接続のロギングを有効にする。この属性を使用すると、システム管理者は受信時接続をログファイルで動的にチェックして、フィルタ処理を実行する必要があるかどうかを決定できる。	Administration Console のフィールド表示: [接続ログを有効化] 必須: いいえ デフォルト値: false
GuestDisabled	非推奨 。guest ログインを使用して WebLogic Server リソースにアクセスできるかどうかを指定する。この属性は互換性モードで使用する。	Administration Console のフィールド表示: [ゲスト不可] 必須: いいえ デフォルト値: true
Name	MBean の名前を設定する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ
PasswordPolicy	非推奨 。パスワードポリシーを設定する。このインタフェースは互換性モードで使用する。	必須: いいえ
Realm	レルム ポリシーを設定する。	Administration Console のフィールド表示: [レルム] 必須: いいえ
SystemUser	system ユーザの名前。この属性は、必ず指定しなければならない。	Administration Console のフィールド表示: [システム ユーザ] デフォルト値: system

SecurityConfiguration

説明

ドメイン全体のセキュリティ コンフィグレーション情報を指定します。

構文

```
<SecurityConfiguration  
  Name="String"  
  Notes="String"  
  WebAppFilesCaseInsensitive="String"  
>
```

親要素

- Domain

属性

表 51-1 SecurityConfiguration の属性

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
WebAppFilesCaseInsensitive		必須 : いいえ デフォルト値 : os

Server

説明

このクラスは、WebLogic Server を表します。WebLogic Server は、J2EE アプリケーションのコンテナである Java プロセスです。

構文

```
<Server
  AcceptBacklog=number
  AdministrationPort=number
  AutoKillIfFailed=( "true" | "false" )
  AutoRestart=( "true" | "false" )
  COMEnabled=( "true" | "false" )
  ClasspathServletDisabled=( "true" | "false" )
  ClientCertProxyEnabled=( "true" | "false" )
  Cluster=Cluster name
  ClusterWeight=number
  CompleteCOMMessageTimeout=number of seconds
  CompleteHTTPMessageTimeout=number of seconds
  CompleteIIOPMessageTimeout=number of seconds
  CompleteT3MessageTimeout=number of seconds
  DefaultIIOPPassword=String
  DefaultIIOPUser=String
```

Server

```
DefaultInternalServletsDisabled=( "true" | "false" )
DefaultProtocol=( "t3" | "t3s" | "http" | "https" | "iiop" )
DefaultSecureProtocol=( "t3s" | "https" )
DefaultTGIOPPassword="String"
DefaultTGIOPUser="String"
DomainLogFilter="DomainLogFilter name"
EnabledForDomainLog=( "true" | "false" )
ExpectedToRun=( "true" | "false" )
ExternalDNSName="String"
ExtraEjbcOptions="String"
ExtraRmicOptions="String"
HealthCheckIntervalSeconds="number of seconds"
HealthCheckTimeoutSeconds="number of seconds"
HttpTraceSupportEnabled=( "true" | "false" )
HttpdEnabled=( "true" | "false" )
IIOPEnabled=( "true" | "false" )
IdleIIOPConnectionTimeout="number of seconds"
InstrumentStackTraceEnabled=( "true" | "false" )
InterfaceAddress="String"
JDBCLogFileNames="String"
JDBCLoggingEnabled=( "true" | "false" )
JMSDefaultConnectionFactoryEnabled=( "true" | "false" )
JNDITransportableObjectFactoryList="list of Strings"
JavaCompiler="String"
JavaCompilerPostClassPath="String"
JavaCompilerPreClassPath="String"
ListenAddress="String"
ListenDelaySecs="number"
ListenPort="number"
ListenPortEnabled=( "true" | "false" )
LogRemoteExceptionsEnabled=( "true" | "false" )
LoginTimeout="number of milliseconds"
LoginTimeoutMillis="number of milliseconds"
LowMemoryGCThreshold="number"
LowMemoryGranularityLevel="number"
LowMemorySampleSize="number"
LowMemoryTimeInterval="number of seconds"
MSIFileReplicationEnabled=( "true" | "false" )
Machine="Machine name"
ManagedServerIndependenceEnabled=( "true" | "false" )
MaxCOMMessageSize="number of bytes"
MaxHTTPMessageSize="number of bytes"
MaxIIOPMessageSize="number of bytes"
MaxOpenSockCount="number"
MaxT3MessageSize="number of bytes"
Name="String"
NativeIOEnabled=( "true" | "false" )
NetworkAccessPoints="number"
Notes="String"
```

```
PreferredSecondaryGroup="String"
ReplicationGroup="String"
RestartDelaySeconds="number of seconds"
RestartIntervalSeconds="number of seconds"
RestartMax="number"
ReverseDNSAllowed=( "true" | "false" )

ServerLifecycleTimeoutVal="number of seconds"
ServerVersion="String"
StagingMode=( "stage" | "nostage" | "external_stage" )
StartupMode="String"
StdoutDebugEnabled=( "true" | "false" )
StdoutEnabled=( "true" | "false" )
StdoutFormat=( "standard" | "noid" )
StdoutLogStack=( "true" | "false" )
StdoutSeverityLevel=( "64" | "32" | "16" | "8" | "4" | "2" | "1" )
StuckThreadMaxTime="number of seconds"
StuckThreadTimerInterval="number of seconds"
TGIOPEnabled=( "true" | "false" )
ThreadPoolPercentSocketReaders="number"
TransactionLogFilePrefix="String"
TransactionLogFileWritePolicy=( "Cache-Flush" | "Direct-Write" )
TunnelingClientPingSecs="number of seconds"
TunnelingClientTimeoutSecs="number of seconds"
TunnelingEnabled=( "true" | "false" )
UploadDirectoryName="String"
VerboseEJBDeploymentEnabled="String"
WeblogicPluginEnabled=( "true" | "false" )
XMLEntityCache="XMLEntityCache name"
XMLRegistry="XMLRegistry name"
/>
```

親要素

- Domain

子要素

- COM

Server

- IIOP
- JTA
- JTAMigratableTarget
- JTARecoveryService
- Log
- SSL
- ServerStart
- ExecuteQueue
- WebServer

属性

表 52-1 Server の属性

属性	説明	指定できる値とデフォルト値
AcceptBacklog	プレーンテキストと SSL ポートの両方に対する新しい TCP 接続要求のバックログを許可する。バックログを 0 に設定すると、一部の OS 上で着信接続の受け付けを防ぐことができる。	Administration Console のフィールド表示: [バックログを受け入れ] 必須: いいえ デフォルト値: 50 最小値: 0
AdministrationPort	サーバのセキュア管理ポート。この設定は、このサーバの DomainMBean の同じフィールドをオーバーライドするために使用する。この値が 0 以外の場合、DomainMBean の同じフィールドがこのサーバに対して使用される。このポートでは、SSL をコンフィグレーションおよび有効化する必要がある。アクティブな管理ポートは、管理者資格を使用した接続のみを許可する。また、サーバ上の他のポートが管理者資格を使用した接続を受け付けないようにする。 T3S および HTTPS プロトコルをサポートする。管理対象サーバでは、「-Dweblogic.management.server=https://admin_server:administration_port」を使用して管理サーバに接続する必要がある。	Administration Console のフィールド表示: [ローカル管理ポートのオーバーライド (0: オーバーライドなし)] 必須: いいえ デフォルト値: 0 セキュアな値: (値 > 0) 最小値: 0 最大値: 65534

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
AutoKillIfFailed	障害が発生したサーバをノードマネージャが自動的に強制停止する処理を有効化/無効化する。	Administration Console のフィールド表示 : [失敗時の自動強制停止] 必須 : いいえ デフォルト値 : false
AutoRestart	クラッシュしたサーバをノードマネージャが自動的に再起動する処理を有効化/無効化する。	Administration Console のフィールド表示 : [自動再起動] 必須 : いいえ デフォルト値 : true
COMEnabled	プレーンテキストポート上で COM サポートを有効にするかどうか (COM は SSL ポートではサポートされない)。	Administration Console のフィールド表示 : [COM を有効化] 必須 : いいえ デフォルト値 : false
ClasspathServletDisabled	ClasspathServlet は、クラスパス内のクラス ファイルを処理し、管理も含めすべての Web アプリケーションでデフォルトとして登録される。多くのアプリケーションでは、ClasspathServlet を有効にする必要はないが、チェックしない場合、セキュリティホールとなる。	必須 : いいえ デフォルト値 : false

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
ClientCertProxyEnabled	<p>値を true に設定すると、クライアントからの ID 証明書が、プロキシサーバプラグインによって、このサーバインスタンスにデプロイされているすべての Web アプリケーションに渡される。</p> <p>プロキシサーバプラグインは、WL-Proxy-Client-Cert ヘッダの各 ID 証明書をエンコードし、そのヘッダを WebLogic Server インスタンスに渡す。</p> <p>WebLogic Server インスタンスは、それがセキュアなソースから来ているものと信頼してヘッダから証明書情報を取得し、その情報を使用してユーザを認証する。</p> <p>true を指定した場合は、weblogic.security.net.ConnectionFilter を使用して、この WebLogic Server インスタンスが、プロキシサーバプラグインが動作しているマシンからの接続のみを受け入れるようにする。接続フィルタを使用しないで true を指定すると、WL-Proxy-Client-Cert ヘッダのなりすましが可能なためにセキュリティが脆弱になる。</p> <p>デフォルトでは (または false を指定した場合は)、このサーバにデプロイされる各 Web アプリケーションの weblogic.xml デプロイメント記述子によって、プロキシサーバプラグインから送られてきた証明書を Web アプリケーションが信頼するかどうかが決められる。デフォルトでは (またはデプロイメント記述子で false が指定された場合)、ユーザはプロキシサーバプラグインから Web アプリケーションにログインできない。</p> <p>このメソッドで設定される値は、サーバがクラスタの一部であり、そのクラスタの ClusterMBean#setClientCertProxyEnabled(boolean) メソッドで true が指定される場合はオーバーライドされる。</p>	<p>Administration Console のフィールド表示 : [クライアント証明書プロキシを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
Cluster	このサーバが所属するクラスタ。設定した場合、サーバはクラスタのマルチキャストイベントをリスンする。	Administration Console の フィールド表示 : [クラスタ] 必須 : いいえ
ClusterWeight	サーバにかかる負荷をクラスタ内の他のサーバとの比率で指定する値を定義する。 すべてのサーバにデフォルトの重み (100) または同じ重みが指定されている場合、各サーバに同じ割合で負荷がかかる。あるサーバの重みが 50 で他のサーバの重みがすべて 100 になっている場合は、重みが 50 のサーバには他のサーバの半分の負荷がかかる。	Administration Console の フィールド表示 : [クラスタの重み] 必須 : いいえ デフォルト値 : 100 最小値 : 1 最大値 : 100
CompleteCOMMessageTimeout	完全な COM メッセージを取得するまでの最大秒数を指定する。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。この設定は、デフォルトポート (<code>ServerMBean setListenPort</code> および <code>setAdministrationPort</code> または <code>SSLMBean setListenPort</code>) のいずれかを使用して開始された接続にのみ適用される。追加ポート上での接続は、 <code>NetworkChannelMBean</code> によってチューニングする。	Administration Console の フィールド表示 : [COM メッセージタイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 0 最大値 : 480

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
CompleteHTTPMessageTimeout	完全な HTTP メッセージを取得するまでの最大秒数を指定する。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。この設定は、デフォルトポート (ServerMBean setListenPort および setAdministrationPort または SSLMBean setListenPort) のいずれかを使用して開始された接続にのみ適用される。追加ポート上での接続は、NetworkChannelMBean によってチューニングする。	Administration Console のフィールド表示 : [HTTP メッセージタイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 0 最大値 : 480
CompleteIIOPMessageTimeout	完全な IIOP メッセージを取得するまでの最大秒数を指定する。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。	Administration Console のフィールド表示 : [IIOP メッセージタイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 0 最大値 : 480

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
CompleteT3MessageTimeout	完全な T3 メッセージを取得するまでの最大秒数を指定する。この属性は、サービス拒否攻撃に対する防御として有用である。サービス拒否攻撃では、いつまでも送信が終了しない一定のサイズのメッセージを、呼び出し側が送信することが示される。この設定は、デフォルトポート (ServerMBean setListenPort および setAdministrationPort または SSLMBean setListenPort) のいずれかを使用して開始された接続にのみ適用される。追加ポート上での接続は、NetworkChannelMBean によってチューニングする。	Administration Console のフィールド表示 : [T3 メッセージタイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 0 最大値 : 480
DefaultIIOPPassword	デフォルト IIOP ユーザのパスワード。	Administration Console のフィールド表示 : [デフォルト IIOP パスワード] 必須 : いいえ 暗号化 : はい
DefaultIIOPUser	デフォルト IIOP ユーザ。	Administration Console のフィールド表示 : [デフォルト IIOP ユーザ] 必須 : いいえ

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
DefaultInternalServlets Disabled	この属性は、サーブレット エンジンのすべてのデフォルト サーブレットを無効にする。その中には、 weblogic.servlet.ClasspathServlet、 weblogic.servlet.utils.iiop.GetIORServlet、 weblogic.rjvm.http.TunnelSendServlet、 weblogic.rjvm.http.TunnelRecvServlet、 weblogic.rjvm.http.TunnelLoginServlet、 weblogic.rjvm.http.TunnelCloseServlet が含まれる。true に設定すると、このプロパティは ClasspathServletDisabled プロパティをオーバーライドする。	必須: いいえ デフォルト値: false セキュアな値: true
DefaultProtocol	何も指定されていないときに接続に使用されるプロトコルを返す。	Administration Console の フィールド表示: [デフォルト プロトコル] 必須: いいえ デフォルト値: t3
DefaultSecureProtocol	何も指定されていないときにセキュア接続に使用されるプロトコルを返す。	Administration Console の フィールド表示: [デフォルト セキュア プロトコル] 必須: いいえ デフォルト値: t3
DefaultTGIOPPassword	デフォルト TGIOP ユーザのパスワード。	必須: いいえ 暗号化: はい デフォルト値: guest セキュアな値: null
DefaultTGIOPUser	デフォルト TGIOP ユーザ。	必須: いいえ デフォルト値: guest セキュアな値: null

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
DomainLogFilter	<p>このサーバがどのメッセージをドメインログに送るかを決定する。none を指定すると、サーバは ERROR 以上の重要度のすべてのメッセージをドメインログに送る。このリストには、ドメインに対して定義されているすべてのフィルタが含まれている。サーバは 1 つのドメインログフィルタしか使用できない。</p> <p>このプロパティは、[ドメイン ログ ファイルにログを書き込む] が有効の場合にのみ意味を持つ。</p>	<p>Administration Console のフィールド表示: [ドメイン ログ フィルタ]</p> <p>必須: いいえ</p> <p>セキュアな値: none</p>
EnabledForDomainLog	<p>サーバが (自身のログを保持するほかに) ドメイン ログにメッセージを送るかどうかを指定する。</p>	<p>Administration Console のフィールド表示: [ドメイン ログファイルにログを書き込む]</p> <p>必須: いいえ</p> <p>セキュアな値: true</p>
ExpectedToRun	<p>ドメインが起動した場合に、このサーバを実行するかどうかを指定する。</p>	<p>Administration Console のフィールド表示: [実行:]</p> <p>必須: いいえ</p> <p>デフォルト値: true</p>
ExternalDNSName	<p>現在のサーバの外部 DNS 名。これは、http セッションクッキーと動的サーバリストと一緒に http プロキシに送信される。これは、ファイアウォールでネットワークアドレス変換が実行される場合に必要となる。</p>	<p>Administration Console のフィールド表示: [外部 DNS 名]</p> <p>必須: いいえ</p>
ExtraEjbcOptions	<p>動的な ejb のコンパイル時に ejbc に渡される追加オプションを返す。</p>	<p>Administration Console のフィールド表示: [追加 ejbc オプション]</p> <p>必須: いいえ</p>

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
ExtraRmicOptions	サーバサイド生成中に rmic に渡される追加のオプションを返す。	Administration Console の フィールド表示 : [追加 rmic オプション] 必須 : いいえ
HealthCheckIntervalSeconds	サーバの状態チェックの間隔 (単位は秒)。サーバの自動状態モニタとノードマネージャの状態クエリの間隔を指定する。	Administration Console の フィールド表示 : [状態チェック間隔] 必須 : いいえ 単位 : 秒 デフォルト値 : 180 最小値 : 1 最大値 : $2^{31}-1$
HealthCheckTimeoutSeconds	ノードマネージャがサーバの状態クエリをタイムアウトするまでの時間 (単位は秒)。	Administration Console の フィールド表示 : [状態チェックタイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 1 最大値 : $2^{31}-1$

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
HttpTraceSupportEnabled	攻撃者が HTTP TRACE トレース機能を悪用し、HTTP ヘッダ内の情報 (クッキー、認証データなど) にアクセスする場合がある。Web ブラウザに他のクロスドメイン脆弱性が存在する場合、HTTP TRACE メソッドをサポートするすべてのドメインから、機密性の高いヘッダ情報が読み取られるおそれがある。この属性は、HTTP TRACE のサポートを無効にするために使用する。HttpTraceSupportEnabled 属性は、クラスタ全体で使用できるように ClusterMBean と ServerMBean の両方で複製される。ClusterMBean は ServerMBean をオーバーライドする。	必須 : いいえ デフォルト値 : false
HttpdEnabled	プレーンテキストポートまたは SSL ポート上で HTTP サポートを有効にするかどうかを設定する。	Administration Console のフィールド表示 : [HTTPD を有効化] 必須 : いいえ デフォルト値 : true
IIOPEnabled	SSL ポートと非 SSL ポートの両方に対して IIOP サポートを有効にするかどうかを設定する。	Administration Console のフィールド表示 : [IIOP を有効化] 必須 : いいえ デフォルト値 : true
IdleIIOPConnectionTimeout	サーバによって閉じられるまで、IIOP 接続がアイドルであることが許可される最大秒数を指定する。この属性は、開いている接続が多すぎることによって発生するサーバのデッドロックの防止に有用である。	必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 0

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
InstrumentStackTrace Enabled	例外メッセージにサーバサイド スタックトレースを含めるかどうかを指定する。	Administration Console のフィールド表示: [スタックトレースのログ出力] 必須: いいえ デフォルト値: true セキュアな値: false
InterfaceAddress	クラスタのマルチキャストトラフィックを処理する NIC を指定するためのインタフェースアドレスを定義する。	Administration Console のフィールド表示: [インターフェイスアドレス] 必須: いいえ
JDBCLogFileNames	JDBC ログファイルの名前。パス名が絶対パスではない場合、そのパスはサーバが動作しているマシンのルートディレクトリを基準とする相対パスであると見なされる。jdbc.log のようにパス要素を伴わないログファイル名が指定された場合、ネームスペースの競合を防ぐために、ファイルは <code>./SERVER_NAME/</code> のルートディレクトリを基準にした相対位置に配置される。	Administration Console のフィールド表示: [JDBC ログファイル名] 必須: いいえ デフォルト値: jdbc.log
JDBCLoggingEnabled	このサーバが JDBC ログファイルを保持するかどうかを指定する。	Administration Console のフィールド表示: [JDBC ログ記録を有効化] 必須: いいえ デフォルト値: false セキュアな値: true

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
JMSDefaultConnectionFactoriesEnabled	デフォルト JMS 接続ファクトリを有効にする。	Administration Console の フィールド表示: [デフォルト JMS 接続ファクトリを有効化] 必須: いいえ デフォルト値: true
JNDITransportableObjectFactoryList	転送可能なオブジェクトを作成するファクトリのリスト。	必須: いいえ
JavaCompiler	アプリケーションで Java コードをコンパイルする必要があるときに使われる Java コンパイラ。	Administration Console の フィールド表示: [Java コンパイラ] 必須: いいえ デフォルト値: javac セキュアな値: javac
JavaCompilerPostClassPath	Java コードをコンパイルする必要があるときに Java コンパイラのクラスパスの後ろに付加されるオプションを返す。	Administration Console の フィールド表示: [クラスパスの後ろに追加] 必須: いいえ セキュアな値: null
JavaCompilerPreClassPath	Java コードをコンパイルする必要があるときに Java コンパイラのクラスパスの前に付加されるオプションを返す。	Administration Console の フィールド表示: [クラスパスの前に追加] 必須: いいえ セキュアな値: null
ListenAddress	このサーバの TCP リスンアドレス。	Administration Console の フィールド表示: [リスンアドレス] 必須: いいえ
ListenDelaySecs	非推奨。 6.1 との互換性を保持するためにだけ存在する。	必須: いいえ デフォルト値: 0

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
ListenPort	このサーバのプレーンテキスト (非 SSL) リスポート。これが無効になっている場合、SSL ポートを有効にする必要がある。追加のポートをコンフィグレーションするには、ネットワークチャネルを使用する (NetworkChannelMBean を参照) 。クラスタ (マルチキャスト) ポートは別途コンフィグレーションする (getCluster を参照) 。	Administration Console のフィールド表示 : [リスポート] 必須 : いいえ デフォルト値 : 7001 最小値 : 1 最大値 : 65534
ListenPortEnabled	サーバのプレーンテキスト (非 SSL) ポートが有効になっているかどうかを取得する。false に設定した場合、SSL リスポートをコンフィグレーションして有効にする必要がある。	Administration Console のフィールド表示 : [リスポートを有効化] 必須 : いいえ デフォルト値 : true
LogRemoteExceptions Enabled	サーバメッセージログにリモートシステムで発生した例外を含めるかどうかを指定する。	Administration Console のフィールド表示 : [リモート例外のログ出力] 必須 : いいえ デフォルト値 : false セキュアな値 : true
LoginTimeout	使用しない。代わりに、getLoginTimeoutMillis を使用する。	Administration Console のフィールド表示 : [ログインタイムアウト] 必須 : いいえ 単位 : ミリ秒 デフォルト値 : 1000 最小値 : 0

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
LoginTimeoutMillis	サーバのプレーンテキスト (非 SSL) ポートのログインタイムアウト (単位はミリ秒) 。これは、新しい接続が確立するまでの許容最長期間。値 0 は、期間に制限がないことを示す。この値は 0 以上に設定する必要がある。	Administration Console の フィールド表示 : [ログインタイムアウト] 必須 : いいえ 単位 : ミリ秒 デフォルト値 : 5000 セキュアな値 : 5000 最小値 : 0 最大値 : 100000
LowMemoryGCThreshold	粒度レポート レベルに到達したときにサーバが低メモリの警告メッセージをログに記録するしきい値レベル (パーセント値) 。	Administration Console の フィールド表示 : [低メモリ GC しきい値] 必須 : いいえ デフォルト値 : 5 セキュアな値 : 5 最小値 : 0 最大値 : 99
LowMemoryGranularityLevel	低メモリ情報のレポートに使用される粒度レベルを返す。	Administration Console の フィールド表示 : [低メモリ粒度レベル] 必須 : いいえ デフォルト値 : 5 セキュアな値 : 5 最小値 : 1 最大値 : 100

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
LowMemorySampleSize	LowMemoryTimeInterval 用に使用される合計サンプル サイズを返す。デフォルトでは、各 LowMemoryTimeInterval で「10」個のサンプルがとられる。	Administration Console のフィールド表示 : [低メモリ サンプル サイズ] 必須 : いいえ デフォルト値 : 10 最小値 : 1 最大値 : $2^{31}-1$
LowMemoryTimeInterval	時間間隔を返す。コンフィグレーションした間隔ごとに、1 つのサンプルが LowMemorySampleSize まで抽出され、それが繰り返される。	Administration Console のフィールド表示 : [低メモリ時間間隔] 必須 : いいえ 単位 : 秒 デフォルト値 : 3600 最小値 : 300 最大値 : $2^{31}-1$

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
MSFileReplicationEnabled	<p>管理対象サーバのためにコンフィグレーションファイルのレプリケーションを有効にするかどうかを指定する。ファイルのレプリケーションを有効にした場合、管理サーバはそのコンフィグレーションファイルと SerializedSystemIni.dat を 5 分ごとに管理対象サーバのルートディレクトリにコピーする。このオプションでは、起動アイデンティティファイルはレプリケートされない。</p> <p>コンフィグレーションファイルをレプリケートするには、管理対象サーバの独立を有効にする必要がある。</p> <p>別のサーバとインストールまたはルートディレクトリを共有するサーバに対してはファイルのレプリケーションを有効にしない。両方のサーバで予測できないエラーが発生する可能性がある。</p> <p>バックアップ方針、およびドメインのコンフィグレーションの更新頻度によっては、このオプションは大きいファイルをネットワーク上でコピーすることで生じるパフォーマンスコストに見合わない場合がある。</p>	<p>Administration Console のフィールド表示 : [管理対象サーバ独立のファイルレプリケーションを有効化]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p> <p>セキュアな値 : false</p>

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
Machine	<p>このサーバが実行されるマシン。サーバが既に実行されている場合、サーバは別のマシンに移される。指定したマシンとサーバが既に実行されているマシンが同じである場合、これはノーオペレーション命令となる。</p> <p>移行処理では、サーバの停止メソッド (#stop) を使ってマシンを終了し、新しいマシン上でサーバを再起動する。移行中にエラーが発生すると、対応する例外が送出される。</p> <p>マシンを null に設定すると、サーバは起動されない。すでに実行されている場合はシャットダウンされる。</p>	<p>Administration Console のフィールド表示: [マシン]</p> <p>必須: いいえ</p>
ManagedServerIndependenceEnabled	<p>このサーバに対して管理対象サーバの独立を有効にする。管理対象サーバの独立を有効にした場合、管理サーバが使用不能の場合でも管理対象サーバを起動できる。このような場合、管理対象サーバはコンフィグレーションファイルと他のファイルを直接読み込むことによって自身のコンフィグレーションを取得する。</p>	<p>Administration Console のフィールド表示: [管理対象サーバの独立を有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: true</p> <p>セキュアな値: (値 > 0)</p>

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
MaxCOMMessageSize	COM メッセージ全体の最大サイズを指定する。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。この属性を設定しない場合、 maxMessageSize の値が使用される。この設定は、デフォルトポート (ServerMBean setListenPort および setAdministrationPort または SSLMBean setListenPort) のいずれかを使用して開始された接続にのみ適用される。追加ポート上での接続は、 NetworkChannelMBean によってチューニングする	Administration Console のフィールド表示: [最大 COM メッセージサイズ] 必須: いいえ 単位: バイト デフォルト値: 10000000 最小値: 4096 最大値: 2000000000
MaxHTTPMessageSize	メッセージヘッダ内の最大 HTTP メッセージサイズを指定する。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。この属性を設定しない場合、 maxMessageSize の値が使用される。この設定は、デフォルトポート (ServerMBean setListenPort および setAdministrationPort または SSLMBean setListenPort) のいずれかを使用して開始された接続にのみ適用される。追加ポート上での接続は、 NetworkChannelMBean によってチューニングする。	Administration Console のフィールド表示: [最大 HTTP メッセージサイズ] 必須: いいえ 単位: バイト デフォルト値: 10000000 最小値: 4096 最大値: 2000000000

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
MaxIIOPMessageSize	IIOP メッセージ全体の最大サイズを指定する。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。この属性を設定しない場合、 maxMessageSize の値が使用される。	Administration Console のフィールド表示 : [最大 IIOP メッセージサイズ] 必須 : いいえ 単位 : バイト デフォルト値 : 10000000 最小値 : 4096 最大値 : 2000000000
MaxOpenSockCount	特定の時点におけるサーバ内のオープンソケットの許容最大数を返す。このしきい値に達すると、サーバはソケット数がしきい値を下回るまで新しい要求の受け入れを停止する。	Administration Console のフィールド表示 : [最大オープンソケット数] 必須 : いいえ デフォルト値 : -1 最小値 : -1 最大値 : $2^{31}-1$
MaxT3MessageSize	T3 メッセージ全体の最大サイズを指定する。この属性で、サービス拒否攻撃の回避を試みる。サービス拒否攻撃とは、呼び出し側が使用できる以上のメモリを割り当てるようサーバに強制して、他のリクエストへの迅速な応答を妨げようとする攻撃である。この属性を設定しない場合、 maxMessageSize の値が使用される。この設定は、デフォルト ポート (ServerMBean setListenPort および setAdministrationPort または SSLMBean setListenPort) のいずれかを使用して開始された接続にのみ適用される。追加ポート上での接続は、NetworkChannelMBean によってチューニングする。	Administration Console のフィールド表示 : [最大 T3 メッセージサイズ] 必須 : いいえ 単位 : バイト デフォルト値 : 10000000 最小値 : 4096 最大値 : 2000000000

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	MBean の名前を設定する。	Administration Console のフィールド表示: [名前] 必須: いいえ
NativeIOEnabled	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。	Administration Console のフィールド表示: [ネイティブ IO を有効化] 必須: いいえ デフォルト値: true
NetworkAccessPoints	ネットワーク アクセス ポイント (NAP) は、このサーバをターゲットとしたネットワーク チャネルを必要に応じて微調整する。ネットワーク チャネルと NAP は共同で、このサーバがリスンする追加のポートとアドレスを定義する。また、2つのサーバが両方とも特定のプロトコルで同じチャネルをサポートしている場合、それらの間の新しい接続ではそのチャネルが使用される。チャネルがサーバをターゲットとしているが、NAP がコンフィグレーションされていないという場合でも、サーバはそのサーバでコンフィグレーションされたネットワーク情報を使用してそのチャネルを使用する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ
PreferredSecondaryGroup	サーバ上で作成されたプライマリ HTTP セッション ステートのレプリカのホストとして検討されるセカンダリのクラスタ化されたインスタンスを定義する。	Administration Console のフィールド表示: [セカンダリ プリファレンス グループ] 必須: いいえ

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
ReplicationGroup	サーバ上で作成されたプライマリ HTTP セッション ステートのレプリカのホストとして検討される優先的なクラスタ化されたインスタンスを定義する。	Administration Console のフィールド表示 : [レプリケーショングループ] 必須 : いいえ
RestartDelaySeconds	ノード マネージャがサーバを再起動するまでの待機時間 (単位は秒)。この値は、直ちにリスン ポートを再使用することを OS が許可しない場合などに使う。	Administration Console のフィールド表示 : [再開始遅延] 必須 : いいえ 単位 : 秒 デフォルト値 : 0 最小値 : 0 最大値 : $2^{31}-1$
RestartIntervalSeconds	サーバが RestartMax 回再起動できる期間 (単位は秒)。	Administration Console のフィールド表示 : [再起動間隔] 必須 : いいえ 単位 : 秒 デフォルト値 : 3600 最小値 : 300 最大値 : $2^{31}-1$
RestartMax	RestartIntervalSeconds 秒の間にサーバが再起動できる最大数。	Administration Console のフィールド表示 : [期間内の最大再起動回数] 必須 : いいえ デフォルト値 : 2 最小値 : 0 最大値 : $2^{31}-1$

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
ReverseDNSAllowed	カーネルで、DNS の逆引き参照の実行が許可されるかどうかを返す。	Administration Console の フィールド表示 : [許可された リバース DNS] 必須 : いいえ デフォルト値 : false
ServerLifeCycleTimeoutVal	サーバのライフサイクル操作がタイムアウトするまでの秒数。現在は、停止操作と強制停止操作に対してのみ有効。コンフィグレーションされたタイムアウト秒数の間に操作が完了しなかった場合、その時点でのサーバの状態が SHUTTING_DOWN であるか停止操作が実行された場合には、サーバは自動的に停止される。それ以外の場合は、タイムアウトの状況を示すメッセージとともに ServerLifecycleException が送出される。 デフォルト値は、120 秒。0 を指定すると、サーバはライフサイクル操作の完了を無限に待機する。	Administration Console の フィールド表示 : [サーバ ライフサイクル オペレーションの タイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 120 セキュアな値 : 120 最小値 : 0
ServerVersion	サーバのリリース識別子を返す。これはコンフィグレーションされた属性であるため、必ずコンフィグレーションと同じ精度になる。バージョンの形式は、 major.minor.servicepack.rollingpatch 。 バージョンのすべての部分が必須とは限らない。つまり、「7」だけでも有効である。	必須 : いいえ デフォルト値 : unknown

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
StagingMode	アプリケーションの準備中、そのアプリケーションのファイルは管理サーバ上のソースから管理対象サーバのステージング領域にコピーされる。この属性が nostage または external_stage の場合、コピーは行われない。これは、ステージング領域が共有ディレクトリで、すでにアプリケーションファイルが存在する場合か、またはこれがシングルサーバドメインである場合に役立つ。管理者は、管理対象サーバのステージングディレクトリが適切に設定されていることを確認する必要がある。アプリケーションの準備中またはアクティブ化中にそのアプリケーションを使用できない場合、デプロイメントエラーが発生する。この属性は、ApplicationMBean StagingMode 属性でオーバーライドできる。	Administration Console のフィールド表示 : [ステージングモード] 必須 : いいえ
StartupMode	サーバの起動モード。	Administration Console のフィールド表示 : [起動モード] 必須 : いいえ デフォルト値 : RUNNING
StdoutDebugEnabled	サーバがログファイルに加えて標準出力に重大度 DEBUG のメッセージを送信できるようにする。[Stdout ヘログ出力] を有効にする必要がある。	Administration Console のフィールド表示 : [Stdout ヘダバッグ情報出力] 必須 : いいえ デフォルト値 : false セキュアな値 : false

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
StdoutEnabled	サーバがログ ファイルに加えて標準出力にメッセージを送信できるようにする。StdoutDebugEnabled および StdoutSeverityLevel を使用すると、サーバが標準出力へ送るメッセージの種類を決定できる。	Administration Console のフィールド表示 : [Stdout へログ出力] 必須 : いいえ デフォルト値 : true
StdoutFormat	コンソールにログを記録する際に使用される出力形式。	必須 : いいえ デフォルト値 : standard
StdoutLogStack	ログに記録されたメッセージにスタックトレースも含めてコンソールに出力するかどうかを設定する。	必須 : いいえ デフォルト値 : true

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
StdoutSeverityLevel	<p>サーバが標準出力に送信するメッセージの重要度の下限。[Stdout ヘログ出力]を有効にする必要がある。</p> <p>次に重大度を昇順で示す。</p> <ul style="list-style-type: none"> • INFO(64)。通常の処理を報告する。 • WARNING(32)。不審な処理またはコンフィグレーションが行われたが、通常の処理に影響する恐れはない。 • ERROR(16)。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では、サービスの中断や停止なしでエラーに対処できる。 • NOTICE(8)。サーバのモニタにおいて特に重要な INFO レベルまたは WARNING レベルのメッセージ。この重大度のメッセージを書き込むのは WebLogic Server サブシステムのみ。 • CRITICAL(4)。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。 • ALERT(2)。システムの特定のサービスだけが使用不能の状態にある。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。 • EMERGENCY(1)。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。 	<p>Administration Console のフィールド表示: [Stdout 重大度しきい値]</p> <p>必須: いいえ</p> <p>デフォルト値: 16</p> <p>セキュアな値: weblogic.logging.Severities.WARNING</p>

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
StuckThreadMaxTime	いつスレッドが停止したかを調べるために使用される時間を返す。スレッドがこの時間にわたって動作している場合、不正状態になっている可能性がある。	Administration Console の フィールド表示: [スタック ス レッド最大時間] 必須: いいえ 単位: 秒 デフォルト値: 600 最小値: 0 最大値: 2 ³¹ -1
StuckThreadTimerInterval	実行中のスレッドの状態を調べるために使用される間隔を返す。この属性と StuckThreadMaxTime を使用して、スレッドがいつ停止したかを調べる。	Administration Console の フィールド表示: [スタック ス レッドタイマ間隔] 必須: いいえ 単位: 秒 デフォルト値: 600 最小値: 0 最大値: 2 ³¹ -1
TGIOPEnabled	TGIOP サポートを有効にするかどうかを設定する。	必須: いいえ デフォルト値: false セキュアな値: false
ThreadPoolPercentSocketReaders	ソケットリーダーとして使用できるデフォルトキューの実行スレッドの割合 (1 ~ 99) を返す。	Administration Console の フィールド表示: [ソケット リーダー] 必須: いいえ デフォルト値: 33 最小値: 1 最大値: 99

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
TransactionLogFilePrefix	サーバの JTA トランザクション ログファイルのパス プレフィックス。パス名が絶対パスではない場合、そのパスはサーバが動作しているマシンのルート ディレクトリを基準とする相対パスであると見なされる。	Administration Console のフィールド表示 : [トランザクション ログファイルのプレフィックス] 必須 : いいえ デフォルト値 : /

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
TransactionLogFileWritePolicy	<p>ログ記録のディスクへの書き込みで使用するポリシー。「Cache-Flush」ポリシーでは、各書き込みの後にオペレーティングシステムのキャッシュとオンディスク キャッシュがフラッシュされる。「Cache-Flush」はデフォルトのポリシー。「Direct-Write」ポリシーでは、各書き込みで直にディスクに書き込むようにオペレーティングシステムに通知される。「Direct-Write」は「Cache-Flush」よりもパフォーマンスが良く、Windows、HP-UX、および Solaris で利用できる。「Direct-Write」がホスト プラットフォームでサポートされていない場合、ポリシーは「Cache-Flush」になり、ログメッセージが出力される。</p> <p>警告：Windows で「Direct-Write」ポリシーを有効にした場合、Solaris や HP とは異なり、トランザクション データはオンディスク キャッシュに残され、すぐにはディスクに書き込まれない。この場合、停電によってオンディスク キャッシュのデータが消失するおそれがあり、トランザクションとしては安全とはいえない。Windows で「Direct-Write」を使用してトランザクションとして安全な書き込みを行うには、ディスクの書き込みキャッシュをすべて無効にするか、バッテリー バックアップ キャッシュのあるディスクを使用する。</p> <ul style="list-style-type: none"> Windows でハードディスクのオンディスク キャッシュを無効にするには、[コントロール パネル] システム] を選択し、[ハードウェア] タブで [デバイス マネージャ] ボタンをクリックして [ディスク ドライブ] アイコンをダブルクリックする。次に、ドライブの名前をダブルクリックし、[ディスクのプロパティ] タブの [書き込みキャッシュを有効にする] チェック ボックスをチェックする。この値を変更できないファイル システムもある。たとえば、信頼性のあるキャッシュを備えた RAID システム。 	<p>Administration Console の フィールド表示 : [トランザク ション ログ書き込みポリシー] デフォルト値 : Cache-Flush</p>

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
TunnelingClientPingSecs	HTTP トンネリング クライアントがまだ応答しているかどうかを ping する間隔 (単位 : 秒) 。	Administration Console のフィールド表示 : [トンネリング クライアント Ping] 必須 : いいえ 単位 : 秒 デフォルト値 : 45 最小値 : 1
TunnelingClientTimeoutSecs	見つからない HTTP トンネリング クライアントを応答なしと見なすまでの時間 (単位 : 秒) 。	Administration Console のフィールド表示 : [トンネリング クライアント タイムアウト] 必須 : いいえ 単位 : 秒 デフォルト値 : 40 最小値 : 1
TunnelingEnabled	HTTP を介したトンネリングを有効化する。	Administration Console のフィールド表示 : [トンネリングを有効化] 必須 : いいえ デフォルト値 : false
UploadDirectoryName	すべてのアップロードされたアプリケーションが置かれる AdminServer 上のディレクトリパスを指定する。絶対ディレクトリパスを指定しない場合、そのパスはルートディレクトリ (/) を基準とする相対パスになる。デフォルトステージングディレクトリは「stage」(サーバのルートを基準とする)。管理対象サーバでは、この属性は null を返し、コンフィグレーションできない。	Administration Console のフィールド表示 : [アップロードディレクトリ名] 必須 : いいえ セキュアな値 : すべての WebLogic Server インスタンスまたはアプリケーションのルートディレクトリの外にあり、WebLogic Server ホストのシステム ディスクとは別の物理ディスクに存在する絶対ディレクトリ。

表 52-1 Server の属性 (続き)

属性	説明	指定できる値とデフォルト値
VerboseEJBDeploymentEnabled	EJB の冗長デプロイメントを有効化するかどうかを設定する。	必須: いいえ デフォルト値: false
WeblogicPluginEnabled	WLS <code>HttpRequest.getRemoteAddr()</code> では、その戻り値で <code>X-Forwarded-For</code> を利用していた。これは、HTTP ヘッダが容易に模倣可能であり、最終的に不正な値が返されるのでセキュリティホールとなる。この問題はプラグインから独自のヘッダ <code>WL-Proxy-Client-IP</code> を導入することで改善され、このヘッダは WLS がプラグインを使用するようにコンフィグレーションされている場合にのみ使用される。これは <code>ClusterMBean</code> と <code>ServerMBean</code> の両方で複製されるので、属性 <code>WeblogicPluginEnabled</code> はクラスタ全体で使用できる。 <code>ClusterMBean</code> は <code>ServerMBean</code> をオーバーライドする。	Administration Console のフィールド表示: [WebLogic プラグインを有効化] 必須: いいえ デフォルト値: false
XMLEntityCache		必須: いいえ
XMLRegistry		Administration Console のフィールド表示: [XML レジストリ] 必須: いいえ

ServerStart

説明

この **Bean** は、リモートマシン上のサーバを起動するために必要な属性をコンフィグレーションするために使用します。

構文

```
<ServerStart
  Arguments="String"
  BeaHome="String"
  ClassPath="String"
  JavaHome="String"
  Name="String"
  Notes="String"
  Password="String"
  RootDirectory="String"
  SecurityPolicyFile="String"
  Username="String"
/>
```

親要素

- Server

属性

表 53-1 ServerStart の属性

属性	説明	指定できる値とデフォルト値
Arguments	このサーバを起動する際に使用される起動引数。	Administration Console のフィールド表示:[引数] 必須: いいえ
BeaHome	このサーバを起動する際に使用される BEA ホームを返す。ノードマネージャマシン上のパス。	Administration Console のフィールド表示:[BEA Home] 必須: いいえ
ClassPath	このサーバを起動する際に使用されるクラスパス。すべてのパスはノードマネージャマシン上のパス。	Administration Console のフィールド表示:[クラスパス] 必須: いいえ
JavaHome	このサーバの起動時に使用する (ノードマネージャマシン上の) Java のホームディレクトリ。管理対象サーバを起動するための Java バイナリ (Windows では java.exe 実行可能ファイル) は、Java ホームの /bin ディレクトリから取得される。	Administration Console のフィールド表示:[Java ホーム] 必須: いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ

表 53-1 ServerStart の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
Password	サーバの起動時およびサーバ状態モニタの実行時に使用するユーザ名のパスワード。	Administration Console の フィールド表示: [パスワード] 必須: いいえ 暗号化: はい
RootDirectory	このサーバを起動する際に使用される RootDirectory を返す。ノード マネージャ マシン上のパス。	Administration Console の フィールド表示: [ルート ディレクトリ] 必須: いいえ
SecurityPolicyFile	このサーバを起動する際に使用するセキュリティ ポリシー ファイル。ディレクトリとファイル名はノード マネージャ マシン上のパス。	Administration Console の フィールド表示: [セキュリティ ポリシー ファイル] 必須: いいえ
Username	サーバの起動時およびサーバ状態モニタの実行時に使用するユーザ名。	Administration Console の フィールド表示: [ユーザ名] 必須: いいえ

ShutdownClass

説明

停止クラスをコンフィグレーションするためのメソッドを提供します。停止クラスは、**WebLogic Server** インスタンスが正常に停止されるときに自動的にロードされて実行される Java プログラムです。

構文

```
<ShutdownClass
  Arguments="String"
  ClassName="String"
  DeploymentOrder="number"
  Name="String"
  Notes="String"
  Targets="list of Target names"
/>
```

親要素

- Domain

属性

表 54-1 ShutdownClass の属性

属性	説明	指定できる値とデフォルト値
Arguments	サーバがクラスを初期化するとき使用する引数。複数の引数はカンマで区切る。次に例を示す。 <code>first=MyFirstName, last=MyLastName</code>	Administration Console の フィールド表示:[引数] 必須: いいえ
ClassName		Administration Console の フィールド表示:[クラス名]
DeploymentOrder	デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。 デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。	Administration Console の フィールド表示:[デプロイ順] 必須: いいえ デフォルト値: 1000 最小値: 0 最大値: $2^{31}-1$

表 54-1 ShutdownClass の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須 : いいえ

SNMPAgent

説明

SNMP エージェント コンフィグレーションを表す MBean です。SNMPAgent MBean は、SNMP エージェント コンフィグレーションのシングルトンであり、他のどのようなコンフィグレーションにも属しません。

構文

```
<SNMPAgent
  CommunityPrefix="String"
  DebugLevel=( "0" | "1" | "2" | "3" )
  Enabled=( "true" | "false" )
  MibDataRefreshInterval="number of seconds"
  Name="String"
  Notes="String"
  SNMPAttributeChanges="list of SNMPAttributeChange names"
  SNMPCounterMonitors="list of SNMPCounterMonitor names"
  SNMPGaugeMonitors="list of SNMPGaugeMonitor names"
  SNMPLogFilters="list of SNMPLogFilter names"
  SNMPPort="number"
  SNMPProxies="list of SNMPProxy names"
  SNMPStringMonitors="list of SNMPStringMonitor names"
  ServerStatusCheckIntervalFactor="number"
```

```
TargetedTrapDestinations="list of SNMPTrapDestination names"
/>
```

親要素

- Domain

属性

表 55-1 SNMPAgent の属性

属性	説明	指定できる値とデフォルト値
CommunityPrefix	SNMP コミュニティ名の作成に使用されるプレフィックス文字列を返す。	Administration Console のフィールド表示:[コミュニティ プレフィックス] デフォルト値: public
DebugLevel	デバッグ レベルを返す。 <ul style="list-style-type: none"> • 0- デバッグなし • 1- 致命的 • 2- 重要 • 3- 重要ではない 	Administration Console のフィールド表示:[デバッグ レベル] 必須: いいえ デフォルト値: 0
Enabled	SNMP サービスが有効かどうかを示す。	Administration Console のフィールド表示:[有効化] 必須: いいえ デフォルト値: false

表 55-1 SNMPAgent の属性 (続き)

属性	説明	指定できる値とデフォルト値
MibDataRefreshIntervals	すべての MIB 値がキャッシュされてから、エージェントがそれらを更新するまでの最短時間を定義する。	Administration Console のフィールド表示 : [MIB データの更新間隔] 必須 : いいえ 単位 : 秒 デフォルト値 : 120 最小値 : 30 最大値 : 65535
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
SNMPAttributeChanges	属性の変更が観察されたときに属性変更トラップが送信される MBean 型と属性名を記述する SNMPAttributeChangeMBean。	必須 : いいえ
SNMPCounterMonitors	JMX CounterMonitor に基づいてトラップを生成するための基準を記述する SNMP CounterMonitorMbean。	必須 : いいえ
SNMPGaugeMonitors	JMX GaugeMonitor に基づいてトラップを生成するための基準を記述する SNMP GaugeMonitorMbean。	必須 : いいえ
SNMPLogFilters	サーバログ メッセージに基づいてトラップを生成するためのフィルタを記述する SNMPLogFilterMBean。	必須 : いいえ

表 55-1 SNMPAgent の属性 (続き)

属性	説明	指定できる値とデフォルト値
SNMPPort	SNMP トラップ通知を対象の SNMP マネージャに送信する際に使用するポートを定義する。	Administration Console の フィールド表示 : [SNMP ポート] 必須 : いいえ デフォルト値 : 161 最小値 : 1 最大値 : 65535
SNMPProxies	このマスター SNMP エージェントによってプロキシされる SNMP エージェントを定義する。このマスターエージェントによってプロキシされる SNMP エージェントの設定を記述する SNMPProxyMBean。	必須 : いいえ
SNMPStringMonitors	JMX StringMonitor に基づいてトラップを生成するための基準を記述する SNMP StringMonitorMbean。	必須 : いいえ
ServerStatusCheckIntervalFactor	サーバの状態がチェックされる間隔を計算するのに使用される乗数を定義する。 間隔 = n * [MIB データの更新間隔]	Administration Console の フィールド表示 : [サーバ状態 チェック間隔係数] 必須 : いいえ デフォルト値 : 1 最小値 : 1 最大値 : 65535
TargetedTrapDestinations		Administration Console の フィールド表示 : [対象トラップの送り先] 必須 : いいえ

SNMPAttributeChange

説明

このクラスは、MBean の属性変更トラップを受信する設定を記述します。

構文

```
<SNMPAttributeChange
  AttributeMBeanName="String"
  AttributeMBeanType="String"
  AttributeName="String"
  EnabledServers="list of Server names"
  Name="String"
  Notes="String"
/>
```

親要素

- Domain

属性

表 56-1 SNMPAttributeChange の属性

属性	説明	指定できる値とデフォルト値
AttributeMBeanName	モニタする MBean の名前を定義する。	Administration Console の フィールド表示 : [MBean 属性 名] 必須 : いいえ
AttributeMBeanType	モニタする MBean の型を定義する。	Administration Console の フィールド表示 : [MBean 属性 型] 必須 : いいえ
AttributeName	モニタする属性の名前を定義する。	Administration Console の フィールド表示 : [属性名] 必須 : いいえ
EnabledServers	トラップ生成の対象サーバリストを返す。	Administration Console の フィールド表示 : [有効なサーバ] 必須 : いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ

SNMPJMXMonitor

説明

これは、モニターベースのトラップ コンフィグレーション MBean (SNMPCounterMonitorMBean、SNMPStringMonitorMBean、および SNMPGaugeMonitorMBean) の基本クラスです。

構文

```
<SNMPJMXMonitor
  EnabledServers="list of Server names"
  MonitoredAttributeName="String"
  MonitoredMBeanName="String"
  MonitoredMBeanType="String"
  Name="String"
  Notes="String"
  PollingInterval="number of seconds"
/>
```

親要素

- Domain

属性

表 57-1 SNMPJMXMonitor の属性

属性	説明	指定できる値とデフォルト値
EnabledServers	トラップ生成の対象サーバリストを返す。	必須: いいえ
MonitoredAttributeName	モニタする属性の名前を定義する。	Administration Console のフィールド表示: [モニタする属性名] 必須: いいえ
MonitoredMBeanName	モニタする MBean の名前を定義する。	Administration Console のフィールド表示: [モニタする MBean 名] 必須: いいえ
MonitoredMBeanType	モニタする MBean の型を定義する。	Administration Console のフィールド表示: [モニタする MBean 型] 必須: いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ

表 57-1 SNMPJMXMonitor の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
PollingInterval	エージェントが属性値をチェックする頻度を定義する。	Administration Console の フィールド表示: [ポーリング 間隔] 必須: いいえ 単位: 秒 デフォルト値: 0 最小値: 0 最大値: 65535

SNMPProxy

説明

現在の SNMP エージェントによってプロキシされる SNMP エージェントを表す MBean です。

構文

```
<SNMPProxy
  Community="String"
  Name="String"
  Notes="String"
  OldRoot="String"
  Port="number"
  Timeout="number of milliseconds"
/>
```

親要素

- Domain

属性

表 58-1 SNMPProxy の属性

属性	説明	指定できる値とデフォルト値
Community	プロキシされたエージェントのすべてのリクエストに渡されるコミュニティ名を返す。	Administration Console の フィールド表示 : [コミュニティ] 必須 : いいえ デフォルト値 : なし
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
OidRoot	プロキシされたエージェントが応答する OID ツリーのルートを定義する。	Administration Console の フィールド表示 : [Oid ルート] 必須 : いいえ

表 58-1 SNMPProxy の属性 (続き)

属性	説明	指定できる値とデフォルト値
Port	他の SNMP エージェントとの通信に使用するポート番号を定義する。	Administration Console の フィールド表示 : [ポート] 必須 : いいえ デフォルト値 : 0 最小値 : 0 最大値 : 65535
Timeout	WebLogic SNMP プロキシエージェントが、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待つ間隔を定義する。指定された時間が経過しても応答がなかった場合、WebLogic SNMP エージェントはリクエスト元のマネージャにエラーを送信する。	Administration Console の フィールド表示 : [タイムアウト] 必須 : いいえ 単位 : ミリ秒 デフォルト値 : 5000 最小値 : 0

SNMPTrapDestination

説明

この MBean では、SNMP トラップが送信できるすべての送り先を記述します。

構文

```
<SNMPTrapDestination
  Community="String"
  Host="String"
  Name="String"
  Notes="String"
  Port="number"
/>
```

親要素

- Domain

属性

表 59-1 SNMPTrapDestination の属性

属性	説明	指定できる値とデフォルト値
Community	トラップ通知を対象の SNMP マネージャに送るためのパスワードとして機能する SNMP トラップ名を返す。	Administration Console の フィールド表示: [コミュニティ] 必須: いいえ デフォルト値: public
Host	SNMP マネージャがあるマシンのホスト名または IP アドレスを返す。	Administration Console の フィールド表示: [ホスト] 必須: いいえ デフォルト値: localhost
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
Port	SNMP トラップ通知を対象の SNMP マネージャに送信する際に使用するポートを定義する。	Administration Console の フィールド表示: [ポート] 必須: いいえ デフォルト値: 162 最小値: 1 最大値: 65535

SNMPTrapSource

説明

この MBean は、SNMP トラップ関連のコンフィグレーション MBean の基本になります。

構文

```
<SNMPTrapSource
  EnabledServers=list of Server names
  Name=String
  Notes=String
/>
```

親要素

- Domain

属性

表 60-1 SNMPTrapSource の属性

属性	説明	指定できる値とデフォルト値
EnabledServers	トラップ生成の対象サーバリストを返す。	必須: いいえ
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ

SSL

説明

この MBean は、WebLogic Server バージョン 6.x および 7.0 デプロイメントでの SSL プロトコルのコンフィグレーションを表します。

構文

```
<SSL
  CertAuthenticator="String"
  CertificateCacheSize="number"
  Ciphersuites="list of Strings"
  ClientCertificateEnforced=( "true" | "false" )
  Enabled=( "true" | "false" )
  ExportKeyLifespan="number"
  HandlerEnabled=( "true" | "false" )
  HostnameVerificationIgnored=( "true" | "false" )
  HostnameVerifier="String"
  KeyEncrypted=( "true" | "false" )
  ListenPort="number"
  LoginTimeoutMillis="number of milliseconds"
  Name="String"
  Notes="String"
  ServerCertificateChainFileName="String"
```

SSL

```
ServerCertificateFileName="String"  
ServerKeyFileName="String"  
ServerPrivateKeyAlias="String"  
ServerPrivateKeyPassPhrase="String"  
TrustedCAFileName="String"  
TwoWaySSLEnabled=( "true" | "false" )  
UseJava=( "true" | "false" )  
</>
```

親要素

- Server

属性

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
CertAuthenticator	<p><code>weblogic.security.acl.CertAuthenticator</code> クラスを実装する Java クラスの名前。このクラスは WebLogic Server ユーザにクライアントのデジタル証明書をマップする。<code>weblogic.security.acl.CertAuthenticator</code> クラスは、クライアントによって提示されたデジタル証明書を検証した後で、WebLogic Server が呼び出す <code>authenticate()</code> メソッドを備えている。</p>	<p>Administration Console のフィールド表示: [証明書認証プロバイダ]</p> <p>必須: いいえ</p> <p>セキュアな値: <code>weblogic.security.acl.CertAuthenticator</code></p>
CertificateCacheSize	<p>トークンによって使用されていない証明書の数。この属性は読み込み専用である。</p>	<p>Administration Console のフィールド表示: [証明書キャッシュ サイズ]</p> <p>必須: いいえ</p> <p>デフォルト値: 3</p> <p>最小値: 1</p> <p>最大値: $2^{31}-1$</p>

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
Ciphersuites	<p>特定の WebLogic Server で使用される暗号スイートを指定する。 指定できる値は次のとおり。</p> <ul style="list-style-type: none"> • SSL_NULL_WITH_NULL_NULL • SSL_RSA_WITH_NULL_SHA • SSL_RSA_EXPORT_WITH_RC4_40_MD5 • SSL_RSA_WITH_RC4_128_MD5 • SSL_RSA_WITH_RC4_128_SHA • SSL_RSA_EXPORT_WITH_DES_40_CBC_SHA • SSL_RSA_WITH_DES_CBC_SHA • SSL_RSA_WITH_3DES_EDE_CBC_SHA • SSL_DH_anon_EXPORT_WITH_RC4_40_MD5 • SSL_DH_anon_WITH_RC4_128_MD5 • SSL_DH_anon_EXPORT_WITH_DES_40_CBC_SHA • SSL_DH_anon_WITH_DES_CBC_SHA • SSL_DH_anon_WITH_3DES_EDE_CBC_SHA <p>デフォルトは SSL_RSA_EXPORT_WITH_RC4_40_MD5。</p>	<p>必須: いいえ</p>
ClientCertificateEnforced	<p>クライアントが信頼される認証局から取得したデジタル証明書を WebLogic Server に提示する必要があるかどうかを定義する。</p>	<p>Administration Console の フィールド表示: [クライアント認証を強制] 必須: いいえ デフォルト値: false</p>

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
Enabled	SSLMBean で SSL ポートが有効化されているかどうかを示す。この属性を無効にした場合、このサーバに対してプレーンテキスト (非 SSL) リスポートを有効にする必要がある。追加ポートは、を使用してコンフィグレーション可能。	Administration Console のフィールド表示: [SSL リスポートを有効化 (SSL をコンフィグレーションしてください)] 必須: いいえ デフォルト値: false
ExportKeyLifespan	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。	Administration Console のフィールド表示: [エクスポート キーの有効期間] 必須: いいえ デフォルト値: 500 最小値: 1 最大値: 2 ³¹ -1
HandlerEnabled	現在は使用されていないため無視。	Administration Console のフィールド表示: Handler Enabled 必須: いいえ デフォルト値: true
HostnameVerificationIgnored	インストール済みの weblogic.security.SSL.HostnameVerifier クラス実装が有効かどうかを示す。	Administration Console のフィールド表示: [ホスト名検証を無視] 必須: いいえ デフォルト値: false

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
HostnameVerifier	<p>weblogic.security.SSL.HostnameVerifier クラスを実装するクラスの名前。このクラスは、SSL クライアントから受け取った URL 内のホスト名がサーバ証明書の識別名の共通名と一致しているかどうかを検証する。このクラスは、介在者の攻撃を防ぐために使用する。</p> <p>weblogic.security.SSL.HostnameVerifier クラスは、WebLogic Server が SSL ハンドシェイク時にクライアントに対して呼び出す verify() メソッドを持つ。</p>	<p>Administration Console のフィールド表示: [ホスト名の検証]</p> <p>必須: いいえ</p> <p>セキュアな値: weblogic.security.acl.CertAuthenticator</p>
KeyEncrypted	<p>WebLogic Server 用のプライベートキーをパスワードで暗号化するかどうかを指定する。</p> <ul style="list-style-type: none"> この属性が true に設定される場合、キーを使用するためにプライベートキーからパスワードを提供するように要求される。 この属性が false に設定される場合、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。 <p>この属性は読み込み専用である。</p>	<p>Administration Console のフィールド表示: [暗号化キーを使用]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p> <p>セキュアな値: true</p>
ListenPort	<p>WebLogic Server が SSL 接続要求をリスンする TCP/IP ポート。</p>	<p>Administration Console のフィールド表示: [SSL リスンポート]</p> <p>必須: いいえ</p> <p>デフォルト値: 7002</p> <p>最小値: 1</p> <p>最大値: 65535</p>

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
LoginTimeoutMillis	<p>WebLogic Server が SSL 接続のタイムアウトまで待機するミリ秒数を指定する。SSL 接続は、通常の接続よりも時間がかかる。</p> <p>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に対応するためにデフォルト値を大きくする。0 の値は属性を無効にする。</p>	<p>Administration Console のフィールド表示: [SSL ログインタイムアウト]</p> <p>必須: いいえ</p> <p>単位: ミリ秒</p> <p>デフォルト値: 25000</p> <p>最小値: 1</p> <p>最大値: 2³¹-1</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>必須: いいえ</p>
ServerCertificateChainFileName	<p>非推奨。 WebLogic Server によって信頼された認証局の順序付けされたリストを格納したファイルの絶対パスと名前。ファイルの拡張子 .PEM は、ファイルの読み込み方法を示す。</p> <p>WebLogic Server 用のデジタル証明書は、このファイルに格納してはならないことに注意する。チェーンに含まれる証明書が 2 つを超える場合、チェーン全体を PEM フォーマットでこのファイルに含めるべきである。</p>	<p>Administration Console のフィールド表示: [サーバ証明書チェーン ファイル名]</p> <p>必須: いいえ</p> <p>デフォルト値: server-certchain.pem</p>
ServerCertificateFileName	<p>WebLogic Server のデジタル証明書の完全なディレクトリの位置と名前。WebLogic Server は、ファイルの拡張子 (.DER または .PEM) によってファイルの内容の読み込み方法を判別する。</p>	<p>Administration Console のフィールド表示: [サーバ証明書ファイル名]</p> <p>必須: いいえ</p> <p>デフォルト値: server-cert.der</p>

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
ServerKeyFileName	非推奨。WebLogic Server のプライベート キーの絶対パスと名前。ファイルの拡張子 (.PEM) は、ファイルの読み込み方法を示す。	Administration Console のフィールド表示: [サーバキーファイル名] 必須: いいえ デフォルト値: server-key.der
ServerPrivateKeyAlias	サーバのプライベート キーをキーストアに格納し、キーストアから取得するための文字列エイリアス。このプライベート キーは、サーバのデジタル証明書に関連付けられ、通常サーバのホスト名によって格納される。	Administration Console のフィールド表示: [サーバプライベートキーのエイリアス] 必須: いいえ
ServerPrivateKeyPass Phrase	サーバのプライベート キーをキーストアから取得するためのパスワード。パスワードは、生成時にプライベートキーに割り当てられる。	Administration Console のフィールド表示: [サーバプライベートキーの Pass Phrase] 必須: いいえ 暗号化: はい
TrustedCAFileName	PEM エンコードされた信頼性のある認証局の証明書を格納したファイルの名前。	Administration Console のフィールド表示: [信頼性のある CA ファイル名] 必須: いいえ デフォルト値: trusted-ca.pem

表 61-1 SSL 属性

属性	説明	指定できる値とデフォルト値
TwoWaySSLEnabled	サーバがクライアントからの証明書を要求することを指定する。ただし、クライアントが証明書を提示しない場合、SSL 接続は続行される。	Administration Console の フィールド表示：[クライアント証明書を要求(強制しない)] 必須：いいえ デフォルト値：false セキュアな値：true
UseJava	ネイティブ Java ライブラリの使用を有効にする。WebLogic Server は、SSL プロトコルの pure-Java 実装を提供する。Solaris、Windows NT、および IBM AIX プラットフォームでは、ネイティブ ライブラリによって SSL 処理のパフォーマンスが向上する。この属性は読み込み専用である。	Administration Console の フィールド表示：[Java を使用] 必須：いいえ デフォルト値：true

StartupClass

説明

起動クラスをコンフィグレーションします。起動クラスは、**WebLogic Server** インスタンスが起動または再起動するときに自動的にロードされて実行される **Java** プログラムです。

デフォルトでは、起動クラスは、他のサーバサブシステムが初期化され、サーバがモジュールをデプロイした後に、自動的にロードされて実行されます。起動クラスに対しては、デフォルトをオーバーライドして、サーバが **JDBC** 接続プールをデプロイする前、および **Web** アプリケーションと **EJB** をデプロイする前にロードして実行するよう指定することもできます。

構文

```
<StartupClass
  Arguments="String"
  ClassName="String"
  DeploymentOrder="number"
  FailureIsFatal=( "true" | "false" )
  LoadBeforeAppActivation=( "true" | "false" )
  LoadBeforeAppDeployments=( "true" | "false" )
  Name="String"
  Notes="String"
```

```
Targets="list of Target names"
/>
```

親要素

- Domain

属性

表 62-1 StartupClass の属性

属性	説明	指定できる値とデフォルト値
Arguments	サーバがクラスを初期化するとき使用する引数。複数の引数はカンマで区切る。次に例を示す。 <code>first=MyFirstName,last=MyLastName</code>	Administration Console のフィールド表示:[引数] 必須: いいえ
ClassName		Administration Console のフィールド表示:[クラス名]
DeploymentOrder	デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。 デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。	Administration Console のフィールド表示:[デプロイ順] 必須: いいえ デフォルト値: 1000 最小値: 0 最大値: 2 ³¹ -1

表 62-1 StartupClass の属性

属性	説明	指定できる値とデフォルト値
FailureIsFatal	この起動クラスで障害が発生したらサーバを起動しないようにするかどうかを指定する。このチェックボックスがチェックされていない場合(またはAPIを使用して <code>false</code> の値を指定している場合)は、この起動クラスで障害が発生しても、サーバは起動処理を続行する。	Administration Console のフィールド表示:[失敗は致命的] 必須:いいえ デフォルト値: <code>false</code>
LoadBeforeAppActivation	<p>起動クラスを接続プールの作成後、アプリケーションがアクティブになる前にロードすべきかどうかを指定する。アクティブ化は、2 フェーズデプロイメントモデルの第2フェーズ。</p> <p>アプリケーションがデプロイメントでできるようになる前に起動クラスを呼び出す必要がある場合は LoadBeforeAppDeployments を使用する。接続プールはこの段階ではまだ作成されていない。</p> <p>接続プールが利用可能になった後、アプリケーションがアクティブ化されてクライアントの要求に対応できるようになる前に起動クラスを呼び出す必要がある場合は、LoadBeforeAppActivation を使用する。</p>	必須:いいえ デフォルト値: <code>false</code>

表 62-1 StartupClass の属性

属性	説明	指定できる値とデフォルト値
LoadBeforeAppDeployments	<p>サーバが JMS サービスおよび JDBC サービスを作成する前、またはアプリケーションおよび EJB をデプロイする前に起動クラスがロードされて実行されるかどうかを指定する。このオプションで true を指定すると、サーバは 2 フェーズデプロイメントモデルの prepare() フェーズの前に起動クラスをロードして実行する。この時点で、JMS サービスと JDBC サービスはまだ利用できず、アプリケーションや EJB もデプロイされていない。</p> <p>false を指定すると、サーバは他のすべてのタイプのモジュールがデプロイされた後に起動クラスをロードする。</p>	<p>Administration Console のフィールド表示 : [アプリケーションデプロイメントの前に実行]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須 : いいえ</p>

UnixMachine

説明

この Bean は、UNIX オペレーティング システムが稼働しているマシンを表します。これは、UNIX プラットフォーム固有のプロパティで **MachineMBean** を拡張したものです。

構文

```
<UnixMachine
  Addresses="list of Strings"
  Name="String"
  Notes="String"
  PostBindGID="String"
  PostBindGIDEnabled=( "true" | "false" )
  PostBindUID="String"
  PostBindUIDEnabled=( "true" | "false" )
/>
```

親要素

- Domain

子要素

- NodeManager

属性

表 63-1 UnixMachine の属性

属性	説明	指定できる値とデフォルト値
Addresses	このマシンが認知されるアドレスを返す。ホスト名またはリテラル IP アドレス。	Administration Console のフィールド表示:[アドレス] 必須: いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示:[名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示:[メモ] 必須: いいえ

表 63-1 UnixMachine の属性 (続き)

属性	説明	指定できる値とデフォルト値
PostBindGID	このマシンで稼動するサーバが、すべての特権起動アクションの実行後に動作する際の UNIX GID を返す。この値を設定した場合、これは有効な UNIX GID。設定しない場合は null。	Administration Console の フィールド表示 : [Post-Bind GID] 必須 : いいえ デフォルト値 : nobody
PostBindGIDEnabled	このマシンで稼動するサーバが、すべての特権起動アクションの実行後に動作する際の UNIX UID を返す。この値を設定した場合、これは有効な UNIX UID。設定しない場合は null。	Administration Console の フィールド表示 : [Post-Bind GID を有効化] 必須 : いいえ デフォルト値 : false セキュアな値 : true
PostBindUID	このマシンで稼動するサーバが、すべての特権起動アクションの実行後に動作する際の UNIX UID を返す。この値を設定した場合、これは有効な UNIX UID。設定しない場合は null。	Administration Console の フィールド表示 : [Post-Bind UID] 必須 : いいえ デフォルト値 : nobody
PostBindUIDEnabled	このマシンで稼動するサーバが、すべての特権起動アクションの実行後に動作する際の UNIX UID を返す。この値を設定した場合、これは有効な UNIX UID。設定しない場合は null。	Administration Console の フィールド表示 : [Post-Bind UID を有効化] 必須 : いいえ デフォルト値 : false

UnixRealm

説明

構文

```
<UnixRealm  
  AuthProgram="String"  
  Name="String"  
  Notes="String"  
>
```

親要素

- Domain

属性

表 64-1 UnixRealm の属性

属性	説明	指定できる値とデフォルト値
AuthProgram	<p>UNIX セキュリティ レalm でユーザを認証するためのプログラムの名前。ほとんどの場合、プログラム名には <code>wlauth</code> を指定する。この認証プログラムは、<code>setuid root</code> を実行する必要がある。</p> <ul style="list-style-type: none"> プログラム名が <code>wlauth</code> で、<code>CLASSPATH</code> に含まれている場合、この属性を明示的に設定する必要がないので、空白にしておく。 プログラム名が <code>wlauth</code> 以外の場合、または WebLogic Server の <code>CLASSPATH</code> に含まれていない場合は、この属性を指定する。 	<p>Administration Console のフィールド表示 : [認証プログラム]</p> <p>必須 : いいえ</p> <p>デフォルト値 : <code>wlauth</code></p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>

VirtualHost

説明

この Bean は、WebLogic Server 内の仮想 Web サーバのコンフィグレーションを表します。サーバは、仮想ホストをサポートするために複数の Web サーバを定義する場合があります。

この MBean は、仮想ホストを表します。

構文

```
<VirtualHost
  AcceptContextPathInGetRealPath=( "true" | "false" )
  AuthCookieEnabled=( "true" | "false" )
  Charsets="java.util.Map"
  ChunkedTransferDisabled=( "true" | "false" )
  ClusteringEnabled=( "true" | "false" )
  DefaultServerName="String"
  DefaultWebApp="WebAppComponent name"
  DeploymentOrder="number"
  FrontendHTTPPort="number"
  FrontendHTTPSPort="number"
  FrontendHost="String"
  HttpsKeepAliveSecs="number of seconds"
```

VirtualHost

```
KeepAliveEnabled=( "true" | "false" )
KeepAliveSecs="number of seconds"
LogFileBufferKBytes="number of kilobytes"
LogFileFlushSecs="number of seconds"
LogFileFormat=( "common" | "extended" )
LogFileName="String"
LogRotationPeriodMins="number of minutes"
LogRotationTimeBegin="String"
LogRotationType=( "size" | "date" )
LogTimeInGMT=( "true" | "false" )
LoggingEnabled=( "true" | "false" )
MaxLogFileSizeKBytes="number of kilobytes"
MaxPostSize="number of bytes"
MaxPostTimeSecs="number of seconds"
Name="String"
Notes="String"
PostTimeoutSecs="number"
SendServerHeaderEnabled=( "true" | "false" )
SingleSignonDisabled=( "true" | "false" )
Targets="list of Target names"
URLResource="java.util.Map"
UseHeaderEncoding=( "true" | "false" )
UseHighestCompatibleHTTPVersion=( "true" | "false" )
VirtualHostNames="list of Strings"
WAPEnabled=( "true" | "false" )
/>
```

親要素

- Domain

属性

表 65-1 VirtualHost の属性

属性	説明	指定できる値とデフォルト値
AcceptContextPathInGetRealPath	このリリースから context.getRealPath() を呼び出す際に、virtualPath 内に contextPath を含めることは許されなくなった。サブディレクトリが contextPath と同名の場合にこのケースが崩れるためである。古い動作に従って開発された古いアプリケーションをサポートするために、互換性スイッチを提供する。このスイッチは将来のリリースでは非推奨になる予定。	Administration Console のフィールド表示:[実際のパスによるコンテキストパスを許可] 必須: いいえ デフォルト値: false
AuthCookieEnabled	追加のセキュア AuthCookie を使用して、セキュリティ制約のある https ページへのアクセスをよりセキュアにできるようにする。セッションクッキーは、アクセスを得るのに十分ではない。	必須: いいえ デフォルト値: false セキュアな値: true
Charsets	インターネットと Java 文字セット名の間ユーザ定義マッピング。	必須: いいえ
ChunkedTransferDisabled	HTTP/1.1 のチャンク転送エンコーディングを使用できないようにする。	必須: いいえ デフォルト値: false
ClusteringEnabled	HTTP のクラスタ化を有効にする。	必須: いいえ デフォルト値: false セキュアな値: true

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
DefaultServerName	HTTP defaultServerName を設定する。	Administration Console の フィールド表示 : [デフォルト サーバ名] 必須 : いいえ
DefaultWebApp	デフォルトのサーブレット コンテキスト (ContextPath = "/") にマップする サーブレット 2.3 Web アプリケーショ ン。	Administration Console の フィールド表示 : [デフォルト Web アプリケーション] 必須 : いいえ
DeploymentOrder	デプロイの優先順位。サーバではこれ を使って項目のデプロイ順を決定す る。優先順位は、同じタイプの他のデ プロイ可能な項目に対する相対的な順 位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてか ら、起動クラスを優先順位に従ってデ プロイする。 デプロイ順の値が小さい項目から順に デプロイされる。デプロイ順の値が等 しい項目のデプロイメント順序は保証 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。	必須 : いいえ デフォルト値 : 1000 最小値 : 0 最大値 : $2^{31}-1$
FrontendHTTPPort	frontendHTTPPort を設定する。	必須 : いいえ デフォルト値 : 0
FrontendHTTPSPort	frontendHTTPSPort を設定する。	必須 : いいえ デフォルト値 : 0
FrontendHost	HTTP frontendHost を設定する。	必須 : いいえ

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
HttpsKeepAliveSecs	リクエストをタイムアウトするまで HTTPS キープアライブを維持する秒数。	Administration Console の フィールド表示 : [Https Keep Alive 時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 セキュアな値 : 60 最小値 : 30 最大値 : 360
KeepAliveEnabled	HTTP キープアライブを有効にするかどうかを返す。	Administration Console の フィールド表示 : [Keep Alive を 有効化] 必須 : いいえ デフォルト値 : true セキュアな値 : true
KeepAliveSecs	リクエストをタイムアウトするまで HTTP キープアライブを維持する秒数。	Administration Console の フィールド表示 : [Keep Alive 時 間] 必須 : いいえ 単位 : 秒 デフォルト値 : 30 セキュアな値 : 30 最小値 : 5 最大値 : 300

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogFileBufferKBytes	HTTP リクエストを格納するバッファの最大サイズ (単位は KB)。バッファがこのサイズに達すると、データが HTTP ログ ファイルに書き込まれる。サーバがバッファ サイズをチェックする頻度を指定するには、 LogFileFlushSecs プロパティを使う。	Administration Console のフィールド表示 : [ログファイル バッファ サイズ] 必須 : いいえ 単位 : KB デフォルト値 : 8 最小値 : 0 最大値 : 1024
LogFileFlushSecs	サーバが HTTP リクエスト バッファのサイズをチェックする間隔 (単位は秒)。バッファのサイズが LogFileBufferKBytes プロパティで指定されたサイズを超えると、サーバはバッファ内のデータを HTTP リクエスト ログ ファイルに書き込む。	Administration Console のフィールド表示 : [ログファイル 更新間隔] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 最小値 : 1 最大値 : 360
LogFileFormat	HTTP ログ ファイルのフォーマットを指定する。どちらのフォーマットも W3C で定義されている。拡張ログフォーマットでは、ログ ファイルでサーバの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。	Administration Console のフィールド表示 : [ログファイル フォーマット] 必須 : いいえ デフォルト値 : common

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogFileNames	<p>HTTP リクエスト ログを格納するファイルの名前。パス名が絶対パスでない場合、パスはサーバのルート ディレクトリを基準にしている。</p> <p>この値は、HTTP ログが有効化されている場合にのみ意味を持つ。</p> <p>現在のログ ファイルは常に、この属性の値と等しい名前を持つファイル。ログ ファイルのローテーションを有効にすると、現在のファイルがサイズまたは時間の制限に達したときに、ファイルの名前が変更される。</p> <p>ログ ファイルのローテーション時に日付と時刻をファイル名に含めるには、<code>java.text.SimpleDateFormat</code> 変数をファイル名に追加する。各変数はパーセント (%) 記号で囲む。</p> <p>たとえば、ファイル名が <code>access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log</code> として定義されている場合、ログ ファイルの名前は <code>access_yyyy_mm_dd_hh_mm.log</code> となる。</p> <p>ログ ファイルがローテーションされると、ローテーションされたファイルの名前には日付が含まれる。たとえば、ログ ファイルが 2003 年 4 月 2 日の午前 10:05 にローテーションされた場合、古いメッセージの含まれるログ ファイルは <code>access_2003_04_02_10_05.log</code> という名前になる。</p> <p>日付と時刻をファイル名に含めない場合、ローテーションされるログ ファイルは作成の順に番号付けされる。たとえば、<code>access.log0007</code> のようになる。</p>	<p>Administration Console のフィールド表示 : [ログファイル名]</p> <p>必須 : いいえ</p> <p>デフォルト値 : <code>access.log</code></p>

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogRotationPeriodMins	古い HTTP リクエストが別のログファイルに移される間隔 (単位は分)。この値は、date ベースのローテーションタイプを使用する場合にのみ意味を持つ。	Administration Console のフィールド表示 : [ローテーション間隔] 必須 : いいえ 単位 : 分 デフォルト値 : 1440 最小値 : 1 最大値 : 2 ³¹ -1
LogRotationTimeBegin	時間ベースのローテーションの開始時間を指定する。この値によって指定された間隔で、現在のログファイル名が変更される。以後、LogRotationPeriodMins に指定した間隔でログファイル名が変更される。次のフォーマットを使用する。 MM-dd-yyyy-k:mm:ss <ul style="list-style-type: none"> • MM はグレゴリオ暦の月 • dd は日 • yyyy は年 • k は 24 時間形式の時 • mm は分 • ss は秒 指定した時刻が既に過ぎている場合は、ファイルのローテーションが直ちに開始される。 デフォルトでは、ローテーションはサーバインスタンスを再起動してから 24 時間後に開始される。	Administration Console のフィールド表示 : [ローテーション開始時間] 必須 : いいえ

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogRotationType	<p>古い HTTP リクエストを別のログファイルに移すための条件。</p> <ul style="list-style-type: none"> • <code>size</code>。ログファイルが <code>MaxLogFileSizeKBytes</code> に指定したサイズに達すると、ファイル名が <code>LogFileName.n</code> に変更される。 • <code>date</code>。 <code>LogRotationPeriodMin</code> に指定した間隔で、ファイル名が <code>LogFileName.n</code> に変更される。 <p>ファイル名が変更されると、以後のメッセージは <code>LogFileName</code> に指定した名前の新しいファイルに蓄積される。</p>	<p>Administration Console のフィールド表示 : [ローター ションタイプ]</p> <p>必須 : いいえ</p> <p>デフォルト値 : <code>size</code></p>
LogTimeInGMT	<p>ホスト コンピュータが指定したローカルの時間帯に関係なく、HTTP ログメッセージのタイムスタンプがグリニッジ標準時 (GMT) であるかどうかを指定する。</p> <p>このメソッドを使用すると、拡張フォーマットのログファイルに関する W3C の仕様に準拠できる。その仕様では、拡張フォーマットのログエントリのタイムスタンプはすべて GMT であることが明記されている。</p> <p>このメソッドは、<code>extended</code> メッセージフォーマットが指定されている場合のみ適用される。</p>	<p>Administration Console のフィールド表示 : [時間のログ (GMT)]</p> <p>必須 : いいえ</p> <p>デフォルト値 : <code>false</code></p>

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
LoggingEnabled	HTTP リクエストのログを有効化する。	Administration Console の フィールド表示 : [ログ記録を 有効化] 必須 : いいえ デフォルト値 : true セキュアな値 : true
MaxLogFileSizeKBytes	サーバがログ メッセージを独立した ファイルに移動することになるサイズ (1 ~ 65535 KB)。ログ ファイルが指定 されたサイズに達した後、サーバが次 にファイルサイズをチェックしたとき に、現在のログ ファイル名が <i>FileName.n</i> に変更され、それ以降の メッセージを格納する新しいファイル が作成される。 0 を指定すると、ファイルサイズは無 限に大きくなる。 このプロパティは、ファイルを <code>size(</code> <code>サイズ)</code> でローテーションする場合に のみ意味を持つ。	Administration Console の フィールド表示 : [最大ログ ファイル サイズ] 必須 : いいえ 単位 : KB デフォルト値 : 0 最小値 : 0
MaxPostSize	サブレット リクエスト中の HTTP POST データの読み込みに対する最大 POST サイズ (単位はバイト)。 MaxPostSize < 0 は無制限。	Administration Console の フィールド表示 : [最大 POST サイズ] 必須 : いいえ 単位 : バイト デフォルト値 : -1
MaxPostTimeSecs	サブレット リクエスト中の HTTP POST データの読み込みに対する最大 POST 時間 (単位は秒)。MaxPostTime < 0 は無制限。	Administration Console の フィールド表示 : [最大 POST 時間] 必須 : いいえ 単位 : 秒 デフォルト値 : -1

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ
PostTimeoutSecs	サーブレット リクエスト中の HTTP POST データの読み込みに対するタイムアウト (単位は秒) 。	Administration Console の フィールド表示 : [POST タイムアウト秒] 必須 : いいえ デフォルト値 : 30 セキュアな値 : 30 最小値 : 0 最大値 : 120
SendServerHeaderEnabled	このサーバインスタンスがその名前と WebLogic Server のバージョン番号を HTTP 応答ヘッダに含めるかどうかを指定する。この情報を提供すると、攻撃者が特定バージョンの WebLogic Server の脆弱性を知っている場合にセキュリティ上のリスクが生じることになる。	必須 : いいえ デフォルト値 : false セキュアな値 : false
SingleSignonDisabled	Web アプリケーションでシングルサインオンを無効にする。	必須 : いいえ デフォルト値 : false
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	Administration Console の フィールド表示 : [対象] 必須 : いいえ
URLResource	JNDI に URL 接続ファクトリ リソースを追加する。	必須 : いいえ

表 65-1 VirtualHost の属性 (続き)

属性	説明	指定できる値とデフォルト値
UseHeaderEncoding	ISO-8859_1 以外の文字が含まれることがわかっている特定のヘッダで Content-Type エンコーディングを使用できるようにする。	必須: いいえ デフォルト値: false セキュアな値: false
UseHighestCompatible HTTPVersion	最も互換性が高い HTTP プロトコルバージョン文字列を応答で使用できるようにする。たとえば、HTTP の仕様では、HTTP/1.0 を使用したリクエストに対する応答で HTTP/1.1 バージョン文字列を使用すべきことが提案されている。これは必ずしも応答のフォーマットに影響しない。	必須: いいえ デフォルト値: false
VirtualHostNames	Web サーバが代わりにリクエストを処理するホスト名を返す。	Administration Console のフィールド表示: [仮想ホスト名] 必須: いいえ
WAPEnabled	WAP を有効化する。	Administration Console のフィールド表示: [WAP を有効化] 必須: いいえ デフォルト値: false

WebAppComponent

説明

Weblogic Server インスタンスにデプロイされる J2EE Web アプリケーションをコンフィグレーションするためのメソッドを提供します。このインタフェースは、Web アプリケーションをデプロイするときのみインスタンス化されます。

このインタフェースは、WAR ファイルまたは展開ディレクトリとしてデプロイされる Web アプリケーションをコンフィグレーションできます。

構文

```
<WebAppComponent
  AuthFilter="String"
  AuthRealmName="String"
  DeploymentOrder="number"
  IndexDirectoryEnabled=( "true" | "false" )
  Name="String"
  Notes="String"
  PreferWebInfClasses=( "true" | "false" )
  ServletReloadCheckSecs="number"
  SessionMonitoringEnabled=( "true" | "false" )
  SingleThreadedServletPoolSize="number"
  Targets="list of Target names"
```

```
URI="String"  
VirtualHosts="list of VirtualHost names"  
</>
```

親要素

- Application

属性

表 66-1 WebAppComponent の属性

属性	説明	指定できる値とデフォルト値
AuthFilter	すべての認証および認可のチェックが Web アプリケーションで実行される前後に呼び出される AuthFilter サブレット クラスを設定する。	必須 : いいえ
AuthRealmName	ブラウザにポップアップ表示される [Basic Authentication HTTP] ダイアログ ボックスのレルムを設定する。	Administration Console のフィールド表示 : [認証レルム名] 必須 : いいえ デフォルト値 : weblogic

表 66-1 WebAppComponent の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>Administration Console のフィールド表示 : [デプロイ順]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 1000</p> <p>最小値 : 0</p> <p>最大値 : 2³¹-1</p>
IndexDirectoryEnabled	<p>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。</p>	<p>Administration Console のフィールド表示 : [インデックス ディレクトリ]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示 : [メモ]</p> <p>必須 : いいえ</p>
PreferWebInfClasses	<p>有効の場合、Web アプリケーションの WEB-INF ディレクトリ内のクラスが、アプリケーションまたはシステム クラスローダにロードされているクラスに優先してロードされる。</p>	<p>Administration Console のフィールド表示 : [Web Inf クラスを優先]</p> <p>必須 : いいえ</p> <p>デフォルト値 : false</p>

表 66-1 WebAppComponent の属性 (続き)

属性	説明	指定できる値とデフォルト値
ServletReloadCheckSeconds	サーブレットが変更されているかどうかを WebLogic がチェックする頻度。変更されている場合は再ロードする。-1 の場合は再ロードせず、0 の場合は常に再ロードする。	Administration Console のフィールド表示 : [再ロード間隔 (秒)] 必須 : いいえ デフォルト値 : 1
SessionMonitoringEnabled	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。	Administration Console のフィールド表示 : [セッションモニタを有効化] 必須 : いいえ デフォルト値 : false
SingleThreadedServletPoolSize	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。	Administration Console のフィールド表示 : [シングルスレッドサーブレットのプールサイズ] 必須 : いいえ デフォルト値 : 5
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	Administration Console のフィールド表示 : [対象] 必須 : いいえ
URI	アプリケーション コンポーネントを示す URI を設定する (通常は管理サーバ上にある)。	Administration Console のフィールド表示 : [URI]
VirtualHosts	このデプロイメントの対象になる仮想ホストを返す。	Administration Console のフィールド表示 : [仮想ホスト] 必須 : いいえ

WebServer

説明

この Bean は、WebLogic Server 内の仮想 Web サーバのコンフィグレーションを表します。サーバは、仮想ホストをサポートするために複数の Web サーバを定義する場合があります。

この MBean は、仮想ホストを表します。

構文

```
<WebServer
  AcceptContextPathInGetRealPath=( "true" | "false" )
  AuthCookieEnabled=( "true" | "false" )
  Charsets="java.util.Map"
  ChunkedTransferDisabled=( "true" | "false" )
  ClusteringEnabled=( "true" | "false" )
  DefaultServerName="String"
  DefaultWebApp="WebAppComponent name"
  DeploymentOrder="number"
  FrontendHTTPPort="number"
  FrontendHTTPSPort="number"
  FrontendHost="String"
  HttpsKeepAliveSecs="number of seconds"
```

```
KeepAliveEnabled=( "true" | "false" )
KeepAliveSecs="number of seconds"
LogFileBufferKBytes="number of kilobytes"
LogFileFlushSecs="number of seconds"
LogFileFormat=( "common" | "extended" )
LogFileName="String"
LogRotationPeriodMins="number of minutes"
LogRotationTimeBegin="String"
LogRotationType=( "size" | "date" )
LogTimeInGMT=( "true" | "false" )
LoggingEnabled=( "true" | "false" )
MaxLogFileSizeKBytes="number of kilobytes"
MaxPostSize="number of bytes"
MaxPostTimeSecs="number of seconds"
Name="String"
Notes="String"
PostTimeoutSecs="number"
SendServerHeaderEnabled=( "true" | "false" )
SingleSignonDisabled=( "true" | "false" )
Targets="list of Target names"
URLResource="java.util.Map"
UseHeaderEncoding=( "true" | "false" )
UseHighestCompatibleHTTPVersion=( "true" | "false" )
WAPEnabled=( "true" | "false" )
/>
```

親要素

- Server

属性

表 67-1 WebServer の属性

属性	説明	指定できる値とデフォルト値
AcceptContextPathInGetRealPath	このリリースから context.getRealPath() を呼び出す際に、virtualPath 内に contextPath を含めることは許されなくなった。サブディレクトリが contextPath と同名の場合にこのケースが崩れるためである。古い動作に従って開発された古いアプリケーションをサポートするために、互換性スイッチを提供する。このスイッチは将来のリリースでは非推奨になる予定。	Administration Console のフィールド表示: [実際のパスによるコンテキストパスを許可] 必須: いいえ デフォルト値: false
AuthCookieEnabled	追加のセキュア AuthCookie を使用して、セキュリティ制約のある https ページへのアクセスをよりセキュアにできるようにする。セッションクッキーは、アクセスを得るのに十分ではない。	必須: いいえ デフォルト値: false セキュアな値: true
Charsets	インターネットと Java 文字セット名の間ユーザ定義マッピング。	必須: いいえ
ChunkedTransferDisabled	HTTP/1.1 のチャンク転送エンコーディングを使用できないようにする。	必須: いいえ デフォルト値: false
ClusteringEnabled	HTTP のクラスタ化を有効にする。	必須: いいえ デフォルト値: false セキュアな値: false
DefaultServerName	HTTP defaultServerName を設定する。	Administration Console のフィールド表示: [デフォルトサーバ名] 必須: いいえ

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
DefaultWebApp	デフォルトのサーブレット コンテキスト (ContextPath="/") にマップするサーブレット 2.3 Web アプリケーション。	Administration Console の フィールド表示: [デフォルト Web アプリケーション] 必須: いいえ
DeploymentOrder	デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。 デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。	必須: いいえ デフォルト値: 1000 最小値: 0 最大値: 2 ³¹ -1
FrontendHTTPPort	frontendHTTPPort を設定する。	Administration Console の フィールド表示: [フロントエ ンド HTTP ポート] 必須: いいえ デフォルト値: 0
FrontendHTTPSPort	frontendHTTPSPort を設定する。	Administration Console の フィールド表示: [フロントエ ンド HTTPS ポート] 必須: いいえ デフォルト値: 0
FrontendHost	HTTP frontendHost を設定する。	Administration Console の フィールド表示: [フロントエ ンドホスト] 必須: いいえ

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
HttpsKeepAliveSecs	リクエストをタイムアウトするまで HTTPS キープアライブを維持する秒数。	Administration Console の フィールド表示 : [HTTPS 持続 時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 60 セキュアな値 : 60 最小値 : 30 最大値 : 360
KeepAliveEnabled	HTTP キープアライブを有効にするかどうかを返す。	Administration Console の フィールド表示 : [Keep Alive を有効化] 必須 : いいえ デフォルト値 : true セキュアな値 : true
KeepAliveSecs	リクエストをタイムアウトするまで HTTP キープアライブを維持する秒数。	Administration Console の フィールド表示 : [持続時間] 必須 : いいえ 単位 : 秒 デフォルト値 : 30 セキュアな値 : 30 最小値 : 5 最大値 : 300

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogFileBufferKBytes	HTTP リクエストを格納するバッファの最大サイズ (単位は KB)。バッファがこのサイズに達すると、データが HTTP ログ ファイルに書き込まれる。サーバがバッファ サイズをチェックする頻度を指定するには、 <code>LogFileFlushSecs</code> プロパティを使う。	Administration Console のフィールド表示: [ログ バッファ サイズ] 必須: いいえ 単位: KB デフォルト値: 8 最小値: 0 最大値: 1024
LogFileFlushSecs	サーバが HTTP リクエスト バッファのサイズをチェックする間隔 (単位は秒)。バッファのサイズが <code>LogFileBufferKBytes</code> プロパティで指定されたサイズを超えると、サーバはバッファ内のデータを HTTP リクエスト ログ ファイルに書き込む。	Administration Console のフィールド表示: [更新間隔] 必須: いいえ 単位: 秒 デフォルト値: 60 最小値: 1 最大値: 360
LogFileFormat	HTTP ログ ファイルのフォーマットを指定する。どちらのフォーマットも W3C で定義されている。拡張ログフォーマットでは、ログ ファイルでサーバの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。	Administration Console のフィールド表示: [フォーマット] 必須: いいえ デフォルト値: common

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogFile	<p>HTTP リクエスト ログを格納するファイルの名前。パス名が絶対パスでない場合、パスはサーバのルート ディレクトリを基準にしている。</p> <p>この値は、HTTP ログが有効化されている場合にのみ意味を持つ。</p> <p>現在のログ ファイルは常に、この属性の値と等しい名前を持つファイル。ログ ファイルのローテーションを有効にすると、現在のファイルがサイズまたは時間の制限に達したときに、ファイルの名前が変更される。</p> <p>ログ ファイルのローテーション時に日付と時刻をファイル名に含めるには、<code>java.text.SimpleDateFormat</code> 変数をファイル名に追加する。各変数はパーセント (%) 記号で囲む。</p> <p>たとえば、ファイル名が <code>access_%yyyy%_%MM%_%dd%_%hh%_%mm%.log</code> として定義されている場合、ログ ファイルの名前は <code>access_yyyy_mm_dd_hh_mm.log</code> となる。</p> <p>ログ ファイルがローテーションされると、ローテーションされたファイルの名前には日付が含まれる。たとえば、ログ ファイルが 2003 年 4 月 2 日の午前 10:05 にローテーションされた場合、古いメッセージの含まれるログ ファイルは <code>access_2003_04_02_10_05.log</code> という名前になる。</p> <p>日付と時刻をファイル名に含めない場合、ローテーションされるログ ファイルは作成の順に番号付けされる。たとえば、<code>access.log0007</code> のようになる。</p>	<p>Administration Console のフィールド表示: [ログファイル名]</p> <p>必須: いいえ</p> <p>デフォルト値: <code>access.log</code></p>

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogRotationPeriodMins	古い HTTP リクエストが別のログ ファイルに移される間隔 (単位は分)。この値は、date ベースのローテーションタイプを使用する場合にのみ意味を持つ。	Administration Console のフィールド表示 : [ローテーション間隔] 必須 : いいえ 単位 : 分 デフォルト値 : 1440 最小値 : 1 最大値 : $2^{31}-1$
LogRotationTimeBegin	時間ベースのローテーションの開始時間を指定する。この値によって指定された間隔で、現在のログ ファイル名が変更される。以後、LogRotationPeriodMins に指定した間隔でログ ファイル名が変更される。次のフォーマットを使用する。 MM-dd-yyyy-k:mm:ss <ul style="list-style-type: none"> • MM はグレゴリオ暦の月 • dd は日 • yyyy は年 • k は 24 時間形式の時 • mm は分 • ss は秒 指定した時刻が既に過ぎている場合は、ファイルのローテーションが直ちに開始される。 デフォルトでは、ローテーションはサービンスタンスを再起動してから 24 時間後に開始される。	Administration Console のフィールド表示 : [ローテーション時間] 必須 : いいえ

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
LogRotationType	<p>古い HTTP リクエストを別のログ ファイルに移すための条件。</p> <ul style="list-style-type: none"> • size。ログ ファイルが <code>MaxLogFileSizeKBytes</code> に指定したサイズに達すると、ファイル名が <code>LogFileName.n</code> に変更される。 • date。 <code>LogRotationPeriodMin</code> に指定した間隔で、ファイル名が <code>LogFileName.n</code> に変更される。 <p>ファイル名が変更されると、以後のメッセージは <code>LogFileName</code> に指定した名前の新しいファイルに蓄積される。</p>	<p>Administration Console のフィールド表示: [ローテーションタイプ]</p> <p>必須: いいえ</p> <p>デフォルト値: <code>size</code></p>
LogTimeInGMT	<p>ホスト コンピュータが指定したローカルの時間帯に関係なく、HTTP ログメッセージのタイムスタンプがグリニッジ標準時 (GMT) であるかどうかを指定する。</p> <p>このメソッドを使用すると、拡張フォーマットのログファイルに関する W3C の仕様に準拠できる。その仕様では、拡張フォーマットのログエントリのタイムスタンプはすべて GMT であることが明記されている。</p> <p>このメソッドは、<code>extended</code> メッセージフォーマットが指定されている場合のみ適用される。</p>	<p>Administration Console のフィールド表示: [時間のログ (GMT)]</p> <p>必須: いいえ</p> <p>デフォルト値: <code>false</code></p>
LoggingEnabled	<p>HTTP リクエストのログを有効化する。</p>	<p>Administration Console のフィールド表示: [ログを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: <code>true</code></p> <p>セキュアな値: <code>true</code></p>

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
MaxLogFileSizeKBytes	<p>サーバがログメッセージを独立したファイルに移動することになるサイズ (1 ~ 65535 KB)。ログファイルが指定されたサイズに達した後、サーバが次にファイルサイズをチェックしたときに、現在のログファイル名が <i>FileName.n</i> に変更され、それ以降のメッセージを格納する新しいファイルが作成される。</p> <p>0 を指定すると、ファイルサイズは無限に大きくなる。</p> <p>このプロパティは、ファイルを <code>size(サイズ)</code> でローテーションする場合にのみ意味を持つ。</p>	<p>Administration Console のフィールド表示: [最大ログファイルサイズ]</p> <p>必須: いいえ</p> <p>単位: KB</p> <p>デフォルト値: 0</p> <p>最小値: 0</p>
MaxPostSize	<p>サブレットリクエスト中の HTTP POST データの読み込みに対する最大 POST サイズ (単位はバイト)。</p> <p>MaxPostSize < 0 は無制限。</p>	<p>Administration Console のフィールド表示: [最大 POST サイズ]</p> <p>必須: いいえ</p> <p>単位: バイト</p> <p>デフォルト値: -1</p>
MaxPostTimeSecs	<p>サブレットリクエスト中の HTTP POST データの読み込みに対する最大 POST 時間 (単位は秒)。</p> <p>MaxPostTime < 0 は無制限。</p>	<p>Administration Console のフィールド表示: [最大 POST 時間]</p> <p>必須: いいえ</p> <p>単位: 秒</p> <p>デフォルト値: -1</p>
Name	<p>このコンフィグレーションの名前。</p> <p>WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ
PostTimeoutSecs	サーブレットリクエスト中の HTTP POST データの読み込みに対するタイムアウト (単位は秒)。	Administration Console のフィールド表示: [POST タイムアウト秒] 必須: いいえ デフォルト値: 30 セキュアな値: 30 最小値: 0 最大値: 120
SendServerHeaderEnabled	このサービインスタンスがその名前と WebLogic Server のバージョン番号を HTTP 応答ヘッダに含めるかどうかを指定する。この情報を提供すると、攻撃者が特定バージョンの WebLogic Server の脆弱性を知っている場合にセキュリティ上のリスクが生じることになる。	Administration Console のフィールド表示: [Send Server Header を有効化] 必須: いいえ デフォルト値: false セキュアな値: false
SingleSignonDisabled	Web アプリケーションでシングルサインオンを無効にする。	必須: いいえ デフォルト値: false
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須: いいえ
URLResource	JNDI に URL 接続ファクトリリソースを追加する。	必須: いいえ
UseHeaderEncoding	ISO-8859_1 以外の文字が含まれることがわかっている特定のヘッダで Content-Type エンコーディングを使用できるようにする。	必須: いいえ デフォルト値: false セキュアな値: false

表 67-1 WebServer の属性 (続き)

属性	説明	指定できる値とデフォルト値
UseHighestCompatible HTTPVersion	最も互換性が高い HTTP プロトコルバージョン文字列を応答で使用できるようにする。たとえば、HTTP の仕様では、HTTP/1.0 を使用したリクエストに対する応答で HTTP/1.1 バージョン文字列を使用すべきことが提案されている。これは必ずしも応答のフォーマットに影響しない。	必須: いいえ デフォルト値: false
WAPEnabled	WAP を有効化する。	Administration Console の フィールド表示: [WAP を有効化] 必須: いいえ デフォルト値: false

WebServiceComponent

説明

Weblogic Server インスタンスにデプロイされる Web サービスをコンフィグレーションするためのインタフェースを提供します。

Web サービスは、Web サービスを記述する追加のデプロイメント記述子 (`web-services.xml`) が含まれている特殊な J2EE Web アプリケーションです。Web サービスは J2EE Web アプリケーションとしてパッケージ化されるので、そのコンポーネント MBean は Web アプリケーションのそれと同じであり、したがって新しいメソッドは追加せずに単純に `WebAppComponentMBean` インタフェースを拡張したものです。

このインタフェースは、Web サービスをデプロイするときのみインスタンス化されません。

このインタフェースでは、WAR ファイルまたは展開ディレクトリとしてデプロイされる Web サービスをコンフィグレーションできます。

構文

```
<WebServiceComponent  
  AuthFilter="String"  
  AuthRealmName="String"
```

```
DeploymentOrder="number"  
IndexDirectoryEnabled=( "true" | "false" )  
Name="String"  
Notes="String"  
PreferWebInfClasses=( "true" | "false" )  
ServletReloadCheckSecs="number"  
SessionMonitoringEnabled=( "true" | "false" )  
SingleThreadedServletPoolSize="number"  
Targets="list of Target names"  
URI="String"  
VirtualHosts="list of VirtualHost names"  
</>
```

親要素

- Application

属性

表 68-1 WebServiceComponent の属性

属性	説明	指定できる値とデフォルト値
AuthFilter	すべての認証および認可のチェックが Web アプリケーションで実行される前に呼び出される AuthFilter サブレットクラスを設定する。	必須：いいえ
AuthRealmName	ブラウザにポップアップ表示される [Basic Authentication HTTP] ダイアログ ボックスのレルムを設定する。	Administration Console のフィールド表示：[認証レルム名] 必須：いいえ デフォルト値：weblogic

表 68-1 WebServiceComponent の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>Administration Console のフィールド表示: [デプロイ順]</p> <p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: 2³¹-1</p>
IndexDirectoryEnabled	<p>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。</p>	<p>Administration Console のフィールド表示: [インデックス ディレクトリの有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>必須: いいえ</p>
PreferWebInfClasses	<p>有効の場合、Web アプリケーションの WEB-INF ディレクトリ内のクラスが、アプリケーションまたはシステム クラスローダにロードされているクラスに優先してロードされる。</p>	<p>必須: いいえ</p> <p>デフォルト値: false</p>

表 68-1 WebServiceComponent の属性 (続き)

属性	説明	指定できる値とデフォルト値
ServletReloadCheckSeconds	サーブレットが変更されているかどうかを WebLogic がチェックする頻度。変更されている場合は再ロードする。-1 の場合は再ロードせず、0 の場合は常に再ロードする。	Administration Console のフィールド表示: [サーブレット再ロードのチェック間隔 (秒)] 必須: いいえ デフォルト値: 1
SessionMonitoringEnabled	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。	Administration Console のフィールド表示: [セッションモニタを有効化] 必須: いいえ デフォルト値: false
SingleThreadedServletPoolSize	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。	Administration Console のフィールド表示: [シングル スレッドサーブレットのプールサイズ] 必須: いいえ デフォルト値: 5
Targets	現在のドメイン内で、この項目のデプロイ先となる対象。	必須: いいえ
URI	アプリケーション コンポーネントを示す URI を設定する (通常は管理サーバにある)。	Administration Console のフィールド表示: [URI]
VirtualHosts	このデプロイメントの対象になる仮想ホストを返す。	必須: いいえ

WLECCConnectionPool

説明

この Bean は、WLEC 接続プールを定義します。

構文

```
<WLECCConnectionPool
  ApplicationPassword="String"
  CertificateAuthenticationEnabled=( "true" | "false" )
  DeploymentOrder="number"
  FailoverAddresses="list of Strings"
  MaximumEncryptionLevel="number"
  MaximumPoolSize="number"
  MinimumEncryptionLevel="number"
  MinimumPoolSize="number"
  Name="String"
  Notes="String"
  PrimaryAddresses="list of Strings"
  SecurityContextEnabled=( "true" | "false" )
  Targets="list of Target names"
  UserName="String"
  UserPassword="String"
  UserRole="String"
```

```
WLEDomain="String"  
</>
```

親要素

- Domain

属性

表 69-1 WLEConnectionPool の属性

属性	説明	指定できる値とデフォルト値
ApplicationPassword	アプリケーションのパスワードを定義する。このフィールドは、Tuxedo 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 の場合にのみ必須。	Administration Console のフィールド表示: [アプリケーションパスワード] 必須: いいえ 暗号化: はい
CertificateAuthenticationEnabled	証明書認証の状態を定義する。 <ul style="list-style-type: none"> 証明書認証を使用する場合、WLEC では UserName および ApplicationPassword フィールドの値を使用して、WLEC の証明書を取得する。 証明書認証を使用しない場合、WLEC では Tuxedo ドメインのセキュリティ レベルに応じて、パスワード認証を使用するか、または認証を使用しない。 パスワード認証が必要な場合、WLEC では UserName および UserPassword フィールドの値を使用して認証を行う。 	Administration Console のフィールド表示: [証明書を有効化] 必須: いいえ デフォルト値: false セキュアな値: true

表 69-1 WLECConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: $2^{31}-1$</p>
FailoverAddresses	<p>プライマリアドレスで定義された接続が確立できない場合、または失敗した場合に使用される IIOP リスナ/ハンドラアドレスのリストを定義する。複数のアドレスはセミコロンで区切る。</p>	<p>Administration Console のフィールド表示: [フェイルオーバーアドレス]</p> <p>必須: いいえ</p>
MaximumEncryptionLevel	<p>Tuxedo ドメインと WebLogic Server の間で使用される SSL の最大暗号化レベルを定義する。</p> <ul style="list-style-type: none"> 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、および 128 は暗号キーの長さ (ビット単位) を指定する。 デフォルトは、暗号化パッケージキットのライセンスで許可された最小のレベル。 この最小暗号化レベルが一致しない場合、Tuxedo と WebLogic Server の間の SSL 接続は失敗する。 	<p>Administration Console のフィールド表示: [最大暗号化レベル]</p> <p>必須: いいえ</p> <p>デフォルト値: 0</p>

表 69-1 WLECConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
MaximumPoolSize	WLEC 接続プールから開始できる IIOP 接続の最大数を定義する。	Administration Console の フィールド表示: [最大プール サイズ] 必須: いいえ デフォルト値: 1
MinimumEncryptionLevel	Tuxedo ドメインと WebLogic Server の間で使用される SSL の最小暗号化レベルを定義する。 <ul style="list-style-type: none"> 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、および 128 は暗号キーの長さ (ビット単位) を指定する。 デフォルト値は 40。 この最小暗号化レベルが一致しない場合、Tuxedo と WebLogic Server の間の SSL 接続は失敗する。 	Administration Console の フィールド表示: [最小暗号化 レベル] 必須: いいえ デフォルト値: 0
MinimumPoolSize	WebLogic Server の起動時に WLEC 接続プールに追加する IIOP 接続の数を定義する。	Administration Console の フィールド表示: [最小プール サイズ] 必須: いいえ デフォルト値: 1
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示: [メモ] 必須: いいえ

表 69-1 WLECConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
PrimaryAddresses	<p>WLEC 接続プールと Tuxedo ドメインの間の接続を確立するために使用する IIOP リスナ/ハンドラのアドレスのリストを定義する。</p> <ul style="list-style-type: none"> 各アドレスの形式は //hostname:port。 アドレスは、UBBCONFIG ファイルに定義されている ISL アドレスと一致しなければならない。複数のアドレスはセミコロンで区切る。たとえば //main1.com:1024;//main2.com:1044 のようにする。 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/ハンドラアドレスに corbalocs プレフィックスを付ける。たとえば、corbalocs://hostname:port のようにする。 	<p>Administration Console のフィールド表示: [プライマリアドレス]</p> <p>必須: いいえ</p>
SecurityContextEnabled	<p>WebLogic Server ユーザから Tuxedo ドメインに渡されるセキュリティコンテキストの状態を定義する。選択した場合、セキュリティコンテキストは有効。</p>	<p>Administration Console のフィールド表示: [セキュリティコンテキストを有効化]</p> <p>必須: いいえ</p> <p>デフォルト値: false</p> <p>セキュアな値: true</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須: いいえ</p>

表 69-1 WLECConnectionPool の属性 (続き)

属性	説明	指定できる値とデフォルト値
UserName	有効なユーザの名前を定義する。このフィールドは、Tuxedo ドメインのセキュリティ レベルが USER_AUTH、ACL、または MANDATORY_ACL の場合にのみ必須。	Administration Console の フィールド表示: [ユーザ名] 必須: いいえ
UserPassword	UserName フィールドで指定した有効なユーザのパスワードを定義する。このフィールドは UserName フィールドを定義する場合にのみ必須。	Administration Console の フィールド表示: [ユーザ パスワード] 必須: いいえ 暗号化: はい
UserRole	この接続プールのユーザ ロールを定義する。このフィールドは、Tuxedo 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 の場合にのみ必須。	Administration Console の フィールド表示: [ユーザ ロール] 必須: いいえ
WLEDomain	プールが接続される WLEC ドメインの名前を定義する。 <ul style="list-style-type: none"> WLEC 接続プールは、1つの Tuxedo ドメインにつき 1つしか定義できない。 ドメイン名は、Tuxedo ドメインの UBBCONFIG ファイルにある RESOURCES セクションの domainid パラメータに一致しなければならない。 	Administration Console の フィールド表示: [ドメイン] 必須: いいえ

WTCEXport

説明

このインタフェースは、WTC エクスポート コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの WTC コンフィグレーションに適用されます。

構文

```
<WTCEXport
  EJBName="String"
  LocalAccessPoint="String"
  Name="String"
  Notes="String"
  RemoteName="String"
  ResourceName="String"
/>
```

親要素

- WTCServer

属性

表 70-1 WTCEXport の属性

属性	説明	指定できる値とデフォルト値
EJBName	<p>サービスを呼び出すときに使用される EJB ホーム インタフェースの完全な名前を定義する。指定しない場合、デフォルトのインタフェースとして <code>tuxedo.services.servicenameHome</code> が使用される。</p> <p>次に例を示す。呼び出されるサービスが <code>TOUPPER</code> で、<code>EJBName</code> 属性を指定しない場合、<code>JNDI</code> でルックアップされるホーム インタフェースは <code>tuxedo.services.TOUPPERHome</code> になる。</p>	<p>Administration Console のフィールド表示 : [EJB 名]</p> <p>必須 : いいえ</p>
LocalAccessPoint	<p>サービスをエクスポートするローカル アクセス ポイントの名前を定義する。</p>	<p>Administration Console のフィールド表示 : [ローカル アクセス ポイント]</p>
Name	<p>このコンフィグレーションの名前。<code>WebLogic Server</code> では <code>MBean</code> を使用してコンフィグレーションを実装および保持する。</p>	<p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>必須 : いいえ</p>

表 70-1 WTCEXport の属性 (続き)

属性	説明	指定できる値とデフォルト値
RemoteName	サービスのリモート名を定義する。指定しない場合、 ResourceName 属性が使用される。	Administration Console の フィールド表示:[リモート名] 必須: いいえ
ResourceName	エクスポートされたサービスを識別するための名前を定義する。 ResourceName 属性と LocalAccessPoint 属性の組み合わせは、 WTCEXportMbean の中でユニークでなければならない。	Administration Console の フィールド表示:[リソース名]

WTCImport

説明

このインタフェースは、WTC インポート コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの WTC コンフィグレーションに適用されます。

構文

```
<WTCImport
  LocalAccessPoint="String"
  Name="String"
  Notes="String"
  RemoteAccessPointList="String"
  RemoteName="String"
  ResourceName="String"
/>
```

親要素

- WTCServer

属性

表 71-1 WTCImport の属性

属性	説明	指定できる値とデフォルト値
LocalAccessPoint	サービスを提供するローカルアクセスポイントの名前を定義する。	Administration Console のフィールド表示: [ローカルアクセスポイント]
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
RemoteAccessPointList	リソースがインポートされるリモートドメインアクセスポイントを示す、カンマ区切りのフェイルオーバーリストを定義する。 例: TDOM1,TDOM2,TDOM3	Administration Console のフィールド表示: [リモートアクセスポイントリスト]

表 71-1 WTCImport の属性 (続き)

属性	説明	指定できる値とデフォルト値
RemoteName	サービスのリモート名を定義する。指定しない場合、 ResourceName 属性が使用される。	Administration Console の フィールド表示 : [リモート名] 必須 : いいえ
ResourceName	インポートされたサービスを識別するための名前を定義する。 ResourceName 属性、 LocalAccessPoint 属性、および RemoteAccessPointList 属性の組み合わせは、 WTCImportMbean の中でユニークでなければならない。	Administration Console の フィールド表示 : [リソース名]

WTCLocalTuxDom

説明

このインタフェースは、WTC ローカル Tuxedo ドメイン コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの WTC コンフィグレーションに適用されます。

構文

```
<WTCLocalTuxDom
  AccessPoint="String"
  AccessPointId="String"
  BlockTime="number"
  CmpLimit="number"
  ConnPrincipalName="String"
  ConnectionPolicy=( "ON_DEMAND" | "ON_STARTUP" | "INCOMING_ONLY" )
  Interoperate="String"
  MaxEncryptBits=( "0" | "40" | "56" | "128" )
  MaxRetries="number"
  MinEncryptBits=( "0" | "40" | "56" | "128" )
  NWAddr="String"
  Name="String"
  Notes="String"
  RetryInterval="number"
```

```
Security=( "NONE" | "APP_PW" | "DM_PW" )
/>
```

親要素

- WTCServer

属性

表 72-1 WTCLocalTuxDom の属性

属性	説明	指定できる値とデフォルト値
AccessPoint	WTCServer MBean でドメインを識別するために使用する名前を定義する。この名前は WTCServer MBean における WTCLocalTuxDom および WTCRemoteTuxDom の AccessPoint 名のスコープ内でユニークでなければならない。	Administration Console のフィールド表示:[アクセス ポイント]
AccessPointId	別のドメインへの接続を確立するとき、ドメインを識別するための接続プリンシパル名を定義する。 WTCLocalTuxDom MBean の AccessPointId は、Tuxedo DMCONFIG ファイルの *DM_REMOTE_DOMAINS セクションにある対応する DOMAINID と一致する必要がある。	Administration Console のフィールド表示:[アクセス ポイント ID]

表 72-1 WTCLocalTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
BlockTime	ブロッキング呼び出しで許可される最大待機時間 (秒) を定義する。	Administration Console の フィールド表示 : [ブロッキング タイムアウト] 必須 : いいえ デフォルト値 : 60 最小値 : 0 最大値 : $2^{31}-1$
CmpLimit	リモート ドメインへのデータの送信に使用する圧縮しきい値を定義する。このサイズより大きいアプリケーションバッファは圧縮される。	Administration Console の フィールド表示 : [圧縮制限] 必須 : いいえ デフォルト値 : $2^{31}-1$ 最小値 : 0 最大値 : $2^{31}-1$
ConnPrincipalName	別のドメインへの接続を確立するとき、このドメインの識別子を検証するために使用されるプリンシパル名を指定する。 <ul style="list-style-type: none"> このパラメータは、BEA Tuxedo 7.1 以降のソフトウェアを実行しているタイプ TDOMAIN のドメインにのみ適用される。 指定しない場合、接続プリンシパル名はこのドメインの AccessPointID にデフォルト設定される。 注意 : ConnectionPrincipalName はこのリリースではサポートされていない。	Administration Console の フィールド表示 : [接続プリン シパル名] 必須 : いいえ

表 72-1 WTCLocalTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnectionPolicy	<p>ローカル ドメインがリモート ドメインとの接続の確立を試行する場合の条件を定義する。</p> <ul style="list-style-type: none"> • ON_DEMAND: リモート サービスへのクライアント リクエストまたは管理接続コマンドのいずれかによってリクエストされたときのみ、接続が試行される。 • ON_STARTUP: ドメイン ゲートウェイは、ゲートウェイ サーバの初期化時にそのリモート ドメイン アクセス ポイントで接続を確立する。リモート サービス (このローカル アクセス ポイントのドメイン ゲートウェイによって JNDI に通知されるサービス) は、接続がそのリモート ドメイン アクセス ポイントに正常に確立されたときのみ通知される。リモート ドメイン アクセス ポイントへのアクティブな接続がない場合、リモート サービスはサスペンドされる。デフォルトでは、この接続ポリシーは失敗した接続を 60 秒ごとに再試行する。アプリケーション固有の値を指定するには、MaxRetry および RetryInterval 属性を使用する。 • INCOMING_ONLY: ドメイン ゲートウェイは、起動時にリモート ドメイン アクセス ポイントへの初期接続を行わず、リモート サービスは最初にサスペンドされる。ドメイン ゲートウェイはリモート ドメイン アクセス ポイントからの受信接続に使用可能で、リモート サービスはこのローカル ドメイン アクセス ポイントのドメイン ゲートウェイが受信接続を受け付けたときに通知される。接続の再試行処理は実行できない。 	<p>Administration Console のフィールド表示: [接続ポリシー]</p> <p>必須: いいえ</p> <p>デフォルト値: ON_DEMAND</p>

表 72-1 WTCLocalTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
Interoperate	ローカル ドメインが、Tuxedo リリース 6.5 に基づいたリモート ドメインと相互運用するかどうかを指定する。 Yes の場合、ドメインは Tuxedo 6.5 ドメインと相互運用する。	Administration Console の フィールド表示: [相互運用] 必須: いいえ デフォルト値: No
MaxEncryptBits	ローカル ドメインのネットワーク接続確立時に使用される最大の暗号化キー長 (ビット単位) を定義する。 <ul style="list-style-type: none"> • 値が 0 の場合、暗号化は使用されない。 • MaxEncryptBits 属性の値は、MinEncryptBits 属性の値以上でなければならない。 • MaxEncryptBits の値 40 は、Tuxedo 7.1 以降が動作するドメインでのみ使用できる。 	Administration Console の フィールド表示: [最大暗号化レベル] 必須: いいえ デフォルト値: 128 セキュアな値: "128"
MaxRetries	ドメイン ゲートウェイが、リモート ドメイン アクセス ポイントへの接続の確立を試行する回数を定義する。 ConnectionPolicy が「ON_STARTUP」に設定されている場合にのみ使用する。 <ul style="list-style-type: none"> • 再試行メカニズムを無効にするには、最小値を使用する。 • 接続が確立されるまで試行するには、最大値を使用する。 	Administration Console の フィールド表示: [最大再試行回数] 必須: いいえ デフォルト値: 2 ⁶³ -1 最小値: 0 最大値: 2 ⁶³ -1

表 72-1 WTCLocalTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
MinEncryptBits	<p>ローカル ドメインのネットワーク接続確立時に使用される最小の暗号化キー長 (ビット単位) を定義する。</p> <ul style="list-style-type: none"> • 値が 0 の場合、暗号化は使用されない。 • MinEncryptBits 属性の値は、MaxEncryptBits 属性の値以下でなければならない。 • MinEncryptBits の値 40 は、Tuxedo 7.1 以降が動作するドメインでのみ使用できる。 	<p>Administration Console のフィールド表示 : [最小暗号化レベル]</p> <p>必須 : いいえ デフォルト値 : 0 セキュアな値 : "40"</p>
NWAddr	<p>ローカル ドメイン ゲートウェイのネットワーク アドレスを定義する。以下のフォーマットのいずれかで TCP/IP アドレスを指定する。</p> <ul style="list-style-type: none"> • //hostname:port_number • //#. #. #. #:port_number <p>hostname を使用する場合、ドメインはローカル名解決機能 (通常は DNS) を使用して hostname のアドレスを検索する。ドットで区切った 10 進フォーマットを使用する場合、各 # は 0 ~ 255 の数字でなければならない。このドット区切りの 10 進数は、ローカルマシンの IP アドレスを表す。port_number は、ドメイン プロセスが受信するリクエストをリスンする TCP ポート番号。</p>	<p>Administration Console のフィールド表示 : [ネットワーク アドレス]</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>必須 : いいえ</p>

表 72-1 WTCLocalTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
RetryInterval	リモート ドメインアクセス ポイントへの接続を確立するために自動的に試行される接続の間隔の秒数を定義する。ConnectionPolicy が「ON_STARTUP」に設定されている場合のみ使用する。	Administration Console の フィールド表示: [再試行間隔] 必須: いいえ デフォルト値: 60 最小値: 0 最大値: 2 ³¹ -1
Security	セキュリティを実施するアプリケーションのタイプを定義する。 <ul style="list-style-type: none"> • NONE: セキュリティは使用されない。 • APP_PW: リモート ドメインからの接続が確立されたときに、パスワードセキュリティが強化される。アプリケーションパスワードは WTCResourcesMBean で定義される。 • DM_PW: リモート ドメインからの接続が確立されたときに、ドメインパスワードセキュリティが強化される。ドメインパスワードは WTCPasswordsMBean で定義される。 	Administration Console の フィールド表示: [セキュリティ] 必須: いいえ デフォルト値: NONE セキュアな値: "DM_PW"

WTCPassword

説明

このインターフェースは、WTC パスワード コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの WTC コンフィグレーションに適用されます。

構文

```
<WTCPassword
  LocalAccessPoint="String"
  LocalPassword="String"
  LocalPasswordIV="String"
  Name="String"
  Notes="String"
  RemoteAccessPoint="String"
  RemotePassword="String"
  RemotePasswordIV="String"
/>
```

親要素

- WTCServer

属性

表 73-1 WTCPassword の属性

属性	説明	指定できる値とデフォルト値
LocalAccessPoint	パスワードを適用するローカル アクセス ポイントの名前を定義する。	Administration Console のフィールド表示:[ローカル アクセス ポイント]
LocalPassword	genpasswd ユーティリティから返される LocalPassword を定義する。このパスワードは、 LocalAccessPoint によって識別されるローカル ドメイン アクセス ポイントと、 RemoteAccessPoint によって識別されるリモート ドメイン アクセス ポイント間の接続を認証するために使用する。	Administration Console のフィールド表示:[ローカル パスワード]
LocalPasswordIV	ローカル パスワードを暗号化するために使用する初期化ベクトルを定義する。	Administration Console のフィールド表示:[ローカル パスワード IV]
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須:いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須:いいえ

表 73-1 WTCPassword の属性 (続き)

属性	説明	指定できる値とデフォルト値
RemoteAccessPoint	パスワードを適用するリモート アクセス ポイントの名前を定義する。	Administration Console の フィールド表示 : [リモート ア クセス ポイント]
RemotePassword	genpasswd ユーティリティから返さ れる RemotePassword を定義する。こ のパスワードは、 LocalAccessPoint に よって識別されるローカル ドメイン アクセス ポイントと、 RemoteAccessPoint によって識別され るリモート ドメイン アクセス ポイン ト間の接続を認証するために使用す る。	Administration Console の フィールド表示 : [リモート パ スワード]
RemotePasswordIV	リモート パスワードを暗号化するた めに使用する初期化ベクトルを定義す る。	Administration Console の フィールド表示 : [リモート パ スワード IV]

WTCPassword

WTCRemoteTuxDom

説明

このインタフェースは、WTC リモート Tuxedo ドメイン コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの WTC コンフィグレーションに適用されます。

構文

```
<WTCRemoteTuxDom
  AccessPoint="String"
  AccessPointId="String"
  AclPolicy=( "GLOBAL" | "LOCAL" )
  CmpLimit="number"
  ConnPrincipalName="String"
  ConnectionPolicy=( "ON_DEMAND" | "ON_STARTUP" | "INCOMING_ONLY" | "LOCAL" )
  CredentialPolicy=( "GLOBAL" | "LOCAL" )
  FederationName="String"
  FederationURL="String"
  LocalAccessPoint="String"
  MaxEncryptBits=( "0" | "40" | "56" | "128" )
  MaxRetries="number"
  MinEncryptBits=( "0" | "40" | "56" | "128" )
  NWAddr="String"
```

```

Name="String"
Notes="String"
RetryInterval="number"
TpUsrFile="String"
/>

```

親要素

- WTCServer

属性

表 74-1 WTCRemoteTuxDom の属性

属性	説明	指定できる値とデフォルト値
AccessPoint	WTCServer MBean でドメインを識別するために使用する名前を定義する。この名前は WTCServer MBean における WTCLocalTuxDom および WTCRemoteTuxDom の AccessPoint 名のスコープ内でユニークでなければならない。	Administration Console のフィールド表示:[アクセス ポイント]
AccessPointId	別のドメインへの接続を確立するとき、ドメインを識別するための接続プリンシパル名を定義する。WTCRemoteTuxDom MBean の AccessPointId は、Tuxedo DMCONFIG ファイルの *DM_LOCAL_DOMAINS セクションにある対応する DOMAINID と一致する必要がある。	Administration Console のフィールド表示:[アクセス ポイント ID]

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
AcIPolicy	<p>リモート ドメインからの着信アクセス制御リスト (ACL) ポリシーを定義する。</p> <ul style="list-style-type: none"> • Interoperate を Yes に設定している場合、AcIPolicy は無視される。 • LOCAL : ローカル ドメインは、指定のリモート ドメインから受信したサービス リクエストの ID を、指定のリモート ドメインのローカル プリンシパル名で指定されたプリンシパル名に変更する。 • GLOBAL : ローカル ドメインは、ID を変更せずにサービス リクエストを渡す。 	<p>Administration Console のフィールド表示 : [ACL ポリシー]</p> <p>必須 : いいえ</p> <p>デフォルト値 : LOCAL</p> <p>セキュアな値 : "GLOBAL" または "LOCAL"</p>
CmpLimit	<p>リモート ドメインへのデータの送信に使用する圧縮しきい値を定義する。このサイズより大きいアプリケーションバッファは圧縮される。</p>	<p>Administration Console のフィールド表示 : [Cmp 制限]</p> <p>必須 : いいえ</p> <p>デフォルト値 : $2^{31}-1$</p> <p>最小値 : 0</p> <p>最大値 : $2^{31}-1$</p>

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnPrincipalName	<p>別のドメインへの接続を確立するとき、このドメインの識別子を検証するために使用されるプリンシパル名を指定する。</p> <ul style="list-style-type: none"> このパラメータは、BEA Tuxedo 7.1 以降のソフトウェアを実行しているタイプ TDOMAIN のドメインにのみ適用される。 指定しない場合、接続プリンシパル名はこのドメインの AccessPointID にデフォルト設定される。 <p>注意 : ConnectionPrincipalName はこのリリースではサポートされていない。</p>	<p>Administration Console のフィールド表示 : [接続プリンシパル名]</p> <p>必須 : いいえ</p>

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
ConnectionPolicy	<p>ローカル ドメインがリモート ドメインとの接続の確立を試行する場合の条件を定義する。</p> <ul style="list-style-type: none"> • ON_DEMAND: リモート サービスへのクライアント リクエストまたは管理接続コマンドのいずれかによってリクエストされたときのみ、接続が試行される。 • ON_STARTUP: ドメイン ゲートウェイは、ゲートウェイ サーバの初期化時にそのリモート ドメイン アクセス ポイントで接続を確立する。リモート サービス (このローカル アクセス ポイントのドメイン ゲートウェイによって JNDI に通知されるサービス) は、接続がそのリモート ドメイン アクセス ポイントに正常に確立されたときのみ通知される。リモート ドメイン アクセス ポイントへのアクティブな接続がない場合、リモート サービスはサスペンドされる。デフォルトでは、この接続ポリシーは失敗した接続を 60 秒ごとに再試行する。アプリケーション固有の値を指定するには、MaxRetry および RetryInterval 属性を使用する。 • INCOMING_ONLY: ドメイン ゲートウェイは、起動時にリモート ドメイン アクセス ポイントへの初期接続を行わず、リモート サービスは最初にサスペンドされる。ドメイン ゲートウェイはリモート ドメイン アクセス ポイントからの受信接続に使用可能で、リモート サービスはこのローカル ドメイン アクセス ポイントのドメイン ゲートウェイが受信接続を受け付けたときに通知される。接続の再試行処理は実行できない。 	<p>Administration Console のフィールド表示: [接続ポリシー]</p> <p>必須: いいえ</p> <p>デフォルト値: ON_DEMAND</p>

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
CredentialPolicy	<p>リモート ドメインへの発信アクセス制御リスト (ACL) ポリシーを定義する。</p> <ul style="list-style-type: none"> • Interoperate を Yes に設定している場合、CredentialPolicy は無視される。 • LOCAL : リモート ドメインは、ローカル ドメインから受信したサービス リクエストの ID を、このリモート ドメインのローカル プリンシパル名で指定されたプリンシパル名に設定する。 • GLOBAL : リモート ドメインは、変更せずにサービス リクエストを渡す。 	<p>Administration Console のフィールド表示 : [資格ポリシー]</p> <p>必須 : いいえ</p> <p>デフォルト値 : LOCAL</p>
FederationName	<p>外部ネーム サービスと結合するコンテキストを定義する。省略すると、結合ポイントは tuxedo.domains になる。</p>	<p>Administration Console のフィールド表示 : [フェデレーション名]</p> <p>必須 : いいえ</p>
FederationURL	<p>JNDI に結合される外部ネーム サービスの URL を定義する。省略した場合は、以下のように扱われる。</p> <ul style="list-style-type: none"> • WebLogic Tuxedo Connector は外部ドメインに CosNaming サーバがあると見なす。 • WebLogic Tuxedo Connector は TGIOP を使用して CosNaming サーバに結合する。 <p>WebLogic Tuxedo Connector は非 CORBA サービス プロバイダと結合することもできる。</p>	<p>Administration Console のフィールド表示 : [フェデレーション URL]</p> <p>必須 : いいえ</p>

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
LocalAccessPoint	リモート ドメインがアクセスするローカル ドメイン名を定義する。	Administration Console のフィールド表示: [ローカル アクセス ポイント]
MaxEncryptBits	<p>ローカル ドメインのネットワーク接続確立時に使用される最大の暗号化キートン長 (ビット単位) を定義する。</p> <ul style="list-style-type: none"> 値が 0 の場合、暗号化は使用されない。 MaxEncryptBits 属性の値は、MinEncryptBits 属性の値以上でなければならない。 MaxEncryptBits の値 40 は、Tuxedo 7.1 以降が動作するドメインでのみ使用できる。 	<p>Administration Console のフィールド表示: [最大暗号化レベル]</p> <p>必須: いいえ デフォルト値: 128</p>
MaxRetries	<p>ドメイン ゲートウェイが、リモート ドメイン アクセス ポイントへの接続の確立を試行する回数を定義する。ConnectionPolicy が「ON_STARTUP」に設定されている場合にのみ使用する。</p> <ul style="list-style-type: none"> WTCLocalTuxDomMBean の MaxRetries 属性で定義される値にデフォルト設定するには、値 -1 を使用する。 再試行メカニズムを無効にするには、0 を使用する。 接続が確立されるまで試行するには、最大値を使用する。 	<p>Administration Console のフィールド表示: [最大再試行回数]</p> <p>必須: いいえ デフォルト値: -1 最小値: -1 最大値: 2⁶³-1</p>

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
MinEncryptBits	<p>ローカル ドメインのネットワーク接続確立時に使用される最小の暗号化キー長 (ビット単位) を定義する。</p> <ul style="list-style-type: none"> • 値が 0 の場合、暗号化は使用されない。 • MinEncryptBits 属性の値は、MaxEncryptBits 属性の値以下でなければならない。 • MinEncryptBits の値 40 は、Tuxedo 7.1 以降が動作するドメインでのみ使用できる。 	<p>Administration Console のフィールド表示: [最小暗号化レベル]</p> <p>必須: いいえ デフォルト値: 0 セキュアな値: "40"</p>
NWAddr	<p>ローカル ドメイン ゲートウェイのネットワーク アドレスを定義する。以下のフォーマットのいずれかで TCP/IP アドレスを指定する。</p> <ul style="list-style-type: none"> • //hostname:port_number • //#. #. #. #:port_number <p>hostname を使用する場合、ドメインはローカル名解決機能 (通常は DNS) を使用して hostname のアドレスを検索する。ドットで区切った 10 進フォーマットを使用する場合、各 # は 0 ~ 255 の数字でなければならない。このドット区切りの 10 進数は、ローカルマシンの IP アドレスを表す。port_number は、ドメイン プロセスが受信するリクエストをリスンする TCP ポート番号。</p>	<p>Administration Console のフィールド表示: [ネットワーク アドレス]</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>必須: いいえ</p>

表 74-1 WTCRemoteTuxDom の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須: いいえ
RetryInterval	<p>リモート ドメイン アクセス ポイント への接続を確立するために自動的に試行される接続の間隔の秒数を定義する。</p> <ul style="list-style-type: none"> • ConnectionPolicy が「ON_STARTUP」に設定されている場合にのみ使用する。 • WTCLocalTuxDomMBean の RetryInterval 属性で定義される値にデフォルト設定するには、値 -1 を使用する。 	<p>Administration Console のフィールド表示: [再試行間隔]</p> <p>必須: いいえ</p> <p>デフォルト値: -1</p> <p>最小値: -1</p> <p>最大値: 2³¹-1</p>
TpUsrFile	<p>uid/gid 情報を含むユーザ パスワード ファイルへの絶対パスを定義する。このファイルは、リモート ドメインで Tuxedo tpusradd ユーティリティによって生成されるファイルと同じ。ユーザ名、uid および gid 情報が含まれていて、適切な認可、認証、監査に対して有効でなければならない。</p>	<p>Administration Console のフィールド表示: [TP ユーザファイル]</p> <p>必須: いいえ</p>

WTCResources

説明

このインターフェースは、WTC リソース コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの WTC コンフィグレーションに適用されます。

構文

```
<WTCResources
  AppPassword="String"
  AppPasswordIV="String"
  FldTbl16Classes="list of Strings"
  FldTbl32Classes="list of Strings"
  Name="String"
  Notes="String"
  ViewTbl16Classes="list of Strings"
  ViewTbl32Classes="list of Strings"
/>
```

親要素

- WTCServer

属性

表 75-1 WTCResources の属性

属性	説明	指定できる値とデフォルト値
AppPassword	genpasswd ユーティリティから返されるアプリケーションパスワードを定義する。この Tuxedo アプリケーションパスワードは、接続の認証に使用する暗号化されたパスワード。	Administration Console のフィールド表示:[アプリケーションパスワード] 必須: いいえ
AppPasswordIV	AppPassword を暗号化するために使用する初期化ベクトルを定義する。genpasswd ユーティリティから、AppPassword と共に返される。	Administration Console のフィールド表示:[アプリケーションパスワード IV] 必須: いいえ
FldTbl16Classes	クラス ローダによってロードされ、FldTbl 配列に追加される FldTbl16Classes の名前を定義する。 <ul style="list-style-type: none"> 必要なクラスの完全修飾名を使用する。 複数のクラスを入力する場合は、カンマ区切りのリストを使用する。 	Administration Console のフィールド表示:[FldTbl クラス] 必須: いいえ

表 75-1 WTCResources の属性 (続き)

属性	説明	指定できる値とデフォルト値
FldTbl32Classes	<p>クラス ローダによってロードされ、FldTbl 配列に追加される FldTbl32Classes の名前を定義する。</p> <ul style="list-style-type: none"> 必要なクラスの完全修飾名を使用する。 複数のクラスを入力する場合は、カンマ区切りのリストを使用する。 	<p>Administration Console のフィールド表示 : [FldTbl32 クラス]</p> <p>必須 : いいえ</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示 : [名前]</p> <p>必須 : いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>必須 : いいえ</p>
ViewTbl16Classes	<p>クラス ローダによってロードされ、ViewTbl 配列に追加される ViewTbl16Classes の名前を定義する。</p> <ul style="list-style-type: none"> 必要なクラスの完全修飾名を使用する。 複数のクラスを入力する場合は、カンマ区切りのリストを使用する。 	<p>Administration Console のフィールド表示 : [ViewTbl クラス]</p> <p>必須 : いいえ</p>
ViewTbl32Classes	<p>クラス ローダによってロードされ、ViewTbl 配列に追加される ViewTbl32Classes の名前を定義する。</p> <ul style="list-style-type: none"> 必要なクラスの完全修飾名を使用する。 複数のクラスを入力する場合は、カンマ区切りのリストを使用する。 	<p>Administration Console のフィールド表示 : [ViewTbl32 クラス]</p> <p>必須 : いいえ</p>

WTCServer

説明

この MBean は、WTC サーバを定義します。

構文

```
<WTCServer  
  DeploymentOrder=number  
  Name=String  
  Notes=String  
  Targets=list of Target names  
>
```

親要素

- Domain

子要素

- WTCLocalTuxDom
- WTCRemoteTuxDom
- WTCEXport
- WTCImport
- WTCPassword
- WTCResources
- WTCtBridgeGlobal
- WTCtBridgeRedirect

属性

表 76-1 WTCServer の属性

属性	説明	指定できる値とデフォルト値
DeploymentOrder	<p>デプロイの優先順位。サーバではこれを使って項目のデプロイ順を決定する。優先順位は、同じタイプの他のデプロイ可能な項目に対する相対的な順位。たとえば、サーバはすべての EJB を優先順位に従ってデプロイしてから、起動クラスを優先順位に従ってデプロイする。</p> <p>デプロイ順の値が小さい項目から順にデプロイされる。デプロイ順の値が等しい項目のデプロイメント順序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。</p>	<p>Administration Console のフィールド表示: [デプロイ順]</p> <p>必須: いいえ</p> <p>デフォルト値: 1000</p> <p>最小値: 0</p> <p>最大値: 2³¹-1</p>
Name	<p>このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。</p>	<p>Administration Console のフィールド表示: [名前]</p> <p>必須: いいえ</p>
Notes	<p>このコンフィグレーションの説明として任意に入力できる情報。</p>	<p>Administration Console のフィールド表示: [メモ]</p> <p>必須: いいえ</p>
Targets	<p>現在のドメイン内で、この項目のデプロイ先となる対象。</p>	<p>必須: いいえ</p>

WTCtBridgeGlobal

説明

このインターフェースは、WTC tBridge グローバル コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの tBridge コンフィグレーションに適用されます。

構文

```
<WTCtBridgeGlobal
  AllowNonStandardTypes=String
  DefaultReplyDeliveryMode=( "PERSIST" | "NONPERSIST" | "DEFAULT" )
  DeliveryModeOverride=( "PERSIST" | "NONPERSIST" )
  JmsFactory=String
  JmsToTuxPriorityMap=String
  JndiFactory=String
  Name=String
  Notes=String
  Retries=number
  RetryDelay=number
  Timeout=number
  Transactional=String
  TuxErrorQueue=String
  TuxFactory=String
```

```
TuxToJmsPriorityMap="String"  
UserId="String"  
WlsErrorDestination="String"  
</>
```

親要素

- WTCServer

属性

表 77-1 WTCtBridgeGlobal の属性

属性	説明	指定できる値とデフォルト値
AllowNonStandardTypes	<p>非標準データ型が tBridge の通過を許可されているかどうかを指定するためのフラグを定義する。標準型は、ASCII テキスト (TextMessage、STRING) または BLOB (ByteMessage、CARRY)。</p> <ul style="list-style-type: none"> • NO : 非標準型は拒否され、指定したエラー位置に配置される。 • YES : 非標準型は、元の型を示すタグと共に、BLOB としてターゲット位置に配置される。 	<p>Administration Console のフィールド表示 : [非標準型を許可]</p> <p>必須 : いいえ</p> <p>デフォルト値 : NO</p>
DefaultReplyDeliveryMode	<p>ターゲット位置にメッセージを配置するときに、メッセージに関連付ける応答配信モードを定義する。</p> <ul style="list-style-type: none"> • メッセージが JMS から Tuxedo/Q にリダイレクトされるときに、JMS_BEA_TuxGtway_Tuxedo_ReplyDeliveryMode プロパティがメッセージに設定されていない場合に使用する。 • defaultReplyDeliveryMode および JMS_BEA_TuxGtway_Tuxedo_ReplyDeliveryMode が設定されていない場合、Tuxedo/Q サブシステムは Tuxedo に定義されるデフォルトセマンティクスを使用する。 	<p>Administration Console のフィールド表示 : [デフォルト応答配信モード]</p> <p>必須 : いいえ</p> <p>セキュアな値 : 60</p>

表 77-1 WTCtBridgeGlobal の属性 (続き)

属性	説明	指定できる値とデフォルト値
DeliveryModeOverride	<p>ターゲット位置にメッセージを配置するときに使用される配信モードを定義する。</p> <ul style="list-style-type: none"> メッセージと関連付けられている配信モードをオーバーライドする。 指定しない場合、メッセージはソース位置から指定された配信モードと同じモードを使用して、ターゲット位置に配置される。 	<p>Administration Console のフィールド表示 : [配信モードオーバーライド]</p> <p>必須 : いいえ</p>
JmsFactory	<p>JMS 接続ファクトリの名前を定義する。例： weblogic.jms.ConnectionFactory</p>	<p>Administration Console のフィールド表示 : [JMS ファクトリ]</p> <p>セキュアな値 : 60</p>
JmsToTuxPriorityMap	<p>JMS から Tuxedo への優先度のマッピングを定義する。10 の JMS 優先度 (0=>9) を指定できるが、これらは 100 の Tuxedo 優先度 (1=>100) と組み合わせることができる。マッピングは、値と範囲のペアを「 」で区切ったリスト (jmsvalue:tuxrange) で構成される。ペアは「:」で区切られ、範囲は「-」で区切られる。</p> <p>例：</p> <p>0:1 1:12 2:23 3:34 4:45 5:56 6:67 7:78 8:89 9:100</p> <p>または</p> <p>0:1-10 1:11-20 2:21-30 3:31-40 4:41-50 5:51-60 6:61-70 7:71-80 8:81-90 9:91-100</p>	<p>Administration Console のフィールド表示 : [Jms から Tux の優先度マップ]</p> <p>必須 : いいえ</p>

表 77-1 WTCtBridgeGlobal の属性 (続き)

属性	説明	指定できる値とデフォルト値
JndiFactory	JNDI ルックアップ ファクトリ の名前を定義する。例： weblogic.jndi.WLInitialContextFactory	Administration Console の フィールド表示：[JNDI ファクトリ]
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示：[名前] 必須：いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須：いいえ
Retries	指定したエラー位置にメッセージを配置し、エラーをロギングするまでに、メッセージのリダイレクトを試行する回数を定義する。	Administration Console の フィールド表示：[再試行回数] 必須：いいえ デフォルト値：0 最小値：0 最大値： $2^{31}-1$
RetryDelay	エラー発生後にメッセージをリダイレクトするまでに待機する最小時間（ミリ秒単位）を定義する。この期間中、このスレッドから他のメッセージはリダイレクトされない。他のスレッドはメッセージのリダイレクトを継続できる。	Administration Console の フィールド表示：[再試行の遅延] 必須：いいえ デフォルト値：10 最小値：0 最大値： $2^{31}-1$

表 77-1 WTCtBridgeGlobal の属性 (続き)

属性	説明	指定できる値とデフォルト値
Timeout	<p>ターゲット位置にメッセージを配置する場合の、リダイレクト全体に対して効果的なタイムアウトの長さ (秒単位) を定義する。0 は無限に待機することを示す。</p>	<p>Administration Console のフィールド表示 : [タイムアウト]</p> <p>必須 : いいえ</p> <p>デフォルト値 : 60</p> <p>セキュアな値 : 60</p> <p>最小値 : 0</p> <p>最大値 : 2³¹-1</p>
Transactional	<p>ソース位置からメッセージを検索するとき、およびターゲット位置へメッセージを配置するときの、トランザクションの使用を指定するフラグを定義する。</p> <ul style="list-style-type: none"> • YES の場合、両方の操作でトランザクションが使用される。 • NO の場合、いずれの操作でもトランザクションは使用されない。 <p>注意 : Transactional はこのリリースではサポートされていない。</p>	<p>Administration Console のフィールド表示 : [トランザクション]</p> <p>必須 : いいえ</p> <p>デフォルト値 : YES</p>
TuxErrorQueue	<p>Tuxedo/Q ソース キューにリダイレクトできないメッセージを格納するために使用する Tuxedo キューの名前を定義する。このキューは、ソース キューと同じキュー スペースにある。</p> <ul style="list-style-type: none"> • 指定しない場合、リダイレクトされないすべてのメッセージは失われる。 • TuxErrorQueue にメッセージを配置できない場合、エラー ログが記録され、メッセージは失われる。 	<p>Administration Console のフィールド表示 : [Tuxedo エラー キュー]</p> <p>必須 : いいえ</p>

表 77-1 WTCtBridgeGlobal の属性 (続き)

属性	説明	指定できる値とデフォルト値
TuxFactory	Tuxedo 接続ファクトリの名前を定義する。例： tuxedo.services.TuxedoConnection	Administration Console の フィールド表示：[Tuxedo ファクトリ]
TuxToJmsPriorityMap	Tuxedo から JMS への優先度のマッピングを定義する。100 の Tuxedo 優先度 (1=>100) を指定できるが、これらは 10 の JMS 優先度 (0=>9) と組み合わせることができる。マッピングは、値と範囲のペアを「 」で区切ったリスト (tuxvalue:jmsrange) で構成される。ペアは「:」で区切られ、範囲は「-」で区切られる。 例： 1:0 12:1 23:2 34:3 45:4 56:5 67:6 78:7 89:8 100:9 または 20:0-1 40:2-3 60:4-5 80:6-7 100:8-9	Administration Console の フィールド表示：[Tux から Jms の優先度マップ] 必須：いいえ

表 77-1 WTCtBridgeGlobal の属性 (続き)

属性	説明	指定できる値とデフォルト値
Userld	<p>セキュリティがコンフィグレーションされる場合、ACL チェックのために tBridge によって処理されるすべてのメッセージのユーザ ID を定義する。</p> <ul style="list-style-type: none"> • セキュリティ / 認証コンテキストがサブシステム間で渡されるまで、すべてのメッセージはこの ID を想定する。セキュリティ コンテキストが渡されるまでは、ソース位置から受信したメッセージの生成者を識別する安全な方法はない。 • 引数のユーザは、ユーザ名またはユーザ ID 番号 (uid) のいずれかで指定できる。 	<p>Administration Console のフィールド表示 : [ユーザ ID]</p> <p>必須 : いいえ</p>
WlsErrorDestination	<p>メッセージをリダイレクトできない場合、WebLogic Server JMS メッセージを格納するために使用される場所の名前を定義する。</p> <ul style="list-style-type: none"> • 指定しない場合、リダイレクトされないすべてのメッセージは失われる。 • 何らかの理由により WlsErrorDestination にメッセージを配置できない場合、エラーログが記録され、メッセージは失われる。 	<p>Administration Console のフィールド表示 : [WLS エラー送り先]</p> <p>必須 : いいえ</p>

WTCtBridgeRedirect

説明

このインターフェースは、WTC tBridge リダイレクト コンフィグレーション属性へのアクセスを提供します。ここで定義されるメソッドは、WLS ドメイン レベルの tBridge コンフィグレーションに適用されます。

構文

```
<WTCtBridgeRedirect
  Direction=( "JmsQ2TuxQ" | "TuxQ2JmsQ" | "JmsQ2TuxS" | "JmsQ2JmsQ" )
  MetadataFile="String"
  Name="String"
  Notes="String"
  ReplyQ="String"
  SourceAccessPoint="String"
  SourceName="String"
  SourceQspace="String"
  TargetAccessPoint="String"
  TargetName="String"
  TargetQspace="String"
  TranslateFML=( "NO" | "FLAT" | "WLXT" )
/>
```

親要素

- WTCServer

属性

表 78-1 WTCtBridgeRedirect の属性

属性	説明	指定できる値とデフォルト値
Direction	<p>データ フローの方向を定義する。定義された各方向は新しいスレッドを開始することにより処理される。tBridge コンフィグレーションで少なくとも1つのリダイレクトを指定しないと、tBridge は起動に失敗し、エラーがログに記録される。</p> <p>リダイレクトのキーワードは以下のとおり。</p> <ul style="list-style-type: none"> • JmsQ2TuxQ - JMS から TUXEDO /Q へ • TuxQ2JmsQ - TUXEDO /Q から JMS へ • JmsQ2TuxS - JMS から JMS への TUXEDO サービス応答へ • JmsQ2JmsQ - JMS から JMS へ 	Administration Console のフィールド表示:[ソート順]
MetaDataFile	<p>WLXT の呼び出しに渡すための metadataFile URL の名前を定義する。</p> <p>注意: このリリースではサポートされていない。</p>	Administration Console のフィールド表示:[メタ データ ファイル] 必須: いいえ

表 78-1 WTCtBridgeRedirect の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console の フィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ
ReplyQ	特に Tuxedo サービスへの同期呼び出しを行う JMS キューの名前を定義する。応答は JMS ReplyQ に返される。	Administration Console の フィールド表示 : [応答キュー] 必須 : いいえ
SourceAccessPoint	ソースが存在するローカルまたはリモートアクセスポイントの名前を定義する。	Administration Console の フィールド表示 : [ソース アクセス ポイント] 必須 : いいえ
SourceName	ソース キューまたはサービスの名前を定義する。JMS キュー名、Tuxedo キュー名、または Tuxedo サービス名を指定する。	Administration Console の フィールド表示 : [ソース名]
SourceQspace	ソース位置の Qspace の名前を定義する。	Administration Console の フィールド表示 : [ソース Qspace] 必須 : いいえ
TargetAccessPoint	ターゲットが存在するローカルまたはリモートアクセスポイントの名前を定義する。	Administration Console の フィールド表示 : [ターゲット アクセス ポイント] 必須 : いいえ
TargetName	ターゲットのキューまたはソースを定義する。JMS キュー名、Tuxedo キュー名、または Tuxedo サービス名を指定する。	Administration Console の フィールド表示 : [ターゲット名]

表 78-1 WTCtBridgeRedirect の属性 (続き)

属性	説明	指定できる値とデフォルト値
TargetQspace	ターゲット位置の Qspace の名前を定義する。	Administration Console の フィールド表示 : [ターゲット Qspace] 必須 : いいえ
TranslateFML	XML/FML 変換のタイプを定義する。 <ul style="list-style-type: none"> • NO : データ変換は実行されない。転送の方向に応じて、TextMessage は STRING に (またはその逆に) マップされる。BytesMessage は、CARRAY に (またはその逆に) マップされる。他のすべてのデータ型の場合、リダイレクトは失敗する。 • FLAT : メッセージペイロードは WebLogic Tuxedo Connector のトランスレータを使用して変換される。 • WLXT : 変換は XML から 非 XML への WebLogic XML トランスレータ (WLXT) によって実行される。 注意 : WLXT はこのリリースではサポートされていない。	Administration Console の フィールド表示 : [FML 変換] 必須 : いいえ デフォルト値 : NO

XMLEntityCache

説明

サーバ上の JAXP (Java API for XML Parsing) の動作をコンフィグレーションします。

構文

```
<XMLEntityCache
  CacheDiskSize="number"
  CacheLocation="String"
  CacheMemorySize="number"
  CacheTimeoutInterval="number"
  MaxSize="number"
  Name="String"
  Notes="String"
/>
```

親要素

- Domain

属性

表 79-1 XMLEntityCache の属性

属性	説明	指定できる値とデフォルト値
CacheDiskSize	キャッシュのディスク サイズ (単位は MB) を返す。	Administration Console の フィールド表示: [キャッシュ ディスク サイズ] 必須: いいえ デフォルト値: 5 最小値: 0
CacheLocation	永続キャッシュ ファイルのパス名を返す。	必須: いいえ デフォルト値: xmlcache
CacheMemorySize	キャッシュのメモリ サイズ (単位は KB) を返す。	Administration Console の フィールド表示: [キャッシュ メモリ サイズ] 必須: いいえ デフォルト値: 500 最小値: 0
CacheTimeoutInterval	キャッシュのデフォルト タイムアウト 間隔 (単位は秒) を返す。	Administration Console の フィールド表示: [キャッシュ タイムアウト 間隔] 必須: いいえ デフォルト値: 120 最小値: 0
MaxSize		必須: いいえ デフォルト値: 0

表 79-1 XMLEntityCache の属性 (続き)

属性	説明	指定できる値とデフォルト値
Name	このコンフィグレーションの名前。 WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	必須 : いいえ

XMLEntitySpecRegistryEntry

説明

XML レジストリのエン트리です。XML レジストリ エントリは、特定の XML ドキュメントタイプに関連付けられたコンフィグレーション情報です。このインタフェースでアクセスされるエントリは、リモート エンティティ参照のローカルインスタンスを指定するために使用されます。このタイプのレジストリ エントリでは、パブリック ID (「-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN」など)、またはシステム ID (「http://java.sun.com/j2ee/dtds/ejb-jar_2_0.dtd」など) のいずれか、または両方でドキュメントタイプが識別されます。このコンフィグレーション情報は、SAX EntityResolver を設定するために WebLogic JAXP 実装で使用されます。

構文

```
<XMLEntitySpecRegistryEntry
  CacheTimeoutInterval="number"
  EntityURI="String"
  Name="String"
  Notes="String"
  PublicId="String"
  SystemId="String"
  WhenToCache=( "cache-on-reference" | "cache-at-initialization" | "cache-never"
```

```
| "defer-to-registry-setting" )
/>
```

親要素

- XMLRegistry

属性

表 80-1 XMLEntitySpecRegistryEntry の属性

属性	説明	指定できる値とデフォルト値
CacheTimeoutInterval	キャッシュのデフォルト タイムアウト間隔 (単位は秒) を返す。-1 の値を指定した場合、この値はキャッシュ MBean から委託される。	Administration Console のフィールド表示: [キャッシュ タイムアウト間隔] 必須: いいえ デフォルト値: -1 最小値: -1
EntityURI	このレジストリ エントリと関連付けられた外部エンティティ (例: DTD) のローカル コピーの場所を返す。この場所は、インストール先の XML レジストリ ディレクトリの 1 つを基準とした相対パス名か、またはあるローカル リポジトリ (例: dbms) におけるエンティティの場所の URI。	Administration Console のフィールド表示: [エンティティ URI] 必須: いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ

表 80-1 XMLEntitySpecRegistryEntry の属性 (続き)

属性	説明	指定できる値とデフォルト値
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console の フィールド表示 : [メモ] 必須 : いいえ
PublicId	このレジストリ エントリで表されるドキュメント タイプのパブリック ID を取得する。	Administration Console の フィールド表示 : [パブリック ID] 必須 : いいえ
SystemId	このレジストリ エントリで表されるドキュメント タイプのシステム ID を取得する。	Administration Console の フィールド表示 : [システム ID] 必須 : いいえ
WhenToCache	この項目を直ちにキャッシュするか、参照されるまで待機するかを設定する。	Administration Console の フィールド表示 : [キャッシュのタイミング] 必須 : いいえ デフォルト値 : defer-to-registry-setting

XMLParserSelectRegistryEntry

説明

XML レジストリのエン트리です。XML レジストリ エントリは、特定の XML ドキュメント タイプに関連付けられたコンフィグレーション情報です。このインタフェースでアクセスされるエントリは、提供されたドキュメント タイプの SAX パーサまたはドキュメントビルダを指定するために使用されます。ドキュメントタイプは、パブリック ID (「-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans 2.0//EN」など)、システム ID (「http://java.sun.com/j2ee/dtds/ejb-jar_2_0.dtd」など)、またはドキュメントルートタグ名 (「ejb-jar」など) の 1 つまたは複数によって識別されます。このコンフィグレーション情報は、WebLogic JAXP の実装で適切なパーサ ファクトリ (SAX および DOM) を選択するために使用されます。

構文

```
<XMLParserSelectRegistryEntry
  DocumentBuilderFactory="String"
  Name="String"
  Notes="String"
  ParserClassName="String"
  PublicId="String"
  RootElementTag="String"
```

XMLParserSelectRegistryEntry

```
SAXParserFactory="String"  
SystemId="String"  
TransformerFactory="String"  
</>
```

親要素

- XMLRegistry

属性

表 81-1 XMLParserSelectRegistryEntry の属性

属性	説明	指定できる値とデフォルト値
DocumentBuilderFactory	レジストリ エントリ と関連付けられた DocumentBuilderFactory のクラス名を返す。	Administration Console のフィールド表示 : [Document Builder ファクトリ] 必須 : いいえ
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示 : [名前] 必須 : いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示 : [メモ] 必須 : いいえ
ParserClassName	非推奨。 レジストリ エントリに関連付けられているカスタム XML パーサのクラス名を返す。	Administration Console のフィールド表示 : [パーサ クラス名] 必須 : いいえ

表 81-1 XMLParserSelectRegistryEntry の属性 (続き)

属性	説明	指定できる値とデフォルト値
PublicId	このレジストリ エントリで表されるドキュメント タイプのパブリック ID を取得する。	Administration Console のフィールド表示 : [パブリック ID] 必須 : いいえ
RootElementTag	このレジストリ エントリによって表されるドキュメント タイプのドキュメント ルート要素のタグ名を取得する。	Administration Console のフィールド表示 : [ルート要素タグ] 必須 : いいえ
SAXParserFactory	レジストリ エントリと関連付けられた SAXParserFactory のクラス名を返す。	Administration Console のフィールド表示 : [SAXParserファクトリ] 必須 : いいえ
SystemId	このレジストリ エントリで表されるドキュメント タイプのシステム ID を取得する。	Administration Console のフィールド表示 : [システム ID] 必須 : いいえ
TransformerFactory	デフォルトの TransformerFactory のクラス名を返す。	必須 : いいえ

XMLRegistry

説明

サーバ上の JAXP (Java API for XML Parsing) の動作をコンフィグレーションします。

構文

```
<XMLRegistry
  DocumentBuilderFactory=String
  Name=String
  Notes=String
  SAXParserFactory=String
  TransformerFactory=String
  WhenToCache=( "cache-on-reference" | "cache-at-initialization" | "cache-never"
)
/>
```

親要素

- Domain

子要素

- XMLEntitySpecRegistryEntry
- XMLParserSelectRegistryEntry

属性

表 82-1 XMLRegistry の属性

属性	説明	指定できる値とデフォルト値
DocumentBuilderFactory	デフォルトの DocumentBuilderFactory のクラス名を返す。	Administration Console のフィールド表示: [Document Builder ファクトリ] 必須: いいえ デフォルト値: weblogic.apache.xerces.jaxp.DocumentBuilderFactoryImpl
Name	このコンフィグレーションの名前。WebLogic Server では MBean を使用してコンフィグレーションを実装および保持する。	Administration Console のフィールド表示: [名前] 必須: いいえ
Notes	このコンフィグレーションの説明として任意に入力できる情報。	Administration Console のフィールド表示: [メモ] 必須: いいえ

表 82-1 XMLRegistry の属性 (続き)

属性	説明	指定できる値とデフォルト値
SAXParserFactory	デフォルトの SAXParserFactory のクラス名を返す。	Administration Console の フィールド表示: [SAX パーサ ファクトリ] 必須: いいえ デフォルト値: weblogic.apache.xerces.jaxp.SA XParserFactoryImpl
TransformerFactory	デフォルトの TransformerFactory のクラス名を返す。	Administration Console の フィールド表示: [Transformer ファクトリ] 必須: いいえ デフォルト値: weblogic.apache.xalan.processor .TransformerFactoryImpl
WhenToCache	直ちに項目をキャッシュするか、参照されるまで待機するかを設定する。	Administration Console の フィールド表示: [キャッシュ のタイミング] 必須: いいえ デフォルト値: cache-on-reference

索引

A

AbandonTimeoutSeconds 属性 37-3
AcceptBacklog 属性 46-3, 52-5
AcceptContextPathInGetRealPath 属性 65-3, 67-3
AccessPoint 属性 72-2, 74-2
AccessPointId 属性 72-2, 74-2
AcknowledgeMode 属性 32-2
AcknowledgePolicy 属性 24-3
ACLName 属性 17-2, 20-3
AclPolicy 属性 74-3
AdapterJNDIName 属性 3-3, 23-2
Addresses 属性 41-2, 63-2
AdministrationMBeanAuditingEnabled 属性 8-4
AdministrationPort 属性 8-4, 52-5
AdministrationPortEnabled 属性 8-5
AlgorithmType 属性 20-3
AllowCloseInOnMessage 属性 24-4
AllowNonStandardTypes 属性 77-3
ApartmentThreaded 属性 5-2
Application 要素
 DeploymentTimeout 属性 2-3
 DeploymentType 属性 2-3
 LoadOrder 属性 2-3
 Name 属性 2-3
 Notes 属性 2-3
 Path 属性 2-4
 StagingMode 属性 2-5
ApplicationPassword 属性 36-2, 69-3
AppPassword 属性 75-2
AppPasswordIV 属性 75-2
Arguments 属性 53-2, 54-2, 62-2
AsyncEnabled 属性 43-3
AttributeMBeanName 属性 56-2

AttributeMBeanType 属性 56-2
AttributeName 属性 56-2
AuditProviderClassName 属性 50-2
AuthCookieEnabled 属性 65-3, 67-3
AuthFilter 属性 66-2, 68-2
AuthProgram 属性 64-2
AuthRealmName 属性 66-2, 68-2
AutoKillIfFailed 属性 52-6
AutoRestart 属性 52-6

B

BackupCopies 属性 12-2
BackupHour 属性 12-2
BackupMinute 属性 12-3
BatchInterval 属性 43-4
BatchSize 属性 43-4
BeaHome 属性 53-2
BeforeCompletionIterationLimit 属性 37-3
BlockTime 属性 72-3
BridgeDestination 要素
 AdapterJNDIName 属性 3-3
 Classpath 属性 3-3
 Name 属性 3-3
 Notes 属性 3-3
 Properties 属性 3-4
 UserName 属性 3-5
 UserPassword 属性 3-5
BytesMaximum 属性 30-3, 31-3, 34-3, 35-3
BytesPagingEnabled 属性 30-4, 31-4, 34-4, 35-4
BytesThresholdHigh 属性 30-5, 31-5, 34-5, 35-5
BytesThresholdLow 属性 30-6, 31-6, 34-6, 35-6

C

- CacheDiskSize 属性 79-2
- CacheEnabled 属性 12-3
- CacheLocation 属性 79-2
- CacheMemorySize 属性 79-2
- CacheSize 属性 12-3
- CacheTimeoutInterval 属性 79-2, 80-2
- CacheTTL 属性 12-3
- CachingRealm 属性 48-2
- CapacityIncrement 属性 17-2
- CertAuthenticator 属性 61-3
- Certificate 属性 47-2
- CertificateAuthenticationEnabled 属性 69-3
- CertificateCacheSize 属性 61-3
- CertificatePassword 属性 47-2
- CertificateType 属性 47-2
- ChannelWeight 属性 46-3
- Charsets 属性 65-3, 67-3
- CheckpointIntervalSeconds 属性 37-4
- ChunkedTransferDisabled 属性 65-3, 67-3
- Ciphersuites 属性 61-4
- ClassName 属性 54-2, 62-2
- ClassPath 属性 53-2
- Classpath 属性 3-3, 23-2
- ClasspathServletDisabled 属性 52-6
- ClientCertificateEnforced 属性 61-4
- ClientCertProxyEnabled 属性 4-3, 52-7
- ClientId 属性 24-4
- Cluster 属性 52-8
- Cluster 要素
 - ClientCertProxyEnabled 属性 4-3
 - ClusterAddress 属性 4-4
 - DefaultLoadAlgorithm 属性 4-4
 - HttpTraceSupportEnabled 属性 4-4
 - IdlePeriodsUntilTimeout 属性 4-5
 - MulticastAddress 属性 4-5
 - MulticastBufferSize 属性 4-5
 - MulticastPort 属性 4-5
 - MulticastSendDelay 属性 4-5
 - MulticastTTL 属性 4-6
 - Name 属性 4-6
 - Notes 属性 4-6
 - WeblogicPluginEnabled 属性 4-6
- ClusterAddress 属性 4-4, 46-3
- ClusteringEnabled 属性 65-3, 67-3
- ClusterWeight 属性 52-8
- CmpLimit 属性 72-3, 74-3
- COM 要素
 - ApartmentThreaded 属性 5-2
 - MemoryLoggingEnabled 属性 5-2
 - Name 属性 5-3
 - NativeModeEnabled 属性 5-3
 - Notes 属性 5-3
 - NTAuthHost 属性 5-2
 - PrefetchEnums 属性 5-3
 - VerboseLoggingEnabled 属性 5-3
- COMEnabled 属性 46-3, 52-6
- Community 属性 58-2, 59-2
- CommunityPrefix 属性 55-2
- CompatibilityMode 属性 50-2
- CompleteCOMMessageTimeout 属性 46-4, 52-8
- CompleteHTTPMessageTimeout 属性 46-4, 52-9
- CompleteIOPMessageTimeout 属性 52-9
- CompleteMessageTimeout 属性 16-2
- CompleteT3MessageTimeout 属性 46-4, 52-10
- ConfigurationData 属性 7-2
- ConfigurationVersion 属性 8-5
- ConnectionFactory 属性 32-2
- ConnectionFactoryJNDIName 属性 23-2
- ConnectionFactory 属性 50-2
- ConnectionFactoryRules 属性 50-2
- ConnectionLoggerEnabled 属性 50-3
- ConnectionPolicy 属性 72-4, 74-5
- ConnectionPool 属性 29-2
- ConnectionPoolFailoverCallbackHandler 属性 20-4
- ConnectionURL 属性 23-3
- ConnectionWaitPeriod 属性 18-3
- ConnectorComponent 要素
 - DeploymentOrder 属性 6-2
 - Name 属性 6-2
 - Notes 属性 6-2
 - Targets 属性 6-2
 - URI 属性 6-2

ConnLeakProfilingEnabled 属性 17-3
ConnPrincipalName 属性 72-3, 74-4
ConsoleContextPath 属性 8-5
ConsoleEnabled 属性 8-5
CountOfRefreshFailuresTillDisable 属性 17-3
CountOfTestFailuresTillFlush 属性 17-3
Credential 属性 12-3
CredentialPolicy 属性 74-6
CustomRealm 要素
 ConfigurationData 属性 7-2
 Name 属性 7-2
 Notes 属性 7-2
 Password 属性 7-2
 RealmClassName 属性 7-2

D

DebugLevel 属性 55-2
DefaultDeliveryMode 属性 24-5
DefaultIOPPassword 属性 52-10
DefaultIOPUser 属性 52-10
DefaultInternalServletsDisabled 属性 52-11
DefaultLoadAlgorithm 属性 4-4
DefaultPriority 属性 24-5
DefaultProtocol 属性 52-11
DefaultRedeliveryDelay 属性 24-5
DefaultReplyDeliveryMode 属性 77-3
DefaultSecureProtocol 属性 52-11
DefaultServerName 属性 65-4, 67-3
DefaultTGIOPPassword 属性 52-11
DefaultTGIOPUser 属性 52-11
DefaultTimeToDeliver 属性 24-6
DefaultTimeToLive 属性 24-6
DefaultWebApp 属性 65-4, 67-4
DeliveryModeOverride 属性 30-6, 34-6, 35-6,
 77-4
DeploymentOrder 属性 6-2, 10-2, 15-2, 17-4,
 18-3, 20-4, 21-2, 22-3, 24-6, 26-2, 27-2,
 31-7, 36-2, 42-2, 43-4, 44-2, 46-5, 49-2,
 54-2, 62-2, 65-4, 66-3, 67-4, 68-3, 69-4,
 76-3
DeploymentTimeout 属性 2-3
DeploymentType 属性 2-3

Description 属性 46-5
DestinationJNDIName 属性 23-3
DestinationKeys 属性 30-7, 34-7, 35-7
DestinationType 属性 23-3
Direction 属性 25-2, 78-2
Directory 属性 28-2
DocumentBuilderFactory 属性 81-2, 82-2
Domain 要素
 AdministrationMBeanAuditingEnabled 属
 性 8-4
 AdministrationPort 属性 8-4
 AdministrationPortEnabled 属性 8-5
 ConfigurationVersion 属性 8-5
 ConsoleContextPath 属性 8-5
 ConsoleEnabled 属性 8-5
 Name 属性 8-6
 Notes 属性 8-6
DomainLogFilter 属性 52-12
DomainLogFilter 要素
 Name 属性 9-2
 Notes 属性 9-2
 SeverityLevel 属性 9-3
 SubsystemNames 属性 9-4
 UserIds 属性 9-4
DriverClassName 属性 19-2
DriverName 属性 17-4
DurabilityEnabled 属性 43-5

E

EJBComponent 要素
 DeploymentOrder 属性 10-2
 ExtraEjbcOptions 属性 10-2
 ExtraRmicOptions 属性 10-3
 ForceGeneration 属性 10-3
 JavaCompiler 属性 10-3
 JavaCompilerPostClassPath 属性 10-3
 JavaCompilerPreClassPath 属性 10-3
 KeepGenerated 属性 10-3
 Name 属性 10-4
 Notes 属性 10-4
 Targets 属性 10-4
 TmpPath 属性 10-4

- URI 属性 10-4
- VerboseEJBDeploymentEnabled 属性 10-4
- EJBContainer 要素
 - ExtraEjbcOptions 属性 11-2
 - ExtraRmicOptions 属性 11-2
 - ForceGeneration 属性 11-2
 - JavaCompiler 属性 11-3
 - JavaCompilerPostClassPath 属性 11-3
 - JavaCompilerPreClassPath 属性 11-3
 - KeepGenerated 属性 11-3
 - Name 属性 11-3
 - Notes 属性 11-3
 - TmpPath 属性 11-3
 - VerboseEJBDeploymentEnabled 属性 11-3
- EJBName 属性 70-2
- EmbeddedLDAP 要素
 - BackupCopies 属性 12-2
 - BackupHour 属性 12-2
 - BackupMinute 属性 12-3
 - CacheEnabled 属性 12-3
 - CacheSize 属性 12-3
 - CacheTTL 属性 12-3
 - Credential 属性 12-3
 - Name 属性 12-4
 - Notes 属性 12-4
- Enabled 属性 61-5
- EnabledForDomainLog 属性 52-12
- EnabledServers 属性 56-2, 57-2, 60-2
- Enabled 属性 55-2
- EnableResourceHealthMonitoring 属性 17-4
- EnableTwoPhaseCommit 属性 22-3
- EntityURI 属性 80-2
- EnumerationAllowed 属性 48-2
- ErrorDestination 属性 30-7, 34-7, 35-7
- ExecuteQueue 要素
 - Name 属性 13-2
 - Notes 属性 13-2
 - ThreadCount 属性 13-2
- ExpectedToRun 属性 52-12
- ExportKeyLifespan 属性 61-5
- ExternalDNSName 属性 52-12
- ExtraEjbcOptions 属性 10-2, 11-2, 52-12
- ExtraRmicOptions 属性 10-3, 11-2, 52-13

F

- FactoryName 属性 19-2
- FailoverAddresses 属性 36-3, 69-4
- FailoverRequestIfBusy 属性 20-5
- FailureIsFatal 属性 62-3
- FederationName 属性 74-6
- FederationURL 属性 74-6
- FileCount 属性 40-2
- FileMinSize 属性 40-2
- FileName 属性 40-2
- FileRealm 要素
 - MaxACLs 属性 14-2
 - MaxGroups 属性 14-2
 - MaxUsers 属性 14-2
 - Name 属性 14-3
 - Notes 属性 14-3
- FileT3 要素
 - DeploymentOrder 属性 15-2
 - Name 属性 15-2
 - Notes 属性 15-2
 - Path 属性 15-3
 - Targets 属性 15-3
- FileTimeSpan 属性 40-3
- FldTbl16Classes 属性 75-2
- FldTbl32Classes 属性 75-3
- FlowControlEnabled 属性 24-7
- FlowInterval 属性 24-7
- FlowMaximum 属性 24-8
- FlowMinimum 属性 24-9
- FlowSteps 属性 24-9
- ForceGeneration 属性 10-3, 11-2
- ForgetHeuristics 属性 37-4
- ForwardDelay 属性 26-2
- FrontendHost 属性 65-4, 67-4
- FrontendHTTPPort 属性 65-4, 67-4
- FrontendHTTPSPort 属性 65-4, 67-4

G

- GuestDisabled 属性 50-3

H

HandlerEnabled 属性 61-5
HealthCheckFrequencySeconds 属性 20-5
HealthCheckIntervalSeconds 属性 52-13
HealthCheckTimeoutSeconds 属性 52-13
Host 属性 59-2
HostingServer 属性 38-2, 45-2
HostnameVerificationIgnored 属性 61-5
HostnameVerifier 属性 61-6
HttpdEnabled 属性 52-14
HTTPEEnabled 属性 46-5
HTTPSEEnabled 属性 46-5
HttpsKeepAliveSecs 属性 65-5, 67-5
HttpTraceSupportEnabled 属性 4-4, 52-14

I

IdleConnectionTimeout 属性 16-2
IdleIOPConnectionTimeout 属性 52-14
IdlePeriodsUntilTimeout 属性 4-5
IdleTimeMaximum 属性 43-5
IIOP 要素
 CompleteMessageTimeout 属性 16-2
 IdleConnectionTimeout 属性 16-2
 MaxMessageSize 属性 16-2
 Name 属性 16-2
 Notes 属性 16-2
IIOPEnabled 属性 52-14
IndexDirectoryEnabled 属性 66-3, 68-3
InitialCapacity 属性 17-5
InitialContextFactory 属性 23-3
InstrumentStackTraceEnabled 属性 52-15
InterfaceAddress 属性 52-15
Interoperate 属性 72-5

J

JavaCompiler 属性 10-3, 11-3, 52-16
JavaCompilerPostClassPath 属性 10-3, 11-3, 52-16
JavaCompilerPreClassPath 属性 10-3, 11-3, 52-16
JavaHome 属性 53-2

JDBCConnectionPool 要素

ACLName 属性 17-2
CapacityIncrement 属性 17-2
ConnLeakProfilingEnabled 属性 17-3
CountOfRefreshFailuresTillDisable 属性 17-3
CountOfTestFailuresTillFlush 属性 17-3
DeploymentOrder 属性 17-4
DriverName 属性 17-4
EnableResourceHealthMonitoring 属性 17-4
InitialCapacity 属性 17-5
LoginDelaySeconds 属性 17-5
MaxCapacity 属性 17-5
Name 属性 17-5
Notes 属性 17-5
Password 属性 17-6
PreparedStatementCacheSize 属性 17-6
Properties 属性 17-6
RefreshMinutes 属性 17-6
ShrinkingEnabled 属性 17-7
ShrinkPeriodMinutes 属性 17-7
SqlStmtProfilingEnabled 属性 17-7
SupportsLocalTransaction 属性 17-7
Targets 属性 17-7
TestConnectionsOnRelease 属性 17-8
TestConnectionsOnReserve 属性 17-8
TestTableName 属性 17-8
URL 属性 17-8
XAPassword 属性 17-9
XAPreparedStatementCacheSize 属性 17-9

JDBCDataSource 要素

ConnectionWaitPeriod 属性 18-3
DeploymentOrder 属性 18-3
JNDIName 属性 18-4
Name 属性 18-4
Notes 属性 18-4
PoolName 属性 18-4
RowPrefetchEnabled 属性 18-4
RowPrefetchSize 属性 18-5
StreamChunkSize 属性 18-5
Targets 属性 18-5
Type 属性 18-5

- WaitForConnectionEnabled 属性 18-5
- JDBCDataSourceFactory 要素
 - DriverClassName 属性 19-2
 - FactoryName 属性 19-2
 - Name 属性 19-2
 - Notes 属性 19-2
 - Properties 属性 19-2
 - URL 属性 19-3
 - UserName 属性 19-3
- JDBCLogFileNames 属性 52-15
- JDBCLoggingEnabled 属性 52-15
- JDBCMultiPool 要素
 - ACLName 属性 20-3
 - AlgorithmType 属性 20-3
 - ConnectionPoolFailoverCallbackHandler 属性 20-4
 - DeploymentOrder 属性 20-4
 - FailoverRequestIfBusy 属性 20-5
 - HealthCheckFrequencySeconds 属性 20-5
 - Name 属性 20-5
 - Notes 属性 20-5
 - PoolList 属性 20-5
 - Targets 属性 20-5
- JDBCPoolComponent 要素
 - DeploymentOrder 属性 21-2
 - Name 属性 21-2
 - Notes 属性 21-2
 - Targets 属性 21-2
 - URI 属性 21-2
- JDBCTxDataSource 要素
 - DeploymentOrder 属性 22-3
 - EnableTwoPhaseCommit 属性 22-3
 - JNDIName 属性 22-4
 - Name 属性 22-4
 - Notes 属性 22-4
 - PoolName 属性 22-4
 - RowPrefetchEnabled 属性 22-4
 - RowPrefetchSize 属性 22-5
 - StreamChunkSize 属性 22-5
 - Targets 属性 22-5
- JMSBridgeDestination 要素
 - AdapterJNDIName 属性 23-2
 - Classpath 属性 23-2
 - ConnectionFactoryJNDIName 属性 23-2
 - ConnectionURL 属性 23-3
 - DestinationJNDIName 属性 23-3
 - DestinationType 属性 23-3
 - InitialContextFactory 属性 23-3
 - Name 属性 23-3
 - Notes 属性 23-3
 - UserName 属性 23-4
 - UserPassword 属性 23-4
- JMSConnectionFactory 要素
 - AcknowledgePolicy 属性 24-3
 - AllowCloseInOnMessage 属性 24-4
 - ClientId 属性 24-4
 - DefaultDeliveryMode 属性 24-5
 - DefaultPriority 属性 24-5
 - DefaultRedeliveryDelay 属性 24-5
 - DefaultTimeToDeliver 属性 24-6
 - DefaultTimeToLive 属性 24-6
 - DeploymentOrder 属性 24-6
 - FlowControlEnabled 属性 24-7
 - FlowInterval 属性 24-7
 - FlowMaximum 属性 24-8
 - FlowMinimum 属性 24-9
 - FlowSteps 属性 24-9
 - JNDIName 属性 24-10
 - LoadBalancingEnabled 属性 24-10
 - MessagesMaximum 属性 24-11
 - Name 属性 24-11
 - Notes 属性 24-11
 - OverrunPolicy 属性 24-12
 - ServerAffinityEnabled 属性 24-12
 - Targets 属性 24-12
 - TransactionTimeout 属性 24-13
 - UserTransactionsEnabled 属性 24-13
 - XAConnectionFactoryEnabled 属性 24-14
 - XAServerEnabled 属性 24-14
- JMSDefaultConnectionFactoriesEnabled 属性 52-16
- JMSDestinationKey 要素
 - Direction 属性 25-2
 - KeyType 属性 25-2
 - Name 属性 25-2
 - Notes 属性 25-3

- Property 属性 25-3
- JMSDistributedQueue 要素
 - DeploymentOrder 属性 26-2
 - ForwardDelay 属性 26-2
 - JNDIName 属性 26-3
 - LoadBalancingPolicy 属性 26-3
 - Name 属性 26-3
 - Notes 属性 26-3
 - Targets 属性 26-3
- JMSDistributedTopic 要素
 - DeploymentOrder 属性 27-2
 - JNDIName 属性 27-2
 - LoadBalancingPolicy 属性 27-3
 - Name 属性 27-3
 - Notes 属性 27-3
 - Targets 属性 27-3
- JmsFactory 属性 77-4
- JMSFileStore 要素
 - Directory 属性 28-2
 - Name 属性 28-2
 - Notes 属性 28-3
 - SynchronousWritePolicy 属性 28-4
- JMSJDBCStore 要素
 - ConnectionPool 属性 29-2
 - Name 属性 29-2
 - Notes 属性 29-2
 - PrefixName 属性 29-2
- JMSQueue 要素
 - BytesMaximum 属性 30-3
 - BytesPagingEnabled 属性 30-4
 - BytesThresholdHigh 属性 30-5
 - BytesThresholdLow 属性 30-6
 - DeliveryModeOverride 属性 30-6
 - DestinationKeys 属性 30-7
 - ErrorDestination 属性 30-7
 - JNDIName 属性 30-7
 - JNDINameReplicated 属性 30-7
 - MessagesMaximum 属性 30-8
 - MessagesPagingEnabled 属性 30-9
 - MessagesThresholdHigh 属性 30-10
 - MessagesThresholdLow 属性 30-11
 - Name 属性 30-11
 - Notes 属性 30-11
 - PriorityOverride 属性 30-12
 - RedeliveryDelayOverride 属性 30-12
 - RedeliveryLimit 属性 30-13
 - StoreEnabled 属性 30-14
 - Template 属性 30-14
 - TimeToDeliverOverride 属性 30-15
 - TimeToLiveOverride 属性 30-15
- JMSServer 要素
 - BytesMaximum 属性 31-3
 - BytesPagingEnabled 属性 31-4
 - BytesThresholdHigh 属性 31-5
 - BytesThresholdLow 属性 31-6
 - DeploymentOrder 属性 31-7
 - MessagesMaximum 属性 31-7
 - MessagesPagingEnabled 属性 31-8
 - MessagesThresholdHigh 属性 31-9
 - MessagesThresholdLow 属性 31-10
 - Name 属性 31-10
 - Notes 属性 31-10
 - PagingStore 属性 31-11
 - Store 属性 31-11
 - Targets 属性 31-11
 - TemporaryTemplate 属性 31-12
- JMSSessionPool 要素
 - AcknowledgeMode 属性 32-2
 - ConnectionFactory 属性 32-2
 - ListenerClass 属性 32-2
 - Name 属性 32-2
 - Notes 属性 32-2
 - SessionsMaximum 属性 32-3
 - Transacted 属性 32-3
- JMSStore 要素
 - Name 属性 33-2
 - Notes 属性 33-2
- JMSTemplate 要素
 - BytesMaximum 属性 34-3
 - BytesPagingEnabled 属性 34-4
 - BytesThresholdHigh 属性 34-5
 - BytesThresholdLow 属性 34-6
 - DeliveryModeOverride 属性 34-6
 - DestinationKeys 属性 34-7
 - ErrorDestination 属性 34-7
 - MessagesMaximum 属性 34-7

MessagesPagingEnabled 属性 34-8
MessagesThresholdHigh 属性 34-9
MessagesThresholdLow 属性 34-10
Name 属性 34-10
Notes 属性 34-10
PriorityOverride 属性 34-11
RedeliveryDelayOverride 属性 34-11
RedeliveryLimit 属性 34-12
TimeToDeliverOverride 属性 34-13
TimeToLiveOverride 属性 34-13

JMSTopic 要素

- BytesMaximum 属性 35-3
- BytesPagingEnabled 属性 35-4
- BytesThresholdHigh 属性 35-5
- BytesThresholdLow 属性 35-6
- DeliveryModeOverride 属性 35-6
- DestinationKeys 属性 35-7
- ErrorDestination 属性 35-7
- JNDIName 属性 35-7
- JNDINameReplicated 属性 35-7
- MessagesMaximum 属性 35-8
- MessagesPagingEnabled 属性 35-9
- MessagesThresholdHigh 属性 35-10
- MessagesThresholdLow 属性 35-11
- MulticastAddress 属性 35-11
- MulticastPort 属性 35-12
- MulticastTTL 属性 35-12
- Name 属性 35-12
- Notes 属性 35-12
- PriorityOverride 属性 35-13
- RedeliveryDelayOverride 属性 35-13
- RedeliveryLimit 属性 35-14
- StoreEnabled 属性 35-15
- Template 属性 35-15
- TimeToDeliverOverride 属性 35-16
- TimeToLiveOverride 属性 35-16

JmsToTuxPriorityMap 属性 77-4

JndiFactory 属性 77-5

JNDIName 属性 18-4, 22-4, 24-10, 26-3, 27-2, 30-7, 35-7, 42-2, 49-2

JNDINameReplicated 属性 30-7, 35-7

JNDITransportableObjectFactoryList 属性 52-16

JoltConnectionPool 要素

- ApplicationPassword 属性 36-2
- DeploymentOrder 属性 36-2
- FailoverAddresses 属性 36-3
- MaximumPoolSize 属性 36-3
- MinimumPoolSize 属性 36-3
- Name 属性 36-3
- Notes 属性 36-3
- PrimaryAddresses 属性 36-3
- RecvTimeout 属性 36-3
- SecurityContextEnabled 属性 36-4
- Targets 属性 36-4
- UserName 属性 36-4
- UserPassword 属性 36-4
- UserRole 属性 36-4

JTA 要素

- AbandonTimeoutSeconds 属性 37-3
- BeforeCompletionIterationLimit 属性 37-3
- CheckpointIntervalSeconds 属性 37-4
- ForgetHeuristics 属性 37-4
- MaxResourceRequestsOnServer 属性 37-4
- MaxResourceUnavailableMillis 属性 37-4
- MaxTransactions 属性 37-4
- MaxUniqueNameStatistics 属性 37-5
- MaxXACallMillis 属性 37-5
- Name 属性 37-5
- Notes 属性 37-5
- TimeoutSeconds 属性 37-5

JTAMigratableTarget 要素

- HostingServer 属性 38-2
- Name 属性 38-2
- Notes 属性 38-2

JTARecoveryService 要素

- Name 属性 39-2
- Notes 属性 39-2

K

KeepAliveEnabled 属性 65-5, 67-5

KeepAliveSecs 属性 65-5, 67-5

KeepGenerated 属性 10-3, 11-3

KeyEncrypted 属性 61-6

KeyType 属性 25-2

L

ListenAddress 属性 47-3, 52-16
ListenDelaySecs 属性 52-16
ListenerClass 属性 32-2
ListenPort 属性 46-6, 47-3, 52-17, 61-6
ListenPortEnabled 属性 46-6, 52-17
LoadBalancingEnabled 属性 24-10
LoadBalancingPolicy 属性 26-3, 27-3
LoadBeforeAppActivation 属性 62-3
LoadBeforeAppDeployments 属性 62-4
LoadOrder 属性 2-3
LocalAccessPoint 属性 70-2, 71-2, 73-2, 74-7
LocalPassword 属性 73-2
LocalPasswordIV 属性 73-2
Log 要素
 FileCount 属性 40-2
 FileMinSize 属性 40-2
 FileName 属性 40-2
 FileTimeSpan 属性 40-3
 Name 属性 40-3
 Notes 属性 40-3
 NumberOfFilesLimited 属性 40-3
 RotationTime 属性 40-4
 RotationType 属性 40-4
LogFileBufferKBytes 属性 65-6, 67-6
LogFileFlushSecs 属性 65-6, 67-6
LogFileFormat 属性 65-6, 67-6
LogFileName 属性 65-7, 67-7
LoggingEnabled 属性 65-10, 67-9
LoginDelaySeconds 属性 17-5
LoginTimeout 属性 52-17
LoginTimeoutMillis 属性 46-6, 52-18, 61-7
LoginTimeoutMillisSSL 属性 46-7
LogRemoteExceptionsEnabled 属性 52-17
LogRotationPeriodMins 属性 65-8, 67-8
LogRotationTimeBegin 属性 65-8, 67-8
LogRotationType 属性 65-9, 67-9
LogTimeInGMT 属性 65-9, 67-9
LowMemoryGCThreshold 属性 52-18
LowMemoryGranularityLevel 属性 52-18
LowMemorySampleSize 属性 52-19
LowMemoryTimeInterval 属性 52-19

M

Machine 属性 52-21
Machine 要素
 Addresses 属性 41-2
 Name 属性 41-2
 Notes 属性 41-2
MailSession 要素
 DeploymentOrder 属性 42-2
 JNDIName 属性 42-2
 Name 属性 42-2
 Notes 属性 42-2
 Properties 属性 42-3
 Targets 属性 42-3
ManagedServerIndependenceEnabled 属性 52-21
MaxACLs 属性 14-2
MaxCapacity 属性 17-5
MaxCOMMessageSize 属性 46-7, 52-22
MaxEncryptBits 属性 72-5, 74-7
MaxGroups 属性 14-2
MaxHTTPMessageSize 属性 46-7, 52-22
MaxIIOPMessageSize 属性 52-23
MaximumEncryptionLevel 属性 69-4
MaximumPoolSize 属性 36-3, 69-5
MaxLogFileSizeKBytes 属性 65-10, 67-10
MaxMessageSize 属性 16-2
MaxOpenSockCount 属性 52-23
MaxPostSize 属性 65-10, 67-10
MaxPostTimeSecs 属性 65-10, 67-10
MaxResourceRequestsOnServer 属性 37-4
MaxResourceUnavailableMillis 属性 37-4
MaxRetries 属性 72-5, 74-7
MaxSize 属性 79-2
MaxT3MessageSize 属性 46-8, 52-23
MaxTransactions 属性 37-4
MaxUniqueNameStatistics 属性 37-5
MaxUsers 属性 14-2
MaxXACallMillis 属性 37-5
MemoryLoggingEnabled 属性 5-2
MessagesMaximum 属性 24-11, 30-8, 31-7, 34-7, 35-8
MessagesPagingEnabled 属性 30-9, 31-8, 34-8,

35-9
MessagesThresholdHigh 属性 30-10, 31-9, 34-9, 35-10
MessagesThresholdLow 属性 30-11, 31-10, 34-10, 35-11
MessagingBridge 要素
 AsyncEnabled 属性 43-3
 BatchInterval 属性 43-4
 BatchSize 属性 43-4
 DeploymentOrder 属性 43-4
 DurabilityEnabled 属性 43-5
 IdleTimeMaximum 属性 43-5
 Name 属性 43-5
 Notes 属性 43-5
 QOSDegradationAllowed 属性 43-6
 QualityOfService 属性 43-6
 ReconnectDelayIncrease 属性 43-7
 ReconnectDelayMaximum 属性 43-8
 ReconnectDelayMinimum 属性 43-9
 Selector 属性 43-9
 SourceDestination 属性 43-10
 Started 属性 43-10
 TargetDestination 属性 43-10
 Targets 属性 43-10
 TransactionTimeout 属性 43-11
MetaDataFile 属性 78-2
MibDataRefreshInterval 属性 55-3
MigratableRMIService 要素
 DeploymentOrder 属性 44-2
 Name 属性 44-2
 Notes 属性 44-2
 Targets 属性 44-2
MigratableTarget 要素
 HostingServer 属性 45-2
 Name 属性 45-2
 Notes 属性 45-2
MinEncryptBits 属性 72-6, 74-8
MinimumEncryptionLevel 属性 69-5
MinimumPoolSize 属性 36-3, 69-5
MonitoredAttributeName 属性 57-2
MonitoredMBeanName 属性 57-2
MonitoredMBeanType 属性 57-2
MSIFileReplicationEnabled 属性 52-20

MulticastAddress 属性 4-5, 35-11
MulticastBufferSize 属性 4-5
MulticastPort 属性 4-5, 35-12
MulticastSendDelay 属性 4-5
MulticastTTL 属性 4-6, 35-12

N

Name 属性 2-3, 3-3, 4-6, 5-3, 6-2, 7-2, 8-6, 9-2, 10-4, 11-3, 12-4, 13-2, 14-3, 15-2, 16-2, 17-5, 18-4, 19-2, 20-5, 21-2, 22-4, 23-3, 24-11, 25-2, 26-3, 27-3, 28-2, 29-2, 30-11, 31-10, 32-2, 33-2, 34-10, 35-12, 36-3, 37-5, 38-2, 39-2, 40-3, 41-2, 42-2, 43-5, 44-2, 45-2, 46-8, 47-3, 48-2, 49-2, 50-3, 51-2, 52-24, 53-2, 54-3, 55-3, 56-2, 57-2, 58-2, 59-2, 60-2, 61-7, 62-4, 63-2, 64-2, 65-11, 66-3, 67-10, 68-3, 69-5, 70-2, 71-2, 72-6, 73-2, 74-8, 75-3, 76-3, 77-5, 78-3, 79-3, 80-2, 81-2, 82-2
NativeIOEnabled 属性 52-24
NativeModeEnabled 属性 5-3
NetworkAccessPoints 属性 52-24
NetworkChannel 要素
 AcceptBacklog 属性 46-3
 ChannelWeight 属性 46-3
 ClusterAddress 属性 46-3
 COMEnabled 属性 46-3
 CompleteCOMMessageTimeout 属性 46-4
 CompleteHTTPMessageTimeout 属性 46-4
 CompleteT3MessageTimeout 属性 46-4
 DeploymentOrder 属性 46-5
 Description 属性 46-5
 HTTPEEnabled 属性 46-5
 HTTPSEEnabled 属性 46-5
 ListenPort 属性 46-6
 ListenPortEnabled 属性 46-6
 LoginTimeoutMillis 属性 46-6
 LoginTimeoutMillisSSL 属性 46-7
 MaxCOMMessageSize 属性 46-7
 MaxHTTPMessageSize 属性 46-7
 MaxT3MessageSize 属性 46-8
 Name 属性 46-8

Notes 属性 46-8
OutgoingEnabled 属性 46-8
SSLListenPort 属性 46-9
SSLListenPortEnabled 属性 46-9
T3Enabled 属性 46-9
T3SEnabled 属性 46-9
Targets 属性 46-9
TunnelingClientPingSecs 属性 46-10
TunnelingClientTimeoutSecs 属性 46-10
TunnelingEnabled 属性 46-10
NodeManager 要素
Certificate 属性 47-2
CertificatePassword 属性 47-2
CertificateType 属性 47-2
ListenAddress 属性 47-3
ListenPort 属性 47-3
Name 属性 47-3
Notes 属性 47-3
TrustedCertsFile 属性 47-3
Notes 属性 2-3, 3-3, 4-6, 5-3, 6-2, 7-2, 8-6, 9-2,
10-4, 11-3, 12-4, 13-2, 14-3, 15-2, 16-2,
17-5, 18-4, 19-2, 20-5, 21-2, 22-4, 23-3,
24-11, 25-3, 26-3, 27-3, 28-3, 29-2,
30-11, 31-10, 32-2, 33-2, 34-10, 35-12,
36-3, 37-5, 38-2, 39-2, 40-3, 41-2, 42-2,
43-5, 44-2, 45-2, 46-8, 47-3, 48-2, 49-2,
50-3, 51-2, 52-24, 53-3, 54-3, 55-3,
56-2, 57-3, 58-2, 59-2, 60-2, 61-7, 62-4,
63-2, 64-2, 65-11, 66-3, 67-11, 68-3,
69-5, 70-2, 71-2, 72-7, 73-2, 74-9, 75-3,
76-3, 77-5, 78-3, 79-3, 80-3, 81-2, 82-2
NTAuthHost 属性 5-2
NumberOfFilesLimited 属性 40-3
NWAddr 属性 72-6, 74-8

O

OidRoot 属性 58-2
OutgoingEnabled 属性 46-8
OverrunPolicy 属性 24-12

P

PagingStore 属性 31-11
ParserClassName 属性 81-2
Password 属性 7-2, 17-6, 53-3
PasswordPolicy 属性 50-3
Path 属性 2-4, 15-3
PollingInterval 属性 57-3
PoolList 属性 20-5
PoolName 属性 18-4, 22-4
Port 属性 58-3, 59-2
PostBindGID 属性 63-3
PostBindGIDEnabled 属性 63-3
PostBindUID 属性 63-3
PostBindUIDEnabled 属性 63-3
PostTimeoutSecs 属性 65-11, 67-11
PreferredSecondaryGroup 属性 52-24
PreferWebInfClasses 属性 66-3, 68-3
PrefetchEnums 属性 5-3
PrefixName 属性 29-2
PreparedStatementCacheSize 属性 17-6
PrimaryAddresses 属性 36-3, 69-6
PriorityOverride 属性 30-12, 34-11, 35-13
Properties 属性 3-4, 17-6, 19-2, 42-3
Property 属性 25-3
PublicId 属性 80-3, 81-3

Q

QOSDegradationAllowed 属性 43-6
QualityOfService 属性 43-6

R

Realm 属性 50-3
Realm 要素
CachingRealm 属性 48-2
EnumerationAllowed 属性 48-2
Name 属性 48-2
Notes 属性 48-2
ResultsBatchSize 属性 48-2
RealmClassName 属性 7-2
ReconnectDelayIncrease 属性 43-7
ReconnectDelayMaximum 属性 43-8

ReconnectDelayMinimum 属性 43-9
RecvTimeout 属性 36-3
RedeliveryDelayOverride 属性 30-12, 34-11, 35-13
RedeliveryLimit 属性 30-13, 34-12, 35-14
RefreshMinutes 属性 17-6
RemoteAccessPoint 属性 73-3
RemoteAccessPointList 属性 71-2
RemoteName 属性 70-3, 71-3
RemotePassword 属性 73-3
RemotePasswordIV 属性 73-3
ReplicationGroup 属性 52-25
ReplyQ 属性 78-3
ResourceName 属性 70-3, 71-3
RestartDelaySeconds 属性 52-25
RestartIntervalSeconds 属性 52-25
RestartMax 属性 52-25
ResultsBatchSize 属性 48-2
Retries 属性 77-5
RetryDelay 属性 77-5
RetryInterval 属性 72-7, 74-9
ReverseDNSAllowed 属性 52-26
RMCFactory 要素
 DeploymentOrder 属性 49-2
 JNDIName 属性 49-2
 Name 属性 49-2
 Notes 属性 49-2
 Targets 属性 49-2
RootDirectory 属性 53-3
RootElementTag 属性 81-3
RotationTime 属性 40-4
RotationType 属性 40-4
RowPrefetchEnabled 属性 18-4, 22-4
RowPrefetchSize 属性 18-5, 22-5

S

SAXParserFactory 属性 81-3, 82-3
Security 属性 72-7
Security 要素
 AuditProviderClassName 属性 50-2
 CompatibilityMode 属性 50-2
 ConnectionFactory 属性 50-2

ConnectionFactoryRules 属性 50-2
ConnectionFactoryEnabled 属性 50-3
GuestDisabled 属性 50-3
Name 属性 50-3
Notes 属性 50-3
PasswordPolicy 属性 50-3
Realm 属性 50-3
SystemUser 属性 50-3
SecurityConfiguration element
 WebAppFilesCaseInsensitive attribute 51-2
SecurityConfiguration 要素
 Name 属性 51-2
 Notes 属性 51-2
SecurityContextEnabled 属性 36-4, 69-6
SecurityPolicyFile 属性 53-3
Selector 属性 43-9
SendServerHeaderEnabled 属性 65-11, 67-11
Server 要素
 AcceptBacklog 属性 52-5
 AdministrationPort 属性 52-5
 AutoKillIfFailed 属性 52-6
 AutoRestart 属性 52-6
 ClassServletDisabled 属性 52-6
 ClientCertProxyEnabled 属性 52-7
 Cluster 属性 52-8
 ClusterWeight 属性 52-8
 COMEnabled 属性 52-6
 CompleteCOMMessageTimeout 属性 52-8
 CompleteHTTPMessageTimeout 属性 52-9
 CompleteIIOPMessageTimeout 属性 52-9
 CompleteT3MessageTimeout 属性 52-10
 DefaultIIOPPassword 属性 52-10
 DefaultIIOPUser 属性 52-10
 DefaultInternalServletsDisabled 属性 52-11
 DefaultProtocol 属性 52-11
 DefaultSecureProtocol 属性 52-11
 DefaultTGIOPPassword 属性 52-11
 DefaultTGIOPUser 属性 52-11
 DomainLogFilter 属性 52-12
 EnabledForDomainLog 属性 52-12
 ExpectedToRun 属性 52-12
 ExternalDNSName 属性 52-12
 ExtraEjbcOptions 属性 52-12

ExtraRmicOptions 属性 52-13
 HealthCheckIntervalSeconds 属性 52-13
 HealthCheckTimeoutSeconds 属性 52-13
 HttpdEnabled 属性 52-14
 HttpTraceSupportEnabled 属性 52-14
 IdleIIOConnectionTimeout 属性 52-14
 IIOEnabled 属性 52-14
 InstrumentStackTraceEnabled 属性 52-15
 InterfaceAddress 属性 52-15
 JavaCompiler 属性 52-16
 JavaCompilerPostClassPath 属性 52-16
 JavaCompilerPreClassPath 属性 52-16
 JDBCLogFileFileName 属性 52-15
 JDBCLoggingEnabled 属性 52-15
 JMSDefaultConnectionFactoryEnabled 属性 52-16
 JNDITransportableObjectFactoryList 属性 52-16
 ListenAddress 属性 52-16
 ListenDelaySecs 属性 52-16
 ListenPort 属性 52-17
 ListenPortEnabled 属性 52-17
 LoginTimeout 属性 52-17
 LoginTimeoutMillis 属性 52-18
 LogRemoteExceptionsEnabled 属性 52-17
 LowMemoryGCThreshold 属性 52-18
 LowMemoryGranularityLevel 属性 52-18
 LowMemorySampleSize 属性 52-19
 LowMemoryTimeInterval 属性 52-19
 Machine 属性 52-21
 ManagedServerIndependenceEnabled 属性 52-21
 MaxCOMMessageSize 属性 52-22
 MaxHTTPMessageSize 属性 52-22
 MaxIIOMessageSize 属性 52-23
 MaxOpenSockCount 属性 52-23
 MaxT3MessageSize 属性 52-23
 MSIFileReplicationEnabled 属性 52-20
 Name 属性 52-24
 NativeIOEnabled 属性 52-24
 NetworkAccessPoints 属性 52-24
 Notes 属性 52-24
 PreferredSecondaryGroup 属性 52-24
 ReplicationGroup 属性 52-25
 RestartDelaySeconds 属性 52-25
 RestartIntervalSeconds 属性 52-25
 RestartMax 属性 52-25
 ReverseDNSAllowed 属性 52-26
 ServerLifeCycleTimeoutVal 属性 52-26
 ServerVersion 属性 52-26
 StagingMode 属性 52-27
 StartupMode 属性 52-27
 StdoutDebugEnabled 属性 52-27
 StdoutEnabled 属性 52-28
 StdoutFormat 属性 52-28
 StdoutLogStack 属性 52-28
 StdoutSeverityLevel 属性 52-29
 StuckThreadMaxTime 属性 52-30
 StuckThreadTimerInterval 属性 52-30
 TGIOPENabled 属性 52-30
 ThreadPoolPercentSocketReaders 属性 52-30
 TransactionLogFilePrefix 属性 52-31
 TransactionLogFileWritePolicy 属性 52-32
 TunnelingClientPingSecs 属性 52-33
 TunnelingClientTimeoutSecs 属性 52-33
 TunnelingEnabled 属性 52-33
 UploadDirectoryName 属性 52-33
 VerboseEJBDeploymentEnabled 属性 52-34
 WeblogicPluginEnabled 属性 52-34
 XMLEntityCache 属性 52-34
 XMLRegistry 属性 52-34
 ServerAffinityEnabled 属性 24-12
 ServerCertificateChainFileName 属性 61-7
 ServerCertificateFileName 属性 61-7
 ServerKeyFileName 属性 61-8
 ServerLifeCycleTimeoutVal 属性 52-26
 ServerPrivateKeyAlias 属性 61-8
 ServerPrivateKeyPassPhrase 属性 61-8
 ServerStart 要素
 Arguments 属性 53-2
 BeaHome 属性 53-2
 ClassPath 属性 53-2
 JavaHome 属性 53-2
 Name 属性 53-2

- Notes 属性 53-3
- Password 属性 53-3
- RootDirectory 属性 53-3
- SecurityPolicyFile 属性 53-3
- Username 属性 53-3
- ServerStatusCheckIntervalFactor 属性 55-4
- ServerVersion 属性 52-26
- ServletReloadCheckSecs 属性 66-4, 68-4
- SessionMonitoringEnabled 属性 66-4, 68-4
- SessionsMaximum 属性 32-3
- SeverityLevel 属性 9-3
- ShrinkingEnabled 属性 17-7
- ShrinkPeriodMinutes 属性 17-7
- ShutdownClass 要素
 - Arguments 属性 54-2
 - ClassName 属性 54-2
 - DeploymentOrder 属性 54-2
 - Name 属性 54-3
 - Notes 属性 54-3
 - Targets 属性 54-3
- SingleSignonDisabled 属性 65-11, 67-11
- SingleThreadedServletPoolSize 属性 66-4, 68-4
- SNMPAgent 要素
 - CommunityPrefix 属性 55-2
 - DebugLevel 属性 55-2
 - Enabled 属性 55-2
 - MibDataRefreshInterval 属性 55-3
 - Name 属性 55-3
 - Notes 属性 55-3
 - ServerStatusCheckIntervalFactor 属性 55-4
 - SNMPAttributeChanges 属性 55-3
 - SNMPCounterMonitors 属性 55-3
 - SNMPGaugeMonitors 属性 55-3
 - SNMPLogFilters 属性 55-3
 - SNMPPort 属性 55-4
 - SNMPProxies 属性 55-4
 - SNMPStringMonitors 属性 55-4
 - TargetedTrapDestinations 属性 55-4
- SNMPAttributeChange 要素
 - AttributeMBeanName 属性 56-2
 - AttributeMBeanType 属性 56-2
 - AttributeName 属性 56-2
 - EnabledServers 属性 56-2
 - Name 属性 56-2
 - Notes 属性 56-2
- SNMPAttributeChanges 属性 55-3
- SNMPCounterMonitors 属性 55-3
- SNMPGaugeMonitors 属性 55-3
- SNMPJMXMonitor 要素
 - EnabledServers 属性 57-2
 - MonitoredAttributeName 属性 57-2
 - MonitoredMBeanName 属性 57-2
 - MonitoredMBeanType 属性 57-2
 - Name 属性 57-2
 - Notes 属性 57-3
 - PollingInterval 属性 57-3
- SNMPLogFilters 属性 55-3
- SNMPPort 属性 55-4
- SNMPProxies 属性 55-4
- SNMPProxy 要素
 - Community 属性 58-2
 - Name 属性 58-2
 - Notes 属性 58-2
 - OidRoot 属性 58-2
 - Port 属性 58-3
 - Timeout 属性 58-3
- SNMPStringMonitors 属性 55-4
- SNMPTrapDestination 要素
 - Community 属性 59-2
 - Host 属性 59-2
 - Name 属性 59-2
 - Notes 属性 59-2
 - Port 属性 59-2
- SNMPTrapSource 要素
 - EnabledServers 属性 60-2
 - Name 属性 60-2
 - Notes 属性 60-2
- SourceAccessPoint 属性 78-3
- SourceDestination 属性 43-10
- SourceName 属性 78-3
- SourceQspace 属性 78-3
- SqlStmtProfilingEnabled 属性 17-7
- SSL 要素
 - CertAuthenticator 属性 61-3
 - CertificateCacheSize 属性 61-3
 - Ciphersuites 属性 61-4

- ClientCertificateEnforced 属性 61-4
- Enabled 属性 61-5
- ExportKeyLifespan 属性 61-5
- HandlerEnabled 属性 61-5
- HostnameVerificationIgnored 属性 61-5
- HostnameVerifier 属性 61-6
- KeyEncrypted 属性 61-6
- ListenPort 属性 61-6
- LoginTimeoutMillis 属性 61-7
- Name 属性 61-7
- Notes 属性 61-7
- ServerCertificateChainFileName 属性 61-7
- ServerCertificateFileName 属性 61-7
- ServerKeyFileName 属性 61-8
- ServerPrivateKeyAlias 属性 61-8
- ServerPrivateKeyPassPhrase 属性 61-8
- TrustedCAFileName 属性 61-8
- TwoWaySSEnabled 属性 61-9
- UseJava 属性 61-9
- SSLListenPort 属性 46-9
- SSLListenPortEnabled 属性 46-9
- StagingMode 属性 2-5, 52-27
- Started 属性 43-10
- StartupClass 要素
 - Arguments 属性 62-2
 - ClassName 属性 62-2
 - DeploymentOrder 属性 62-2
 - FailureIsFatal 属性 62-3
 - LoadBeforeAppActivation 属性 62-3
 - LoadBeforeAppDeployments 属性 62-4
 - Name 属性 62-4
 - Notes 属性 62-4
 - Targets 属性 62-4
- StartupMode 属性 52-27
- StdoutDebugEnabled 属性 52-27
- StdoutEnabled 属性 52-28
- StdoutFormat 属性 52-28
- StdoutLogStack 属性 52-28
- StdoutSeverityLevel 属性 52-29
- Store 属性 31-11
- StoreEnabled 属性 30-14, 35-15
- StreamChunkSize 属性 18-5, 22-5
- StuckThreadMaxTime 属性 52-30

- StuckThreadTimerInterval 属性 52-30
- SubsystemNames 属性 9-4
- SupportsLocalTransaction 属性 17-7
- SynchronousWritePolicy 属性 28-4
- SystemId 属性 80-3, 81-3
- SystemUser 属性 50-3

T

- T3Enabled 属性 46-9
- T3SEnabled 属性 46-9
- TargetAccessPoint 属性 78-3
- TargetDestination 属性 43-10
- TargetedTrapDestinations 属性 55-4
- TargetName 属性 78-3
- TargetQspace 属性 78-4
- Targets 属性 6-2, 10-4, 15-3, 17-7, 18-5, 20-5, 21-2, 22-5, 24-12, 26-3, 27-3, 31-11, 36-4, 42-3, 43-10, 44-2, 46-9, 49-2, 54-3, 62-4, 65-11, 66-4, 67-11, 68-4, 69-6, 76-3
- Template 属性 30-14, 35-15
- TemporaryTemplate 属性 31-12
- TestConnectionsOnRelease 属性 17-8
- TestConnectionsOnReserve 属性 17-8
- TestTableName 属性 17-8
- TGIOPEnabled 属性 52-30
- ThreadCount 属性 13-2
- ThreadPoolPercentSocketReaders 属性 52-30
- Timeout 属性 58-3, 77-6
- TimeoutSeconds 属性 37-5
- TimeToDeliverOverride 属性 30-15, 34-13, 35-16
- TimeToLiveOverride 属性 30-15, 34-13, 35-16
- TmpPath 属性 10-4, 11-3
- TpUsrFile 属性 74-9
- Transacted 属性 32-3
- Transactional 属性 77-6
- TransactionLogFilePrefix 属性 52-31
- TransactionLogFileWritePolicy 属性 52-32
- TransactionTimeout 属性 24-13, 43-11
- TransformerFactory 属性 81-3, 82-3
- TranslateFML 属性 78-4

TrustedCAFileName 属性 61-8
TrustedCertsFile 属性 47-3
TunnelingClientPingSecs 属性 46-10, 52-33
TunnelingClientTimeoutSecs 属性 46-10, 52-33
TunnelingEnabled 属性 46-10, 52-33
TuxErrorQueue 属性 77-6
TuxFactory 属性 77-7
TuxToJmsPriorityMap 属性 77-7
TwoWaySSEnabled 属性 61-9
Type 属性 18-5

U

UnixMachine 要素
 Addresses 属性 63-2
 Name 属性 63-2
 Notes 属性 63-2
 PostBindGID 属性 63-3
 PostBindGIDEnabled 属性 63-3
 PostBindUID 属性 63-3
 PostBindUIDEnabled 属性 63-3
UnixRealm 要素
 AuthProgram 属性 64-2
 Name 属性 64-2
 Notes 属性 64-2
UploadDirectoryName 属性 52-33
URI 属性 6-2, 10-4, 21-2, 66-4, 68-4
URL 属性 17-8, 19-3
URLResource 属性 65-11, 67-11
UseHeaderEncoding 属性 65-12, 67-11
UseHighestCompatibleHTTPVersion 属性
 65-12, 67-12
UseJava 属性 61-9
UserId 属性 77-8
UserIds 属性 9-4
UserName 属性 3-5, 19-3, 23-4, 36-4, 69-7
Username 属性 53-3
UserPassword 属性 3-5, 23-4, 36-4, 69-7
UserRole 属性 36-4, 69-7
UserTransactionsEnabled 属性 24-13

V

VerboseEJBDeploymentEnabled 属性 10-4,
 11-3, 52-34
VerboseLoggingEnabled 属性 5-3
ViewTbl16Classes 属性 75-3
ViewTbl32Classes 属性 75-3
VirtualHost 要素
 AcceptContextPathInGetRealPath 属性
 65-3
 AuthCookieEnabled 属性 65-3
 Charsets 属性 65-3
 ChunkedTransferDisabled 属性 65-3
 ClusteringEnabled 属性 65-3
 DefaultServerName 属性 65-4
 DefaultWebApp 属性 65-4
 DeploymentOrder 属性 65-4
 FrontendHost 属性 65-4
 FrontendHTTPPort 属性 65-4
 FrontendHTTPSPort 属性 65-4
 HttpsKeepAliveSecs 属性 65-5
 KeepAliveEnabled 属性 65-5
 KeepAliveSecs 属性 65-5
 LogFileBufferKBytes 属性 65-6
 LogFileFlushSecs 属性 65-6
 LogFileFormat 属性 65-6
 LogFileName 属性 65-7
 LoggingEnabled 属性 65-10
 LogRotationPeriodMins 属性 65-8
 LogRotationTimeBegin 属性 65-8
 LogRotationType 属性 65-9
 LogTimeInGMT 属性 65-9
 MaxLogFileSizeKBytes 属性 65-10
 MaxPostSize 属性 65-10
 MaxPostTimeSecs 属性 65-10
 Name 属性 65-11
 Notes 属性 65-11
 PostTimeoutSecs 属性 65-11
 SendServerHeaderEnabled 属性 65-11
 SingleSignonDisabled 属性 65-11
 Targets 属性 65-11
 URLResource 属性 65-11
 UseHeaderEncoding 属性 65-12

UseHighestCompatibleHTTPVersion 属性
65-12
VirtualHostNames 属性 65-12
WAPEnabled 属性 65-12
VirtualHostNames 属性 65-12
VirtualHosts 属性 66-4, 68-4

W

WaitForConnectionEnabled 属性 18-5
WAPEnabled 属性 65-12, 67-12
WebAppComponent 要素
AuthFilter 属性 66-2
AuthRealmName 属性 66-2
DeploymentOrder 属性 66-3
IndexDirectoryEnabled 属性 66-3
Name 属性 66-3
Notes 属性 66-3
PreferWebInfClasses 属性 66-3
ServletReloadCheckSecs 属性 66-4
SessionMonitoringEnabled 属性 66-4
SingleThreadedServletPoolSize 属性 66-4
Targets 属性 66-4
URI 属性 66-4
VirtualHosts 属性 66-4
WebAppFilesCaseInsensitive attribute 51-2
WeblogicPluginEnabled 属性 4-6, 52-34
WebServer 要素
AcceptContextPathInGetRealPath 属性
67-3
AuthCookieEnabled 属性 67-3
Charsets 属性 67-3
ChunkedTransferDisabled 属性 67-3
ClusteringEnabled 属性 67-3
DefaultServerName 属性 67-3
DefaultWebApp 属性 67-4
DeploymentOrder 属性 67-4
FrontendHost 属性 67-4
FrontendHTTPPort 属性 67-4
FrontendHTTPSPort 属性 67-4
HttpsKeepAliveSecs 属性 67-5
KeepAliveEnabled 属性 67-5
KeepAliveSecs 属性 67-5

LogFileBufferKBytes 属性 67-6
LogFileFlushSecs 属性 67-6
LogFileFormat 属性 67-6
LogFileName 属性 67-7
LoggingEnabled 属性 67-9
LogRotationPeriodMins 属性 67-8
LogRotationTimeBegin 属性 67-8
LogRotationType 属性 67-9
LogTimeInGMT 属性 67-9
MaxLogFileSizeKBytes 属性 67-10
MaxPostSize 属性 67-10
MaxPostTimeSecs 属性 67-10
Name 属性 67-10
Notes 属性 67-11
PostTimeoutSecs 属性 67-11
SendServerHeaderEnabled 属性 67-11
SingleSignonDisabled 属性 67-11
Targets 属性 67-11
URLResource 属性 67-11
UseHeaderEncoding 属性 67-11
UseHighestCompatibleHTTPVersion 属性
67-12
WAPEnabled 属性 67-12
WebServiceComponent 要素
AuthFilter 属性 68-2
AuthRealmName 属性 68-2
DeploymentOrder 属性 68-3
IndexDirectoryEnabled 属性 68-3
Name 属性 68-3
Notes 属性 68-3
PreferWebInfClasses 属性 68-3
ServletReloadCheckSecs 属性 68-4
SessionMonitoringEnabled 属性 68-4
SingleThreadedServletPoolSize 属性 68-4
Targets 属性 68-4
URI 属性 68-4
VirtualHosts 属性 68-4
WhenToCache 属性 80-3, 82-3
WLECConnectionPool 要素
ApplicationPassword 属性 69-3
CertificateAuthenticationEnabled 属性 69-3
DeploymentOrder 属性 69-4
FailoverAddresses 属性 69-4

- MaximumEncryptionLevel 属性 69-4
- MaximumPoolSize 属性 69-5
- MinimumEncryptionLevel 属性 69-5
- MinimumPoolSize 属性 69-5
- Name 属性 69-5
- Notes 属性 69-5
- PrimaryAddresses 属性 69-6
- SecurityContextEnabled 属性 69-6
- Targets 属性 69-6
- UserName 属性 69-7
- UserPassword 属性 69-7
- UserRole 属性 69-7
- WLEDomain 属性 69-7
- WLEDomain 属性 69-7
- WlsErrorDestination 属性 77-8
- WTCEXport 要素
 - EJBName 属性 70-2
 - LocalAccessPoint 属性 70-2
 - Name 属性 70-2
 - Notes 属性 70-2
 - RemoteName 属性 70-3
 - ResourceName 属性 70-3
- WTCImport 要素
 - LocalAccessPoint 属性 71-2
 - Name 属性 71-2
 - Notes 属性 71-2
 - RemoteAccessPointList 属性 71-2
 - RemoteName 属性 71-3
 - ResourceName 属性 71-3
- WTCLocalTuxDom 要素
 - AccessPoint 属性 72-2
 - AccessPointId 属性 72-2
 - BlockTime 属性 72-3
 - CmpLimit 属性 72-3
 - ConnectionPolicy 属性 72-4
 - ConnPrincipalName 属性 72-3
 - Interoperate 属性 72-5
 - MaxEncryptBits 属性 72-5
 - MaxRetries 属性 72-5
 - MinEncryptBits 属性 72-6
 - Name 属性 72-6
 - Notes 属性 72-7
 - NWAddr 属性 72-6
 - RetryInterval 属性 72-7
 - Security 属性 72-7
- WTCPassword 要素
 - LocalAccessPoint 属性 73-2
 - LocalPassword 属性 73-2
 - LocalPasswordIV 属性 73-2
 - Name 属性 73-2
 - Notes 属性 73-2
 - RemoteAccessPoint 属性 73-3
 - RemotePassword 属性 73-3
 - RemotePasswordIV 属性 73-3
- WTCRemoteTuxDom 要素
 - AccessPoint 属性 74-2
 - AccessPointId 属性 74-2
 - AclPolicy 属性 74-3
 - CmpLimit 属性 74-3
 - ConnectionPolicy 属性 74-5
 - ConnPrincipalName 属性 74-4
 - CredentialPolicy 属性 74-6
 - FederationName 属性 74-6
 - FederationURL 属性 74-6
 - LocalAccessPoint 属性 74-7
 - MaxEncryptBits 属性 74-7
 - MaxRetries 属性 74-7
 - MinEncryptBits 属性 74-8
 - Name 属性 74-8
 - Notes 属性 74-9
 - NWAddr 属性 74-8
 - RetryInterval 属性 74-9
 - TpUsrFile 属性 74-9
- WTCResources 要素
 - AppPassword 属性 75-2
 - AppPasswordIV 属性 75-2
 - FldTbl16Classes 属性 75-2
 - FldTbl32Classes 属性 75-3
 - Name 属性 75-3
 - Notes 属性 75-3
 - ViewTbl16Classes 属性 75-3
 - ViewTbl32Classes 属性 75-3
- WTCServer 要素
 - DeploymentOrder 属性 76-3
 - Name 属性 76-3
 - Notes 属性 76-3

- Targets 属性 76-3
- WTCtBridgeGlobal 要素
 - AllowNonStandardTypes 属性 77-3
 - DefaultReplyDeliveryMode 属性 77-3
 - DeliveryModeOverride 属性 77-4
 - JmsFactory 属性 77-4
 - JmsToTuxPriorityMap 属性 77-4
 - JndiFactory 属性 77-5
 - Name 属性 77-5
 - Notes 属性 77-5
 - Retries 属性 77-5
 - RetryDelay 属性 77-5
 - Timeout 属性 77-6
 - Transactional 属性 77-6
 - TuxErrorQueue 属性 77-6
 - TuxFactory 属性 77-7
 - TuxToJmsPriorityMap 属性 77-7
 - UserId 属性 77-8
 - WlsErrorDestination 属性 77-8
- WTCtBridgeRedirect 要素
 - Direction 属性 78-2
 - MetaDataFile 属性 78-2
 - Name 属性 78-3
 - Notes 属性 78-3
 - ReplyQ 属性 78-3
 - SourceAccessPoint 属性 78-3
 - SourceName 属性 78-3
 - SourceQspace 属性 78-3
 - TargetAccessPoint 属性 78-3
 - TargetName 属性 78-3
 - TargetQspace 属性 78-4
 - TranslateFML 属性 78-4

X

- XAConnectionFactoryEnabled 属性 24-14
- XAPassword 属性 17-9
- XAPreparedStatementCacheSize 属性 17-9
- XAServerEnabled 属性 24-14
- XMLEntityCache 属性 52-34
- XMLEntityCache 要素
 - CacheDiskSize 属性 79-2
 - CacheLocation 属性 79-2

- CacheMemorySize 属性 79-2
- CacheTimeoutInterval 属性 79-2
- MaxSize 属性 79-2
- Name 属性 79-3
- Notes 属性 79-3
- XMLEntitySpecRegistryEntry 要素
 - CacheTimeoutInterval 属性 80-2
 - EntityURI 属性 80-2
 - Name 属性 80-2
 - Notes 属性 80-3
 - PublicId 属性 80-3
 - SystemId 属性 80-3
 - WhenToCache 属性 80-3
- XMLParserSelectRegistryEntry 要素
 - DocumentBuilderFactory 属性 81-2
 - Name 属性 81-2
 - Notes 属性 81-2
 - ParserClassName 属性 81-2
 - PublicId 属性 81-3
 - RootElementTag 属性 81-3
 - SAXParserFactory 属性 81-3
 - SystemId 属性 81-3
 - TransformerFactory 属性 81-3
- XMLRegistry 属性 52-34
- XMLRegistry 要素
 - DocumentBuilderFactory 属性 82-2
 - Name 属性 82-2
 - Notes 属性 82-2
 - SAXParserFactory 属性 82-3
 - TransformerFactory 属性 82-3
 - WhenToCache 属性 82-3

い

- 印刷、製品のマニュアル xxii

か

- カスタマ サポート情報 xxiii

さ

- サポート

技術情報 xxiii

ま

マニュアル、入手先 xxii