



# BEA WebLogic Server™

## WebLogic Tuxedo Connector 管理ガイド

## 著作権

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

## 限定的権利条項

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

**WebLogic Tuxedo Connector** 管理ガイド

---

# 目次

## このマニュアルの内容

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

## 1. WebLogic Tuxedo Connector の概要

WebLogic Tuxedo Connector の概要.....	1-1
主要な機能と管理機能.....	1-2
確認済みの制限.....	1-3
WebLogic Tuxedo Connector と Jolt の相違点.....	1-3
マニュアル.....	1-4
プラットフォームのサポート.....	1-4
ライセンス.....	1-4
WebLogic Tuxedo Connector 6.x アプリケーションのアップグレード.....	1-5

## 2. WebLogic Tuxedo Connector のコンフィグレーション

環境の変更と考慮事項の概要.....	2-1
Tuxedo の変更.....	2-1
WebLogic Server の変更.....	2-2
管理とプログラミング.....	2-2
WebLogic Server のスレッド.....	2-2
アプリケーション用の WebLogic Tuxedo Connector のコンフィグレーション	
2-3	
WebLogic Tuxedo Connector MBean クラス.....	2-4
Administration Console を使用した WebLogic Tuxedo Connector のコン	
フィグレーション.....	2-5
コマンドライン インタフェースを使った WebLogic Tuxedo Connector の	
コンフィグレーション.....	2-7

WebLogic Server 環境の設定 .....	2-7
WebLogic Tuxedo Connector プロパティの設定方法 .....	2-8
TraceLevel の設定 .....	2-8
PasswordKey の設定 .....	2-8
encoding の設定 .....	2-9
WebLogic Tuxedo Connector のコンフィグレーション ガイドライン	2-10

### 3. WebLogic Tuxedo Connector の管理

ドメイン間の通信のコンフィグレーション .....	3-2
起動時に接続を要求する方法 (ON_STARTUP) .....	3-2
RetryInterval のコンフィグレーション方法 .....	3-2
MaxRetries のコンフィグレーション方法 .....	3-3
クライアント要求による接続のリクエスト方法 (ON_DEMAND) .....	3-3
受信時接続の受け付け (INCOMING_ONLY) .....	3-4
LOCAL 接続ポリシーの使用法 .....	3-4
ConnectionPolicy の動的ステータスへの影響 .....	3-4
ドメインレベルのフェイルオーバーとフェイルバックのコンフィグレーション	3-5
ドメインレベルのフェイルオーバーおよびフェイルバックを使用するための要件 .....	3-6
フェイルオーバーをサポートするためのドメインのコンフィグレーション	3-6
フェイルバックをサポートするためのドメインのコンフィグレーション .....	3-7
リモート ドメインの認証 .....	3-7
WTCPasswd MBean のコンフィグレーション .....	3-8
解説 .....	3-9
例 .....	3-10
LocalPasswords .....	3-10
RemotePasswords .....	3-10
AppPasswords .....	3-10
ユーザ認証 .....	3-11
ACL ポリシーは LOCAL .....	3-11
ACL ポリシーは GLobal .....	3-12
CredentialPolicy は Global .....	3-12
CredentialPolicy は Local .....	3-12

Tuxedo 6.5 のユーザ認証.....	3-13
Tuxedo と WebLogic Server の間でセキュリティを提供するように WebLogic Tuxedo Connector をコンフィグレーションする方法.....	3-13
WTCLocalTuxDom MBean のコンフィグレーション.....	3-13
WTCLocalTuxDom MBean のコンフィグレーション例 .....	3-14
Tuxedo *DM_LOCAL_DOMAINS のコンフィグレーション例 ...	3-14
WTCRemoteTuxDom MBean のコンフィグレーション .....	3-14
WTCRemoteTuxDom MBean の例 .....	3-15
Tuxedo *DM_LOCAL_DOMAINS のコンフィグレーションの例	3-15
Simpapp サンプルおよび Simpserv サンプルの ACL ポリシー例.....	3-15
リンクレベルの暗号化.....	3-21

#### 4. クラスタ環境における WebLogic Tuxedo Connector の管理方法

クラスタ環境における WebLogic Tuxedo Connector のガイドライン .....	4-1
Tuxedo ドメインに対する送信リクエストのコンフィグレーション方法....	4-2
クラスタ化した WebLogic Tuxedo Connector のコンフィグレーション例	4-3
Tuxedo ドメインからの着信リクエストのコンフィグレーション方法.....	4-5
ロード バランシング .....	4-5
フェイルオーバー .....	4-6

#### 5. CORBA アプリケーションの管理

CORBA サービスアプリケーション用 WebLogic Tuxedo Connector のコンフィグレーション方法.....	5-1
WTCTServer MBean および Tuxedo UBB ファイルの例.....	5-2
着信 RMI-IIOP の WebLogic Tuxedo Connector を管理およびコンフィグレーションする方法 .....	5-4
WTCTServer MBean のコンフィグレーション.....	5-4
Tuxedo アプリケーション環境の管理.....	5-5
オブジェクト参照としてのサーバ名使用に関するガイドライン ...	5-6
送信 RMI-IIOP の WebLogic Tuxedo Connector をコンフィグレーションする方法 .....	5-7
送信 RMI-IIOP コンフィグレーション例 .....	5-8

#### 6. tBridge メッセージインタフェースのコンフィグレーション

## ン

tBridge の概要	6-1
tBridge による JMS から Tuxedo への接続	6-2
tBridge による Tuxedo から JMS への接続	6-3
tBridge の制限	6-4
tBridge 用 WTCServer MBean コンフィグレーション	6-5
tBridge の起動	6-5
エラーのロギング	6-5
tBridge の接続性	6-5
接続タイプのコンフィグレーションの例	6-6
JmsQ2TuxQ コンフィグレーションの例	6-6
TuxQ2JmsQ コンフィグレーションの例	6-7
JmsQ2TuxS コンフィグレーションの例	6-8
優先度のマッピング	6-10
エラー キュー	6-11
wlsServerErrorDestination	6-11
サポートされていないメッセージ タイプ	6-11
tuxErrorQueue	6-11
制限	6-11

## 7. WebLogic Tuxedo Connector での FML の使用

FML の概要	7-1
WebLogic Tuxedo Connector FML API	7-2
FML フィールド テーブルの管理	7-2
mkfldclass32 クラスに対する DynRdHdr プロパティの使用	7-4
tBridge XML/FML32 変換	7-6
FLAT	7-6
NO	7-7
FML32 の考慮事項	7-7

## 8. WebLogic Process Integrator と Tuxedo アプリケーションの接続

WebLogic Process Integrator と Tuxedo の同期接続	8-1
ビジネス オペレーションの定義	8-2
eLink アダプタの呼び出し	8-2

例外ハンドラの定義 .....	8-2
WebLogic Process Integrator と Tuxedo の同期非ブロッキング接続 .....	8-2
WebLogic Process Integrator と Tuxedo の非同期接続 .....	8-3
非同期 Tuxedo /Q と WebLogic Process Integrator の接続 .....	8-4
Tuxedo と WebLogic Process Integrator の双方向非同期接続 .....	8-4

## 9. WebLogic Tuxedo Connector のトラブルシューティング

WebLogic Tuxedo Connector のモニタ .....	9-1
トレース レベルの設定 .....	9-1
デバッグ モードの有効化 .....	9-2
よくある質問 .....	9-3
この EJB デプロイメント メッセージはどういう意味ですか? .....	9-3
コネクタはどう起動するのですか? .....	9-4
tBridge はどのように起動するのですか? .....	9-4
WTCTServer MBean をサーバに割り当てるにはどのようにするのですか ? .....	9-4
接続の問題はどのように解決するのですか? .....	9-5
旧リリースからの移行はどのように行いますか? .....	9-6

## 10. WebLogic Tuxedo Connector MBean 属性

WTCTServerMBean .....	10-1
WTCTLocalTuxDomMBean .....	10-3
WTCTRemoteTuxDomMBean .....	10-9
WTCTExportMBean .....	10-16
WTCTImportMbean .....	10-17
WTCTPasswordMbean .....	10-18
WTCTResourcesMBean .....	10-19
WTCTtBridgeGlobalMBean .....	10-21
WTCTtBridgeRedirectMBean .....	10-25





---

# このマニュアルの内容

このマニュアルでは、BEA WebLogic Tuxedo Connector™ アプリケーションの開発環境を紹介します。WebLogic Server と Tuxedo との間で相互運用する WebLogic Tuxedo Connector のコンフィグレーションおよび管理方法について説明します。

このマニュアルの構成は次のとおりです。

- 第 1 章「WebLogic Tuxedo Connector の概要」では、WebLogic Tuxedo Connector の概要を説明します。
- 第 2 章「WebLogic Tuxedo Connector のコンフィグレーション」では、WebLogic Tuxedo Connector のコンフィグレーションの方法について説明します。
- 第 3 章「WebLogic Tuxedo Connector の管理」では、WebLogic Tuxedo Connector のコンフィグレーション情報について説明します。
- 第 4 章「クラスタ環境における WebLogic Tuxedo Connector の管理方法」では、クラスタ環境における WebLogic Tuxedo Connector の使用方法について説明します。
- 第 5 章「CORBA アプリケーションの管理」では、CORBA アプリケーションの管理方法について説明します。
- 第 6 章「tBridge メッセージ インタフェースのコンフィグレーション」では、tBridge 機能とコンフィグレーションについて説明します。
- 第 7 章「WebLogic Tuxedo Connector での FML の使用」では、フィールド操作言語 (FML) および WebLogic Tuxedo Connector が FML を使用方法について説明します。
- 第 8 章「WebLogic Process Integrator と Tuxedo アプリケーションの接続」では、WebLogic Tuxedo コネクタを使用して WebLogic Process Integrator と Tuxedo を統合する方法について説明します。

- 
- 第9章「WebLogic Tuxedo Connector のトラブルシューティング」では、WebLogic Tuxedo Connector のトラブルシューティング情報について説明します。
  - 第10章「WebLogic Tuxedo Connector MBean 属性」では、WebLogic Tuxedo Connector MBean および属性のリファレンス情報について説明します。

## 対象読者

このマニュアルは、WebLogic Server と Tuxedo 環境で相互に運用される分散 Java アプリケーションを構築するシステム管理者およびアプリケーション開発者を対象としています。WebLogic Server、Tuxedo、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。

## e-docs Web サイト

BEA 製品のドキュメントは、BEA の Web サイトで入手できます。BEA のホームページで [製品のドキュメント] をクリックするか、または WebLogic Server 製品ドキュメント ページを直接表示してください。

## このマニュアルの印刷方法

Web ブラウザの [ファイル | 印刷] オプションを使用すると、Web ブラウザからこのマニュアルを一度に 1 章ずつ印刷できます。

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

---

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

## 関連情報

BEA の Web サイトでは、WebLogic Server および Tuxedo のすべてのドキュメントを提供しています。

Java および Java CORBA アプリケーションの詳細については、以下を参照してください。

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

## 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](mailto: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](http://www.bea.com)) を通じて BEA カスタマサポートまでお問い合わせください。カスタマサポートへの連絡方法については、製品パッケージに同梱されているカスタマサポートカードにも記載されています。

カスタマサポートでは以下の情報をお尋ねしますので、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。
<i>斜体</i>	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>
<i>斜体の等幅テキスト</i>	コード内の変数を示す。 例： <pre>String <i>CustomerName</i>;</pre>

表記法	適用
すべて大文字のテキスト	デバイス名、環境変数、および論理演算子を示す。 例： LPT1 BEA_HOME OR
{ }	構文の中で複数の選択肢を示す。
[ ]	構文の中で任意指定の項目を示す。 例： <pre>java utils.MulticastTest -n name -a address [-p portnumber] [-t timeout] [-s send]</pre>
	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"> <li>■ 引数を複数回繰り返すことができる。</li> <li>■ 任意指定の引数が省略されている。</li> <li>■ パラメータや値などの情報を追加入力できる。</li> </ul>
.	コード サンプルまたは構文で項目が省略されていることを示す。 . .



---

# 1 WebLogic Tuxedo Connector の概要

以下の節では、WebLogic Server リリース 7.0 向けの WebLogic Tuxedo Connector の概念と機能について説明します。

- WebLogic Tuxedo Connector の概要
- 主要な機能と管理機能確認済みの制限
- WebLogic Tuxedo Connector と Jolt の相違点
- マニュアル
- プラットフォームのサポート
- ライセンス
- WebLogic Tuxedo Connector 6.x アプリケーションのアップグレード

## WebLogic Tuxedo Connector の概要

WebLogic Tuxedo Connector は、WebLogic Server アプリケーションと Tuxedo サービス間の相互運用性を提供します。コネクタでは、WebLogic Server クライアントが Tuxedo サービスを呼び出し、Tuxedo クライアントがサービス要求に応じて WebLogic Server エンタープライズ JavaBean (EJB) を呼び出すことができます。

## 主要な機能と管理機能

WebLogic Tuxedo Connector は、Tuxedo ATMI とほぼ同じ Java Application-to-Transaction Monitor Interface (JATMI) を使用することで、WebLogic Server と Tuxedo を相互運用するアプリケーションの開発とサポートを可能にします。WebLogic Tuxedo Connector の tBridge 機能には、高度な Tuxedo /Q および JMS メッセージサービスが用意されています。

WebLogic Tuxedo Connector によって、以下のような双方向の相互運用性が実現します。

- Tuxedo アプリケーションからの WebLogic Server アプリケーションの呼び出し、およびその逆、または Tuxedo アプリケーションからの EJB の呼び出し
- 既存の Tuxedo 環境への WebLogic Server アプリケーションの統合
- トランザクション サポート
- CORBA Java と CORBA C++ サーバ アプリケーション間の相互運用性を提供する機能
- WebLogic Process Integrator を使用した、eLink 1.2 アダプタなどの Tuxedo ATMI サービス全体にわたるワークフローの管理
- WebLogic Server と Tuxedo の間の複数接続の定義

WebLogic Tuxedo Connector の主要な管理機能を次に示します。

- 簡単な実装。WebLogic Tuxedo Connector では、既存の Tuxedo アプリケーション コードを修正する必要がありません。
  - 既存の Tuxedo クライアントは、WebLogic Tuxedo Connector を使用して WebLogic Server EJB を呼び出します。
  - 新規または修正された WebLogic Server クライアントは、WebLogic Tuxedo Connector を使用して Tuxedo サービスを呼び出します。
- ドメインおよび ACL セキュリティを含む、双方向のセキュリティ伝播。
- ドメインレベルのフェイルオーバおよびフォールバック。
- Tuxedo /Q および JMS が提供する高度なメッセージング サービス。



- eLink を使用した、メインフレームと他の従来のアプリケーションとの相互運用。

## 確認済みの制限

WebLogic Tuxedo Connector には以下の制限があります。

- **WebLogic Tuxedo Connector** ゲートウェイに対するコンフィグレーションの動的な変更がサポートされていません。
- クラスタ環境では、着信 RMI/IIOP または発信 CORBA がサポートされていません。
- VMS および OS/390 プラットフォーム上で動作する Tuxedo 6.5 はサポートされていません。
- IBM AIX プラットフォーム上で動作する **WebLogic Server** からの発信 CORBA アプリケーションはサポートされていません。**WebLogic Server** では、ソケットの作成に、**IBM JDK** と互換性のない **Sun** 固有の拡張を使用します。

## WebLogic Tuxedo Connector と Jolt の相違点

WebLogic Tuxedo Connector は、Jolt の代替機能ではありません。WebLogic Tuxedo Connector は、次の点で Jolt と異なっています。

- **WebLogic Tuxedo Connector** は、Jolt と似てはいるが異なる API を提供します。
- Jolt は汎用的な Java クライアントおよび他の Web サーバアプリケーションの開発が可能ですが、**WebLogic Tuxedo Connector** にはできません。
- Jolt は、統合された **WebLogic Server-Tuxedo** トランザクションのメカニズムを提供しません。

汎用的な Java クライアントまたは他の Web サーバ アプリケーションが必要で、WebLogic Server がソリューションの一部ではない場合は、ソリューションとして WebLogic Tuxedo Connector ではなく Jolt を使用してください。

# マニュアル

WebLogic Tuxedo Connector のマニュアルは、次の場所からダウンロードできません。

- BEA の Web サイト。BEA のホーム ページ (<http://www.beasys.co.jp>) で [製品 のドキュメント] をクリックしてください。
- WebLogic Server の「e-docs」製品マニュアル ページ (<http://edocs.beasys.co.jp/e-docs/>) を直接参照してください。WebLogic Server 7.0 マニュアル センターのリンクに従ってください。

# プラットフォームのサポート

プラットフォームのサポートに関する最も正確な最新情報については、「動作確認状況」 ページを参照してください。

# ライセンス

**注意：** WebLogic Server のライセンスの詳細については、「WebLogic Server ライセンスのインストール」を参照してください。

この節では、WebLogic Tuxedo Connector のライセンス情報について説明しません。

- 暗号を使用しない場合は、コネクタを使用するためのライセンス要件はありません。

- 暗号を使用する場合は、適切な Tuxedo LLE ライセンスおよび適切な *WebLogic Server* ライセンスが必要です。

## WebLogic Tuxedo Connector 6.x アプリケーションのアップグレード

WebLogic Server 7.0 で使用する場合、WebLogic Tuxedo Connector 6.x アプリケーション (WebLogic Tuxedo Connector 1.0 も含めて) の変更が何箇所か必要になります。WebLogic Server 7.0 の WebLogic Tuxedo Connector へのアップグレードに必要な管理とプログラミングの変更点の詳細については、「WebLogic Server 6.x からバージョン 7.0 へのアップグレード」を参照してください。



---

## 2 WebLogic Tuxedo Connector の コンフィグレーション

以下の節では、WebLogic Tuxedo Connector のコンフィグレーションの方法について説明します。

- 環境の変更と考慮事項の概要
- アプリケーション用の WebLogic Tuxedo Connector のコンフィグレーション

### 環境の変更と考慮事項の概要

この節では、WebLogic Tuxedo Connector を使用する前に、Tuxedo および WebLogic Server 環境に行う必要のある変更の概要を示します。

### Tuxedo の変更

**注意：** Tuxedo ドメインの詳細については、「Domains について」ページを参照してください。

Tuxedo ユーザは、次のような環境の変更を行う必要があります。

- 既存の Tuxedo アプリケーションがすでに Tuxedo /T DOMAINS を使用している場合、WebLogic Tuxedo Connector インスタンス化への接続ごとに、ドメイン コンフィグレーション ファイルへ新しいドメインを追加する必要があります。
- 既存の Tuxedo アプリケーションがドメインを使用していない場合、アプリケーションの TUXCONFIG にドメイン サーバを追加する必要があります。WebLogic Tuxedo Connector インスタンス化に対応する Tuxedo /T Domain エントリを使用して、新しい DMCONFIG を作成する必要があります。

- WebLogic Tuxedo Connector は、Tuxedo ドメインで常にエンコーディングが有効になっていることを要求します。DMCONFIG ファイルで MTYPE を設定しないか、NULL に設定する必要があります。

## WebLogic Server の変更

以下の節では、WebLogic Tuxedo Connector を使用するために必要な WebLogic Server の変更について説明します。

- 管理とプログラミング
- WebLogic Server のスレッド

### 管理とプログラミング

WebLogic Server ユーザは、次のような環境の変更を行う必要があります。

- Java クライアントまたはサーバを作成する。WebLogic Tuxedo Connector クライアントまたはサーバの作成の詳細については、『WebLogic Tuxedo Connector プログラマーズ ガイド』を参照してください。
- WebLogic Server コンソールまたはコマンドライン インタフェースを使って、WebLogic Tuxedo Connector をコンフィグレーションします。WebLogic Tuxedo Connector のコンフィグレーション方法の詳細については、2-3 ページの「アプリケーション用の WebLogic Tuxedo Connector のコンフィグレーション」を参照してください。
- [Local WLS Domain] の [ACL ポリシー] が「Local」に設定されている場合、Tuxedo リモート ドメイン DOMAINID はローカル ユーザとして、認証されなければなりません。詳細については、3-11 ページの「ユーザ認証」を参照してください。

### WebLogic Server のスレッド

**注意：** WebLogic Server のパフォーマンスおよびチューニングの詳細については、『BEA WebLogic Server パフォーマンス チューニング ガイド』を参照してください。

ゲートウェイからサービスをディスパッチするときに使用できるクライアントスレッドの数によって、同時に実行できるサービスの数が制限されることがあります。このリリースの **WebLogic Tuxedo Connector** では、利用可能なスレッドの数を増やすための **WebLogic Tuxedo Connector** 属性はありません。サービス **EJB** を呼び出すときは、適切なスレッドモデルを使用します。場合によっては、利用可能な **WebLogic Server** スレッドの数を大きな値に増やす必要があります。

# アプリケーション用の **WebLogic Tuxedo Connector** のコンフィグレーション

**注意：** このリリースの **WebLogic Tuxedo Connector** では、静的なコンフィグレーションのみを提供します。**WebLogic Tuxedo Connector** のコンフィグレーションに使われたパラメータを変更しなければならない場合は、**WebLogic Server** を再起動してその変更を有効にする必要があります。たとえば、ドメイン ネットワーク リンクの追加や削除、ネットワーク アドレスの変更、および新しいサービスのインポートやエクスポートは実行できません。

この節では、**WebLogic Server** アプリケーションと **Tuxedo** アプリケーションの相互運用を可能にするために **WebLogic Tuxedo Connector** をコンフィグレーションする方法について説明します。

- **WebLogic Tuxedo Connector MBean** クラス
- **Administration Console** を使用した **WebLogic Tuxedo Connector** のコンフィグレーション
- コマンドライン インタフェースを使った **WebLogic Tuxedo Connector** のコンフィグレーション
- **WebLogic Server** 環境の設定
- **WebLogic Tuxedo Connector** プロパティの設定方法
- **WebLogic Tuxedo Connector** のコンフィグレーション ガイドライン

## WebLogic Tuxedo Connector MBean クラス

**注意：** MBean パラメータの詳細については、10-1 ページの「WebLogic Tuxedo Connector MBean 属性」を参照してください。

WebLogic Tuxedo Connector は、MBean を使用して、WebLogic Server と Tuxedo との間のサービス リクエストを処理するための接続情報とセキュリティ プロトコルを記述します。これらのコンフィグレーション パラメータは、Tuxedo ドメイン間の通信に必要な相互運用属性に似ています。コンフィグレーション パラメータは、WebLogic Server の config.xml ファイルに格納されます。以下の表に、WebLogic Tuxedo Connector のコンフィグレーションに使用される MBean のタイプを示します。

MBean のタイプ	説明
WTCTServerMBean	WebLogic Server と Tuxedo との間の接続に必要な相互運用属性を持つ親 MBean。
WTCLocalTuxDomMBean	他のドメインで認識されるローカル ドメインのビューを提供する。ローカル ドメインを少なくとも 1 つはコンフィグレーションする必要がある。
WTCRemoteTuxDomMBean	ローカル ドメインで認識されるリモート ドメインのビューを提供する。複数のリモート ドメインをコンフィグレーションしてもよい。
WTCEXportMBean	ローカル ドメインによってエクスポートされるサービス情報を提供する。
WTCImportMBean	インポートされたサービスおよびリモート ドメインで使用可能なサービスに関する情報を提供する。
WTCResourcesMBean	ドメインのグローバル フィールドテーブル クラス、ビュー テーブル クラスおよびアプリケーション パスワードを指定する。
WTCPasswordMBean	相互ドメイン認証にコンフィグレーション情報を指定する。



MBean のタイプ	説明
WTCtBridgeGlobalMBean	WebLogic Server と Tuxedo の間のメッセージ転送に関するグローバル コンフィグレーション情報を指定する。
WTCtBridgeRedirectMBean	WebLogic Server と Tuxedo の間のメッセージのソース、ターゲット、方向および転送を指定する。

## Administration Console を使用した WebLogic Tuxedo Connector のコンフィグレーション

Administration Console により、WebLogic Tuxedo Connector 接続のコンフィグレーション、管理、モニタを行うことができます。これらのタスクに使用するタブを表示するには、次の操作を行います。

1. Administration Console を起動します。
2. 左ペインで [ サービス ] ノードを選択し、[ WebLogic Tuxedo Connector ] ノードを展開します。
3. コンフィグレーションしたいコンポーネント固有のツリー内でノードを作成または変更します。
4. オンライン ヘルプの指示に従います。オンライン ヘルプへのリンクについては、表 2-1 を参照してください。

次の表では、接続タスクを一般的な実行順序で示します。この順序は変更してもかまいません。ただし、オブジェクトは関連付けおよび割り当ての前にコンフィグレーションする必要があります。

**表 2-1 WebLogic Tuxedo Connector コンフィグレーションタスク**

タスク番号	タスク	説明
1	WTCServer の作成	右ペインの [ 一般 ] タブで、[ 名前 ] および [ デプロイ順 ] の属性を設定する。

**表 2-1 WebLogic Tuxedo Connector コンフィグレーション タスク**

タスク番号	タスク	説明
2	ローカル WLS ドメインの作成	[一般]、[接続]、[セキュリティ]の各タブで、ローカルドメインを記述する属性を設定する。ローカルドメインを少なくとも1つはコンフィグレーションする必要がある。
3	リモート Tuxedo ドメインの作成	[一般]、[接続]、[セキュリティ]の各タブで、リモート Tuxedo ドメインを記述する属性を設定する。
4	エクスポートされたサービスの作成	[一般]タブで、エクスポートされた WebLogic Server サービスを記述する属性を設定する。
5	インポートされたサービスの作成	[一般]タブで、インポートされた Tuxedo サービスを記述する属性を設定する。
6	パスワードコンフィグレーションの作成	[接続]タブで、パスワードを記述する属性を設定する。
7	リソースの作成	[接続]タブで、WebLogic Tuxedo Connector のリソースを記述する属性を設定する。
8	tBridge 接続の作成	WebLogic Server と Tuxedo との間のメッセージ転送に関するグローバルコンフィグレーション情報を指定する。
9	tBridge リダイレクションの作成	WebLogic Server と Tuxedo の間のメッセージのソース、ターゲット、方向および転送の指定に使われる属性を設定する。
10	サーバに対する WTCServer の割り当て	WTCServer MBean の対象サーバを選択する。

# コマンドライン インタフェースを使った WebLogic Tuxedo Connector のコンフィグレーション

コマンドライン インタフェースは、WebLogic Tuxedo Connector 接続を作成し管理する方法を提供します。コマンドライン インタフェースの使い方については、「Mbean 管理コマンド リファレンス」を参照してください。

## WebLogic Server 環境の設定

WebLogic Server アプリケーションの環境は、`setEnv` スクリプトを実行して設定します。

- NT/2000 ユーザの場合は、`setEnv.cmd` を実行します。
- UNIX ユーザの場合は、`setEnv.sh` を実行します。

環境を初めて設定する場合は、スクリプトの設定をチェックする必要があります。必要に応じて、次の手順に従ってアプリケーション環境の設定を修正します。

1. コマンドラインで、WebLogic Server アプリケーションの場所にディレクトリを変更します。
2. `vi` などのテキスト エディタを使用して、`setEnv` スクリプトを編集します。
  - NT/2000 ユーザの場合は、`setEnv.cmd` を編集します。
  - UNIX ユーザの場合は、`setEnv.sh` を編集します。
3. ファイルを保存します。

**注意：** `setExamplesEnv` ファイルは、配布キットで提供されている WebLogic Server サンプルの環境を設定するために使用します。

# WebLogic Tuxedo Connector プロパティの設定方法

**注意：** WebLogic Server プロパティの設定に関する詳細は、「WebLogic Server の起動と停止」を参照してください。

TraceLevel、PasswordKey、および encoding は、WebLogic Server のプロパティです。これらのプロパティの設定が必要であれば、サーバ起動スクリプトの中の JAVA\_OPTIONS 変数を更新します。

例：

```
JAVA_OPTIONS=-Dweblogic.wtc.TraceLevel=100000
```

## TraceLevel の設定

**注意：** TraceLevel の詳細については、9-1 ページの「WebLogic Tuxedo Connector のモニタ」を参照してください。

TraceLevel を使って、メッセージのトレースレベルを指定し、それに従って WebLogic Tuxedo Connector がログファイルにトレースを送ります。

```
JAVA_OPTIONS= -Dweblogic.wtc.TraceLevel=tracelevel
```

ここで、*tracelevel* は、10,000 ~ 100,000 の WebLogic Tuxedo Connector のトレースレベルを指定する数字です。

## PasswordKey の設定

**注意：** PasswordKey の詳細については、3-8 ページの「WTCPassWord MBean のコンフィグレーション」を参照してください。

PasswordKey を使って、weblogic.wtc.gwt.genpassword ユーティリティがパスワードの暗号化に使用するキーを指定します。

```
JAVA_OPTIONS=-Dweblogic.wtc.PasswordKey=mykey
```

ここで、*mykey* はキー値です。

## encoding の設定

WebLogic Server アプリケーションと Tuxedo アプリケーションの間で非 ASCII (マルチバイト) 文字列を転送する場合、文字セットを転送できるよう WebLogic Tuxedo Connector をコンフィグレーションする必要があります。WebLogic Tuxedo Connector では、WebLogic Server のプロパティを使用して、WebLogic Tuxedo Connector サービスで指定したすべての Tuxedo リモート ドメインで使用されるエンコーディングを一致させます。複数のコーディングセットを同時に実行する必要がある場合、WebLogic Server インスタンスごとに WebLogic Tuxedo Connector サービスを実行する必要があります。

文字セット転送を有効にするには、サーバ起動スクリプトにある `JAVA_OPTIONS` 変数を更新します。

例：

```
JAVA_OPTIONS=-Dweblogic.wtc.encoding=codesetname
```

ここで、*codesetname* は、リモート Tuxedo ドメインでサポートされており、使用されるコードセットです。サポートされている基本コーディングセットおよび拡張コーディングセットについては、「Supported Encodings」を参照してください。

リモート ドメインで使用されるエンコーディングと一致する正確なエンコーディング名を選択することができない場合があります。このような場合、リモート ドメインと同等のエンコーディング名を選択する必要があります。

例：

- サポートされているエンコーディングのリストには `EUC_JP` がある。
- リモート ドメインでは **Solaris** オペレーティング システムが動作し、`eucJP` がサポートされている。

名前は完全には一致していませんが、`EUC_JP` と `eucJP` はエンコーディングセットとして等価なので、WebLogic Server とリモート ドメイン間で文字列転送が正しく行われます。したがって、この場合は `EUC_JP` のエンコーディング プロパティを次のように設定します。

```
JAVA_OPTIONS=-Dweblogic.wtc.encoding=EUC_JP
```

## WebLogic Tuxedo Connector のコンフィグレーション ガイドライン

WebLogic Tuxedo Connector のコンフィグレーションを行う際には、以下のガイドラインに従ってください。

- ローカル TDOM は、コンフィグレーションのなかに少なくとも 1 つは入れなければなりません。
- config.xml ファイル内に、1 つ以上の WTCServer MBeans を作成してもかまいません。
- 2 つ以上の WTCServer MBeans を同じサーバに対してターゲットにしてはいけません。1 つのサーバは、1 つの WTCServer MBean のターゲットにしかできません。
- 対象サーバが選択された後で、WTCServer MBean に実装されたコンフィグレーションの変更はすべて、その対象サーバインスタンス内では更新されません。WTCServer MBean をサーバから削除してから、更新済みの WTCServer Mbean を対象サーバに追加する必要があります。対象サーバの選択の詳細については、「サーバへの WTC サーバの割り当て」を参照してください。

---

# 3 WebLogic Tuxedo Connector の管理

**注意：** WebLogic Tuxedo Connector MBean 属性に関する詳細なリファレンス情報については、第 10 章「WebLogic Tuxedo Connector MBean 属性」を参照してください。

以下の節では、接続の確立方法および WebLogic Server と Tuxedo 環境のドメイン間にセキュリティを提供する方法について説明します。WebLogic Tuxedo Connector MBean 属性は、Tuxedo ドメイン間の通信に必要な相互運用属性に類似したコンフィグレーションパラメータで構成されます。

以下の節では、WebLogic Tuxedo Connector のコンフィグレーション情報について説明します。

- ドメイン間の通信のコンフィグレーション
- ConnectionPolicy の動的ステータスへの影響
- ドメインレベルのフェイルオーバーとフェイルバックのコンフィグレーション
- リモートドメインの認証
- ユーザ認証
- Tuxedo と WebLogic Server の間でセキュリティを提供するように WebLogic Tuxedo Connector をコンフィグレーションする方法
- Simpapp サンプルおよび Simpserv サンプルの ACL ポリシー例
- リンクレベルの暗号化

## ドメイン間の通信のコンフィグレーション

**注意：** 動的ステータスの詳細については、3-4 ページの「ConnectionPolicy の動的ステータスへの影響」を参照してください。

ローカルドメイン ゲートウェイがリモートドメインとの接続を確立する条件を指定できるオプションがいくつかあります。これらの条件は、WTCLocalTuxDom MBean および WTCRemoteTuxDom MBean の ConnectionPolicy 属性を使用して指定します。次の接続ポリシーのいずれかを選択できます。

- 起動時に接続を要求する方法 (ON\_STARTUP)
- クライアント要求による接続のリクエスト方法 (ON\_DEMAND)
- 受信時接続の受け付け (INCOMING\_ONLY)
- LOCAL 接続ポリシーの使用方法

ON\_STARTUP および INCOMING\_ONLY の接続ポリシーでは、動的ステータスが呼び出されます。動的ステータスは、各リモートドメインに関連してインポートされたサービスのステータスをチェックして、レポートします。

### 起動時に接続を要求する方法 (ON\_STARTUP)

ON\_STARTUP のポリシーは、ゲートウェイサーバの初期化時に、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を使用して接続を確立することを示します。この接続ポリシーは、RetryInterval パラメータと MaxRetries パラメータで指定された一定の間隔で、エラーとなった接続を再試行します。起動時に接続を要求するには、WTCLocalTuxDom MBean 接続ポリシーを ON\_STARTUP に設定します。

### RetryInterval のコンフィグレーション方法

自動接続の試行の頻度は、接続を再試行する前にゲートウェイが待機する間隔(秒)を指定することによって制御できます。最小値は 0、デフォルト値は 60、そして最大値は 2147483647 です。



## MaxRetries のコンフィグレーション方法

**注意：** ConnectionPolicy が ON\_STARTUP に設定されているときだけ使用してください。他の接続ポリシーの場合、再試行処理は無効になります。

MaxRetries パラメータに値を割り当てることによって、ドメイン ゲートウェイが終了するまでにリモートドメイン アクセス ポイントへの接続を試行する回数を指定します。最小値は 0、デフォルトおよび最大値は 2147483647 です。

- MaxRetries を 0 に設定すると、自動接続の試行処理はオフになります。サーバは、リモート ゲートウェイに自動接続しません。
- MaxRetries に数値を設定すると、ゲートウェイは指定された回数だけ接続を再試行してから終了します。
- MaxRetries を 2147483647 に設定すると、再試行処理は無期限でまたは接続が確立するまで繰り返されます。

表 3-1 MaxRetries パラメータおよび RetryInterval パラメータの設定例

設定	処理
ConnectionPolicy:ON_STARTUP RetryInterval : 30 MaxRetries : 3	ゲートウェイは、接続の確立を 30 秒間隔で 3 回試行してから終了する。
ConnectionPolicy:ON_STARTUP MaxRetries : 0	ゲートウェイは初期化時に接続の確立を試行するが、最初の試行がエラーとなった場合は再試行しない。
ConnectionPolicy:ON_STARTUP RetryInterval : 30	ゲートウェイは、接続が確立されるまで、30 秒おきに接続の確立を試行する。

## クライアント要求による接続のリクエスト方法 (ON\_DEMAND)

**注意：** ConnectionPolicy を指定しない場合、WebLogic Tuxedo Connector は ON\_DEMAND の ConnectionPolicy を使用します。

ON\_DEMAND の接続ポリシーは、リモート サービスに対するクライアント 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要求されたときのみ、接続が試行されることを示します。

## 受信時接続の受け付け (INCOMING\_ONLY)

INCOMING\_ONLY の接続ポリシーは、ドメイン ゲートウェイが起動時にリモート ドメインへの接続を確立しないことを示します。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時接続に使用可能で、リモート サービスはドメイン ゲートウェイが受信時接続を受け付けたときに通知されます。

## LOCAL 接続ポリシーの使用方法

**注意：** LOCAL の ConnectionPolicy は、ローカルドメインでは無効です。

LOCAL の接続ポリシーは、リモート ドメインの接続ポリシーが明示的にローカルドメインの ConnectionPolicy 属性値にデフォルト設定されることを示します。リモート ドメインの ConnectionPolicy が定義されていない場合、システムは LocalAccessPoint を使用して、関連するローカルドメインで指定されている設定を使用します。

## ConnectionPolicy の動的ステータスへの影響

動的ステータスは、リモート サービスの可用性を決定します。WebLogic Tuxedo Connector コンフィグレーション ファイルで使用される接続ポリシーは、動的ステータス機能がサービスで使用可能かどうかを決定します。次の表は、ConnectionPolicy の動的ステータス機能に対する影響を示します。

ON_STARTUP	動的ステータスがオンになる。リモートドメインからインポートされたサービスは、そのリモートドメインへの接続が存在する間、通知される。
ON_DEMAND	動的ステータスがオフになる。リモートドメインからインポートされたサービスは、常に通知される。
INCOMING_ONLY	動的ステータスがオンになる。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、リモートドメインから受信時接続に対して使用可能である。リモートサービスは、ローカルドメインゲートウェイが、受信時接続を受け付けたときに通知される。

## ドメインレベルのフェイルオーバーとフェイルバックのコンフィグレーション

**注意：** Tuxedo T/Domain では、バックアップ リモートドメインは2つに制限されています。WebLogic Tuxedo Connector には、サーバにコンフィグレーションできるバックアップドメインの数に制限はありません。

WebLogic Tuxedo Connector が提供するドメインレベルのフェイルオーバーは、プライマリ リモートドメインでエラーが発生したとき、代替リモートドメインにリクエストを転送するメカニズムです。このフェイルオーバーは、ドメインが復元されると、プライマリ リモートドメインにフェイルバックします。

このレベルのフェイルオーバーおよびフェイルバックは、動的ステータスに依存します。ドメインレベルのフェイルオーバーおよびフェイルバックを有効にするには、ドメインを ON\_STARTUP または INCOMING\_ONLY の CONNECTION\_POLICY でコンフィグレーションする必要があります。

ドメインレベルのフェイルオーバーおよびフェイルバックは、リモートドメインへのネットワーク接続がある場合はリモートドメインを使用可能として定義し、リモートドメインへのネットワーク接続がない場合は使用不可能として定義します。

## ドメインレベルのフェイルオーバーおよびフェイルバックを使用するための要件

ドメインレベルのフェイルバックを使用するには、`CONNECTION_POLICY` パラメータの値に `ON_STARTUP` または `INCOMING_ONLY` を指定する必要があります。

`ON_DEMAND` の接続ポリシーは、リモートドメインが常に使用可能であることを想定して動作するため、ドメインレベルのフェイルバックには適しません。接続ポリシーとして `ON_STARTUP` または `INCOMING_ONLY` を指定しないと、サーバは Tuxedo の `RDOM` パラメータを使用して指定した代替リモートドメインにフェイルオーバーできません。

**注意：** リモートドメインはそれに対するネットワーク接続があれば「使用可能」であり、それに対するネットワーク接続がなければ「使用不可能」です。

## フェイルオーバーをサポートするためのドメインのコンフィグレーション

フェイルオーバーをサポートするには、特定のサービスを実行する責任を持つリモートドメインを指定する必要があります。 `WTCServer MBean` で、以下の項目を指定する必要があります。

- 各リモートドメインに対して `WTCRemoteTuxDom MBean` を作成する。
- 各リモートドメインが提供するサービスを指定する `WTCImport MBean` を作成する。

`TOUPPER` というサービスが、`TDOM1` と `TDOM3` という2つのリモートドメインから使用可能であるとします。使用する `WTCServer MBean` に、`WTCRemoteTuxDom MBean` が2つ、`WTCImport MBean` が2つ含まれています。使用する `config.xml` ファイル内の `WTCServer MBean` の内容は次のとおりです。

```
<WTCServer Name="WTCsimpapp"
  <WTCExport EJBNName="tuxedo.services.TOLOWERHome"
    LocalAccessPoint="TDOM2" Name="myExportedResources"
    ResourceName="TOLOWER"/>
```

```
<WTCImport LocalAccessPoint="TDOM2" Name="myImportedResources"
  RemoteAccessPointList="TDOM1" ResourceName="TOUPPER"/>
<WTCImport LocalAccessPoint="TDOM2" Name="2ndImportedResources"
  RemoteAccessPointList="TDOM3" ResourceName="TOUPPER"/>
<WTCLocalTuxDom AccessPoint="TDOM2" AccessPointId="TDOM2"
  ConnectionPolicy="ON_DEMAND" Interoperate="no"
  NWAddr="//123.123.123.123:5678" Name="myLoclTuxDom" Security="NONE"/>
<WTCRemoteTuxDom AccessPoint="TDOM1" AccessPointId="TDOM1"
  LocalAccessPoint="TDOM2" NWAddr="//123.123.123.123:1234"
  Name="myRTuxDom"/>
<WTCRemoteTuxDom AccessPoint="TDOM3" AccessPointId="TDOM3"
  LocalAccessPoint="TDOM2" NWAddr="//234.234.234.234:5555"
  Name="2ndRemoteTuxDom"/>
</WTCServer>
```

## フェイルバックをサポートするためのドメインのコンフィグレーション

フェイルバックは、プライマリ リモート ドメインへのネットワーク接続が、次の理由で再確立された場合に発生します。

- 自動再試行 (ON\_STARTUP のみ)
- 受信時接続

## リモート ドメインの認証

**注意：** Tuxedo 6.5 ユーザは、Interoperate パラメータを Yes に設定します。

ドメイン ゲートウェイは、リモート ドメインから要求された受信時接続、およびローカルドメインから要求された送信時接続を認証するように設定できます。アプリケーション管理者は、リモートドメインからの受信時接続のセキュリティを強化する必要がある場合を定義できます。特定のローカルドメインによって使用されるセキュリティのレベルを指定するには、WTCServer MBean の WTCLocalTuxDom MBean で SECURITY パラメータを設定します。パスワードセキュリティには次のような3つのレベルがあります。

- 「セキュリティなし」(NONE オプションを使用) – リモート ドメインからの受信時接続は認証されません。
- 「アプリケーション パスワード」(APP\_PW オプションを使用) – リモート ドメインからの受信時接続は、WTCServer MBean の WTCPassword MBean で定義されたアプリケーション パスワードを使用して認証されます。暗号化されたアプリケーション パスワードを作成するには、`weblogic.wtc.gwt.genpasswd` ユーティリティを使用します。
- 「リモート ドメイン パスワード」(DM\_PW オプションを使用) – この機能は 2 つ以上のドメイン間のセキュリティを強化します。ローカルドメインとリモートドメイン間の接続は、WTCServer MBean の WTCPassword MBean で定義されたパスワードのペアを使用して認証されます。暗号化されたローカルおよびリモートのドメイン パスワードを作成するには、`weblogic.wtc.gwt.genpasswd` ユーティリティを使用します。

WTCServer MBean の WTCLocalTuxDom MBean の SECURITY パラメータは、Tuxedo ドメイン コンフィグレーション ファイルの \*DM\_LOCAL\_DOMAINS セクションの SECURITY パラメータと一致していなければなりません。

- 認証が必要な場合、ローカルドメインとリモートドメイン間で接続が確立されるたびに認証が行われます。
- WTCLocalTuxDom MBean のセキュリティタイプが \*DM\_LOCAL\_DOMAINS のセキュリティタイプと一致しない場合、またはパスワードが一致しない場合、接続は失敗します。

## WTCPassword MBean のコンフィグレーション

**注意：** PasswordKey の割り当て方法については、2-8 ページの「WebLogic Tuxedo Connector プロパティの設定方法」を参照してください。

LocalPassword、RemotePassword、および AppPassword 要素の暗号パスワードを作成するには、`weblogic.wtc.gwt.genpasswd` を使用します。このユーティリティは、キーを使用して WTCServer MBean の WTCPassword MBean にコピーされるパスワードを暗号化します。

- WTCPassword MBean は、クリア テキストのパスワードを格納しません。

- キー値は、WebLogic Server プロパティです。
  - PasswordKey 属性を使用してキーを割り当てます。

```
-Dweblogic.wtc.PasswordKey=mykey
```

ここで、**mykey** はキー値です。
  - PasswordKey 属性は、1つのキー値にのみ割り当てられます。このキー値は、指定された WebLogic Server で使用するため、生成されるすべての WebLogic Tuxedo Connector (ローカル、リモート、アプリケーション) パスワードに対して使用されます。

## 解説

引数なしでユーティリティを呼び出し、コマンドライン オプションを表示します。

例：

```
$ java weblogic.wtc.gwt.genpasswd
Usage:genpasswd Key <LocalPassword|RemotePassword|AppPassword>
<local|remote|application>
```

ユーティリティは、エンコードされたパスワードおよびパスワード IV を含む XML 要素を返します。この結果を切り取り、WTCServer MBean 内の適切なフィールドに貼り付けます。

```
<LocalPassword IV="my_passwordIV">my_password</LocalPassword>
```

このとき、次のようにします。

- `my_passwordIV` で表された文字列を切り取り、`PasswordIV` フィールドに貼り付けます。引用符 (") は含めません。
- `my_password` で表された文字列を切り取り、`Password` フィールドに貼り付けます。「>」や「<」は含めません。

## 例

この節では、各パスワード要素タイプの例を紹介します。

### LocalPasswords

次の例では、ローカルドメインのパスワードとして「LocalPassword1」を暗号化するために *key1* を使用します。

```
$ java weblogic.wtc.gwt.genpasswd Key1 LocalPassword1 local  
<LocalPassword IV="I#^Da0efo1">!djK*87$klbJJ</LocalPassword>
```

使用する Password MBean 属性は、次のとおりです。

```
Local Password IV:I#^Da0efo1
```

```
Local Password:!djK*87$klbJJ
```

### RemotePasswords

次の例では、リモートドメインのパスワードとして「RemotePassword1」を暗号化するために *mykey* を使用します。

```
$ java weblogic.wtc.gwt.genpasswd mykey RemotePassword1 remote  
<RemotePassword IV="Rq$45%%kK">McFrd3#f41K1</RemotePassword>
```

使用する Password MBean 属性は、次のとおりです。

```
Remote Password IV:Rq$45%%kK
```

```
Remote Password:McFrd3#f41K1
```

### AppPasswords

次の例では、アプリケーションパスワードとして「test123」を暗号化するために *key1* を使用します。

```
$ weblogic.wtc.gwt.genpasswd mykey test123 application  
<AppPassword IV="gx8aSkAgLFg=">c98Y/P94HY3rCAVmkF=</AppPassword>
```



使用する Resources MBean 属性は、次のとおりです。

Application Password:gx8aSkAgLFg=

Application Password IV:c98Y/P94HY3rCAVmkF=

## ユーザ認証

アクセス制御リスト (ACL) は、サービスを実行できるリモート Tuxedo ドメインを制限することによって、ローカル WLS ドメインの内部にあるローカルサービスへのアクセスを制限します。リモート Tuxedo ドメインからの着信ポリシーは、AclPolicy 要素を使用して指定します。リモート Tuxedo ドメインへの発信ポリシーは、CredentialPolicy 要素を使用して指定します。これによって、WebLogic Server と Tuxedo アプリケーションは同じセットのユーザを共有でき、ユーザはあるシステムから別のシステムへユーザの資格を伝播できます。

AclPolicy および CredentialPolicy の有効な値は次のとおりです。

- LOCAL
- GLOBAL

## ACL ポリシーは LOCAL

WebLogic Tuxedo Connector の ACL ポリシーが「Local」に設定されている場合、ローカルサービスへのアクセスは、CredentialPolicy には依存しません。Tuxedo リモートドメインの DOMAINID は、ローカルな WebLogic Server ユーザとして認証されます。WebLogic Tuxedo Connector で DOMAINID がローカルユーザとして認証されるようにするには、WebLogic Server コンソールを使用して、以下の手順を実行します。

1. [セキュリティ] ノードをクリックします。
2. [レルム] をクリックします。
3. デフォルトのセキュリティレルムを選択します。

4. [ ユーザ ] をクリックします。
5. [ 新しい User のコンフィグレーション ] テキスト リンクをクリックします。
6. [ DefaultAuthenticator ] をクリックします。
7. [ 一般 ] タブで、次の作業を行います。
  - a. [ 名前 ] フィールドに **Tuxedo** の DOMAINID を追加します。
  - b. 有効なパスワードを入力します。
  - c. [ 適用 ] をクリックします。

## ACL ポリシーは Global

WebLogic Tuxedo Connector の ACL ポリシーが「GLOBAL」になっている場合、ローカル サービスへのアクセスは、CredentialPolicy によって異なります。

## CredentialPolicy は Global

CredentialPolicy が GLOBAL に設定された状態でリモート ドメインが実行されている場合、リクエストには呼び出し側の資格があります。

## CredentialPolicy は Local

CredentialPolicy が LOCAL に設定された状態でリモート ドメインが実行されている場合、その結果は呼び出しを発行したユーザ コンフィグレーションによって異なります。

## Tuxedo 6.5 のユーザ認証

Tuxedo 6.5 ユーザは、Interoperate パラメータを Yes に設定します。AclPolicy 要素と CredentialPolicy 要素は無視され、Tuxedo リモートドメイン DOMAINID は、ローカル WebLogic Server ユーザとして認証されます。ユーザセキュリティ機能が必要で、かつ WebLogic Tuxedo Connector を使用する場合は、Tuxedo 7.1 以上にアップグレードする必要があります。

# Tuxedo と WebLogic Server の間でセキュリティを提供するように WebLogic Tuxedo Connector をコンフィグレーションする方法

Tuxedo アプリケーションと WebLogic Server アプリケーションの間でセキュリティを提供するように WebLogic Tuxedo Connector をコンフィグレーションするには、以下の手順を行います。

- WTCLocalTuxDom MBean のコンフィグレーション
- WTCRemoteTuxDom MBean のコンフィグレーション

## WTCLocalTuxDom MBean のコンフィグレーション

WTCServer MBean の WTCLocalTuxDom MBean の SECURITY パラメータを、Tuxedo ドメイン コンフィグレーション ファイルの \*DM\_LOCAL\_DOMAINS セクションの SECURITY パラメータと一致するように設定します。

## WTCLocalTuxDom MBean のコンフィグレーション例

```
<WTCLocalTuxDom AccessPoint="TDOM2" AccessPointId="TDOM2"  
  ConnectionPolicy="ON_DEMAND" Interoperate="no"  
  NWAddr="//123.123.123.123:5678" Name="myLocalTuxDom"  
  Security="DM_PW"/>
```

## Tuxedo \*DM\_LOCAL\_DOMAINS のコンフィグレーション例

```
.  
.  
.  
*DM_LOCAL_DOMAINS  
  domain1      DOMAINID = "domain1"  
  GWGRP = "GWGRP"  
  CONNECTION_POLICY=ON_DEMAND  
  DMTLOGDEV = "/nfs/home1/dkumar/lcsol5/tmp.100/DMTLOG"  
  DMTLOGNAME = "DMTLG1"  
  SECURITY=DM_PW  
  BLOCKTIME=10  
.  
.  
.
```

## WTCRemoteTuxDom MBean のコンフィグレーション

WTCRemoteTuxDom MBean をコンフィグレーションすると、着信および発信のアクセス制御リスト (ACL) ポリシーを確立できます。

WebLogic Server 環境を準備するには、次の手順を実行します。

1. WTCServer MBean の WTCRemoteTuxDom MBean の AclPolicy 属性を GLOBAL に設定します。
2. WTCServer MBean の WTCRemoteTuxDom MBean の CredentialPolicy 属性を GLOBAL に設定します。
3. CredentialPolicy が GLOBAL に設定する場合、WebLogic Server 環境で Tuxedo tpusr ファイルのコピーが必要になります。環境に TpUserFile を追加するには、以下の手順を行います。

- a. TUXEDO から WebLogic Server アプリケーション環境に `tpusr` ファイルをコピーするか、独自の `tpusr` ファイルを生成します。
- b. `WTCServer MBean` の `WTCRemoteTuxDom MBean` の `TpUserFile` 属性に、`tpusr` ファイルの絶対パスとその名前を設定します。

**注意：** Tuxedo `tpusr` ファイル作成の詳細については、「ユーザ・レベルの認証によるセキュリティを有効にする方法」を参照してください。

### WTCRemoteTuxDom MBean の例

```
<WTCRemoteTuxDom AccessPoint="TDOM3" AccessPointId="TDOM3"
  AclPolicy="GLOBAL" CredentialPolicy="GLOBAL"
  LocalAccessPoint="TDOM2" NWAddr="//234.234.234.234:5555"
  Name="WTCRemoteTuxDom-1012347174947"
  TpUsrFile="C:\runs\tmp.100\config\wldom1\tpusr" />
```

### Tuxedo \*DM\_LOCAL\_DOMAINS のコンフィグレーションの例

```
.
.
.
*DM_REMOTE_DOMAINS
  wldom1 DOMAINID = "wldom1"
  ACCESSPOINTID="wldom1"
  ACL_POLICY="GLOBAL"
  CREDENTIAL_POLICY="GLOBAL"
.
.
```

## Simpapp サンプルおよび Simpserv サンプルの ACL ポリシー例

この節では、`simpapp` および `simpserv` の例を使用した、ACL 制御の設定方法の例を示します。

- John と Bob だけが Toupper へのアクセスを持ちます。
- Dan と John だけが Tolower へのアクセスを持ちます。
- Toupper は、リモート Tuxedo サービス TOUPPER にアクセスするために使用します。
- Tolower は、リモート Tuxedo ユーザに実際のサービスを提供します。

ACL 制御を確立するには、次の手順に従います。

1. WebLogic Server Console を使用して、ユーザ John、Bob、および Dan を WebLogic Security に追加します。
2. Tuxedo TOUPPER サービスの security-role および method\_permission 要素を追加するために、ejb-jar.xml を修正します。太字の部分は、セキュリティ実装をサポートするための変更箇所を示します。

#### コード リスト 3-1 TOUPPER ejb-jar.xml セキュリティ サンプル コード

---

```
<?xml version="1.0"?>
<!--
Copyright (c) 2000 BEA Systems, Inc. All rights reserved

THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF BEA Systems, Inc. The
copyright notice above does not evidence any actual or intended publication
of such source code.

-->

<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans
2.0//EN" "http://java.sun.com/j2ee/dtds/ejb-jar_2_0.dtd">

<ejb-jar>
  <enterprise-beans>
    <session>
      <ejb-name>Toupper</ejb-name>
      <home>weblogic.wtc.examples.simpapp.ToupperHome</home>
      <remote>weblogic.wtc.examples.simpapp.Toupper</remote>
      <ejb-class>weblogic.wtc.examples.simpapp.ToupperBean</ejb-class>
      <session-type>Stateful</session-type>
      <transaction-type>Container</transaction-type>
    </session>
  </enterprise-beans>
</assembly-descriptor>
```

```

<security-role>
  <role-name>dom2</role-name>
</security-role>
<method-permission>
  <role-name>dom2</role-name>
  <method>
    <ejb-name>Toupper</ejb-name>
    <method-name>Toupper</method-name>
  </method>
</method-permission>
<container-transaction>
  <method>
    <ejb-name>Toupper</ejb-name>
    <method-intf>Remote</method-intf>
    <method-name>*</method-name>
  </method>
  <trans-attribute>Supports</trans-attribute>
</container-transaction>
</assembly-descriptor>
</ejb-jar>

```

---

3. Tuxedo TOUPPER サービスの security-role-assignment 要素を追加するために、weblogic-`ejb-jar.xml` を修正します。太字の部分は、セキュリティ実装をサポートするための変更箇所を示します。

### コード リスト 3-2 TOUPPER Weblogic-`ejb-jar.xml` セキュリティ サンプル コード

---

```

<?xml version="1.0"?>
<!--
Copyright (c) 2000 BEA Systems, Inc. All rights reserved

THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF BEA Systems, Inc.
The copyright notice above does not evidence any actual or intended
publication of such source code.

-->
<!DOCTYPE weblogic-ejb-jar PUBLIC '-//BEA Systems, Inc.//DTD WebLogic 6.0.0
EJB//EN' 'http://www.bea.com/servers/wls600/dtd/weblogic-ejb-jar.dtd'>
<weblogic-ejb-jar>

```

```
<weblogic-enterprise-bean>
  <ejb-name>Toupper</ejb-name>
  <stateful-session-descriptor>
    <stateful-session-cache>
      <max-beans-in-cache>100</max-beans-in-cache>
    </stateful-session-cache>
  </stateful-session-descriptor>
  <jndi-name>tuxedo.services.ToupperHome</jndi-name>
</weblogic-enterprise-bean>
<security-role-assignment>
  <role-name>dom2</role-name>
  <principal-name>john</principal-name>
  <principal-name>bob</principal-name>
</security-role-assignment>
</weblogic-ejb-jar>
```

---

4. Tolower サービスの security-role および method-permission 要素を追加するため、ejb-jar.xml を修正します。太字の部分は、セキュリティ実装をサポートするための変更箇所を示します。

#### コード リスト 3-3 Tolower ejb-jar.xml セキュリティ サンプル コード

---

```
<?xml version="1.0"?>
<!--

Copyright (c) 2000 BEA Systems, Inc. All rights reserved
THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF BEA Systems, Inc.
The copyright notice above does not evidence any actual or intended
publication of such source code.

-->
<!DOCTYPE ejb-jar PUBLIC "-//Sun Microsystems, Inc.//DTD Enterprise JavaBeans
2.0//EN" 'http://java.sun.com/j2ee/dtds/ejb-jar_2_0.dtd'>

<ejb-jar>
  <enterprise-beans>
    <session>
      <ejb-name>Tolower</ejb-name>
      <home>weblogic.wtc.jatmi.TuxedoServiceHome</home>
      <remote>weblogic.wtc.jatmi.TuxedoService</remote>
      <ejb-class>weblogic.wtc.examples.simpsserv.TolowerBean</ejb-class>
      <session-type>Stateless</session-type>
      <transaction-type>Container</transaction-type>
    </session>
```



```
</enterprise-beans>

<assembly-descriptor>
  <security-role>
    <role-name>rdom2</role-name>
  </security-role>
  <method-permission>
    <role-name>rdom2</role-name>
    <method>
      <ejb-name>Tolower</ejb-name>
      <method-name>service</method-name>
    </method>
  </method-permission>
</container-transaction>
  <method>
    <ejb-name>Tolower</ejb-name>
    <method-intf>Remote</method-intf>
    <method-name>*</method-name>
  </method>
  <trans-attribute>Supports</trans-attribute>
</container-transaction>
</assembly-descriptor>
</ejb-jar>
```

---

5. Tolower サービスの security-role-assignment 要素を追加するために、Weblogic-ejb-jar.xml を修正します。太字の部分は、セキュリティ実装をサポートするための変更箇所を示します。

### コード リスト 3-4 Tolower Weblogic-ejb-jar.xml セキュリティ サンプル コード

---

```
<?xml version="1.0"?>
<!--

Copyright (c) 2000 BEA Systems, Inc. All rights reserved

THIS IS UNPUBLISHED PROPRIETARY SOURCE CODE OF BEA Systems, Inc.
The copyright notice above does not evidence any actual or intended
publication of such source code.

-->

<!DOCTYPE weblogic-ejb-jar PUBLIC "-//BEA Systems, Inc.//DTD WebLogic 6.0.0
EJB//EN" 'http://www.bea.com/servers/wls600/dtd/weblogic-ejb-jar.dtd'>
```

```
<weblogic-ejb-jar>
  <weblogic-enterprise-bean>
    <ejb-name>Tolower</ejb-name>
    <stateless-session-descriptor>
      <pool>
        <max-beans-in-free-pool>100</max-beans-in-free-pool>
      </pool>
    </stateless-session-descriptor>
    <jndi-name>tuxedo.services.TOLOWERHome</jndi-name>
  </weblogic-enterprise-bean>
  <security-role-assignment>
    <role-name>rdom2</role-name>
    <principal-name>john</principal-name>
    <principal-name>dan</principal-name>
  </security-role-assignment>
</weblogic-ejb-jar>
```

---

6. 発信リクエストの Tuxedo 環境を準備するには、次の手順を実行します。
  - 必要に応じて、tpgrpadd を使用してグループを追加します。
  - 必要に応じて、tpusradd を使用してユーザ John、Bob、および Dan を追加します。
  - tpacladd を使用して、Tuxedo ACL が保護する TOUPPER サービスを追加します。
  - ACL\_POLICY="GLOBAL" を使用して、リモート ドメイン (WebLogic Server ドメイン) の BDMCONFIG を設定します。
7. 着信リクエストの Tuxedo 環境を準備するには、次の手順を実行します。
  - 必要に応じて、tpgrpadd を使用してグループを追加します。
  - 必要に応じて、tpusradd を使用してユーザ John、Bob、および Dan を追加します。
  - tpacladd を使用して、Tuxedo ACL が保護する TOUPPER サービスを追加します。
  - ACL\_POLICY="GLOBAL" を使用して、リモート ドメイン (WebLogic Server ドメイン) の BDMCONFIG を設定します。ACL\_POLICY="LOCAL"

を設定する場合、`tpusradd` を使用し、ユーザとしてリモート `DOMAINID` または `AccessPointID` をコンフィグレーションする必要があります。

8. **WebLogic Server** 環境を準備するには、次の手順を実行します。
  - **Tuxedo** から `tpusr` ファイルをコピーするか、独自の `tpusr` ファイルを生成します。
  - **WTCServer MBean** の **WTCRemoteTuxDom MBean** の `TpUserFile` 属性に、`tpusr` ファイルの絶対パスとその名前を設定します。
9. **WTCServer MBean** の **WTCRemoteTuxDom MBean** の `CredentialPolicy` 属性を `GLOBAL` に設定します。

## リンクレベルの暗号化

ドメイン間で暗号化を使用すると、データのプライバシーを守ることができます。これによって、ネットワークベースの傍受者は、あるドメイン ゲートウェイから別のドメイン ゲートウェイに送信されるメッセージやアプリケーション生成メッセージの内容を知ることができなくなります。このセキュリティメカニズムは、**WTCServer MBean** の **WTCLocalTuxDom MBean** および **WTCRemoteTuxDom MBean** で `MINENCRYPTBITS` および `MAXENCRYPTBITS` パラメータを設定することによってコンフィグレーションします。

**注意：** 暗号を使用するには、適切なライセンスが必要です。ライセンス要件の詳細については、1-4 ページの「ライセンス」を参照してください。



---

## 4 クラスタ環境における WebLogic Tuxedo Connector の管理方法

**注意：** WebLogic Server クラスタの詳細については、『WebLogic Server クラスタ ユーザーズ ガイド』を参照してください。

以下の節では、クラスタ環境で使用する WebLogic Tuxedo Connector の管理およびコンフィグレーション方法について説明します。

- クラスタ環境における WebLogic Tuxedo Connector のガイドライン
- Tuxedo ドメインに対する送信リクエストのコンフィグレーション方法
- Tuxedo ドメインからの着信リクエストのコンフィグレーション方法

### クラスタ環境における WebLogic Tuxedo Connector のガイドライン

クラスタ環境で WebLogic Tuxedo Connector をデプロイする際は、以下のガイドラインに従ってください。

- バインドはクラスタ内のほかのサーバではレプリケートされないため、クラスタ内のすべての WebLogic Server には、コンフィグレーション済みの WebLogic Tuxedo Connector が必要です。クラスタ内の 1 つのサーバにデプロイ済みの WebLogic Tuxedo Connector がない場合、EJB または MDB は、その接続に対する Tuxedo 接続ファクトリを見つけれられません。
- サーバをシャットダウンせずに、クラスタ内の WebLogic Tuxedo 接続インスタンスをシャットダウンすることはできません。

- WebLogic Server の 1 クラスタ内のすべての WebLogic Tuxedo 接続インスタンスは、WTCImport MBean で定義された同じサービスを持つ必要があります。
- WebLogic Tuxedo Connector を使用する EJB または MDB は、そのクラスタ内のすべての WebLogic Server にデプロイする必要があります。
- 管理者は、クラスタ化されたノード間で、適切なロード バランシングとフェイルオーバーが行われるために、TUXEDO DMCONFIG が正しくコンフィグレーションされていることに責任があります。
- ATMI 接続は、名前を指定して通知された Tuxedo サービスを使用して、WTCImport MBean でサービスを識別します。
- WebLogic Tuxedo Connector は、クラスタ環境における着信 RMI/IIOP または発信 CORBA をサポートしません。

# Tuxedo ドメインに対する送信リクエストのコンフィグレーション方法

**注意：** WebLogic Server クラスタの詳細については、「クラスタでの通信」を参照してください。WebLogic Tuxedo Connector は、ドメインレベルのフェイルオーバーおよびフェイルバック機能も提供します。詳細については、3-5 ページの「ドメインレベルのフェイルオーバーとフェイルバックのコンフィグレーション」を参照してください。

WebLogic Server からの送信リクエストのロード バランシングとフェイルオーバーは、WebLogic Server EJB および MDB によって異なります。

## クラスタ化した WebLogic Tuxedo Connector の コンフィグレーション例

以下のコンフィグレーションは、クラスタ環境にある WebLogic Tuxedo Connector の例です。このクラスタは、1つの管理サーバ (wtcAServer) と、3つの管理対象サーバ (wtcMServer1、wtcMServer2、wtcMServer3) から構成されています。管理対象サーバにはそれぞれ、コンフィグレーション済みの WTCServer MBean があり、その WTCImport MBean に同じサービス (TOUPPER) が含まれています。

### コード リスト 4-1 クラスタ化した WebLogic Tuxedo Connector のコンフィグレーション例

```
<Domain Name="wtcDomain" >
<Security Name="wtcDomain" Realm="mysecurity"/>
<Realm Name="mysecurity" FileRealm="myrealm"/>
<FileRealm Name="myrealm"/>

<Cluster Name="wtcCluster" MulticastAddress="239.0.0.20"
      MulticastPort="7700" MulticastTTL="1"/>

<Security GuestDisabled="false"/>

<Server Name="wtcAServer" NativeIOEnabled="true" ListenPort="3472"
      ListenAddress="mymachine" TunnelingEnabled="true" >
<SystemDataStore ListenPort="7555" ListenAddress="mymachine" />
</Server>

<Server Name="wtcMServer1" Cluster="wtcCluster"
NativeIOEnabled="true" ListenPort="7701" ListenAddress="mymachine"
TunnelingEnabled="true" > </Server>

<Server Name="wtcMServer2" Cluster="wtcCluster"
NativeIOEnabled="true" ListenPort="7702" ListenAddress="mymachine"
TunnelingEnabled="true" > </Server>

<Server Name="wtcMServer3" Cluster="wtcCluster"
NativeIOEnabled="true" ListenPort="7703" ListenAddress="mymachine"
TunnelingEnabled="true" > </Server>
```

```
<WTCServer Name="WTCServer1" Targets="wtcMServer1">
<WTCEXport EJBName="tuxedo.services.TOLOWERHome"
  LocalAccessPoint="WDOM1" Name="exp0" ResourceName="TOLOWER"/>
<WTCImport LocalAccessPoint="WDOM1" Name="imp0"
  RemoteAccessPointList="TDOM2,TDOM1" ResourceName="TOUPPER"/>
<WTCLocalTuxDom AccessPoint="WDOM1" AccessPointId="WDOM1"
  BlockTime="30000" ConnectionPolicy="ON_DEMAND"
  NWAddr="//mymachine:20401" Name="ltd0" Security="NONE"/>
<WTCRemoteTuxDom AccessPoint="TDOM1" AccessPointId="TDOM1"
  LocalAccessPoint="WDOM1" NWAddr="//123.123.123.123:20301"
  Name="rtd0"/>
<WTCRemoteTuxDom AccessPoint="TDOM2" AccessPointId="TDOM2"
  LocalAccessPoint="WDOM1" NWAddr="//123.123.123.123:20302"
  Name="rtd1"/>
</WTCServer>

<WTCServer Name="WTCServer2" Targets="wtcMServer2" >
<WTCEXport EJBName="tuxedo.services.TOLOWERHome"
  LocalAccessPoint="WDOM2" Name="exp1" ResourceName="TOLOWER"/>
<WTCImport LocalAccessPoint="WDOM2" Name="imp1"
  RemoteAccessPointList="TDOM1,TDOM2" ResourceName="TOUPPER"/>
<WTCLocalTuxDom AccessPoint="WDOM2" AccessPointId="WDOM2"
  BlockTime="30000" ConnectionPolicy="ON_DEMAND"
  NWAddr="//mymachine:20402" Name="ltd1" Security="NONE"/>
<WTCRemoteTuxDom AccessPoint="TDOM1" AccessPointId="TDOM1"
  LocalAccessPoint="WDOM2" NWAddr="//123.123.123.123:20301"
  Name="rtd2"/>
<WTCRemoteTuxDom AccessPoint="TDOM2" AccessPointId="TDOM2"
  LocalAccessPoint="WDOM2" NWAddr="//123.123.123.123:20302"
  Name="rtd3"/>
</WTCServer>

<WTCServer Name="WTCServer3" Targets="wtcMServer3" >
<WTCEXport EJBName="tuxedo.services.TOLOWERHome"
  LocalAccessPoint="WDOM3" Name="exp2" ResourceName="TOLOWER"/>
<WTCImport LocalAccessPoint="WDOM3" Name="imp2"
  RemoteAccessPointList="TDOM1,TDOM2" ResourceName="TOUPPER"/>
<WTCLocalTuxDom AccessPoint="WDOM3" AccessPointId="WDOM3"
  BlockTime="30000" ConnectionPolicy="ON_DEMAND"
  NWAddr="//mymachine:20403" Name="ltd2" Security="NONE"/>
<WTCRemoteTuxDom AccessPoint="TDOM1" AccessPointId="TDOM1"
  LocalAccessPoint="WDOM3" NWAddr="//123.123.123.123:20301"
  Name="rtd4"/>
<WTCRemoteTuxDom AccessPoint="TDOM2" AccessPointId="TDOM2"
  LocalAccessPoint="WDOM3" NWAddr="//123.123.123.123:20302"
```



```
        Name="rtd5"/>
</WTCServer>

</Domain>
```

---

## Tuxedo ドメインからの着信リクエストの コンフィグレーション方法

Tuxedo からの着信リクエストのロード バランシングとフェイルオーバーは、Tuxedo ドメインの DMCONFIG コンフィグレーションによって異なります。

### ロード バランシング

**注意：** 詳細については、「ロード・バランシング」を参照してください。

Tuxedo からクラスタ化された WTC までのロード バランシングを行う Tuxedo DMCONFIG の例を以下に示します。このコンフィグレーションでは、WebLogic Server クラスタ内に 3 つのノードがあります。各ノードには、正しくコンフィグレーションされた WebLogic Tuxedo Connector インスタンスが 1 つあり、エクスポートされたサービスを提供し、そのサービスから Tuxedo クライアントへのアクセスが可能です。

```
*DM_IMPORT
TOUPPER LDOM=tuxedo_dom RDOM=WDOM1 LOAD=50
TOUPPER LDOM=tuxedo_dom RDOM=WDOM2 LOAD=50
TOUPPER LDOM=tuxedo_dom RDOM=WDOM3 LOAD=50
```

# フェイルオーバー

**注意：** 詳細については、「Domains 環境でのフェイルオーバーとフェイルバックを設定する」を参照してください。

以下に、Tuxedo フェイルオーバー機能で示したのと同様に、WebLogic Server ノード間のロード バランシングを行う、より洗練されたコンフィグレーションを使用する Tuxedo DMCONFIG の例を示します。ドメインレベルのフェイルオーバーおよびフェイルバックを有効にするには、Tuxedo ドメインを ON\_STARTUP または INCOMING\_ONLY の CONNECTION\_POLICY でコンフィグレーションする必要があります。

```
*DM_IMPORT  
  
TOUPPER LDOM=tuxedo_dom RDOM=WDOM1,WDOM2,WDOM3 LOAD=50  
TOUPPER LDOM=tuxedo_dom RDOM=WDOM2,WDOM3,WDOM1 LOAD=50  
TOUPPER LDOM=tuxedo_dom RDOM=WDOM3,WDOM1,WDOM2 LOAD=50
```

---

## 5 CORBA アプリケーションの管理

**注意：** CORBA アプリケーションの詳細については、「CORBA のトピック」ページを参照してください。

以下の節では、Tuxedo CORBA クライアントおよびサービスをサポートする WebLogic Tuxedo Connector の管理およびコンフィグレーション方法について説明します。

- CORBA サービスアプリケーション用 WebLogic Tuxedo Connector のコンフィグレーション方法
- 着信 RMI-IIOP の WebLogic Tuxedo Connector を管理およびコンフィグレーションする方法
- 送信 RMI-IIOP の WebLogic Tuxedo Connector をコンフィグレーションする方法

### CORBA サービス アプリケーション用 WebLogic Tuxedo Connector のコンフィ グレーション方法

**注意：** WebLogic Tuxedo Connector MBean のコンフィグレーション方法の詳細については、2-3 ページの「アプリケーション用の WebLogic Tuxedo Connector のコンフィグレーション」を参照してください。

この節では、WebLogic Server EJB から Tuxedo CORBA サーバへの呼び出しをサポートする WTCServer MBean のコンフィグレーション方法について説明します。WTCServer MBean のコンフィグレーションは、次の手順に従います。

1. WebLogic Server ドメインの `WTCLocalTuxDom MBean` をコンフィグレーションします。
2. Tuxedo CORBA ドメインの `WTCRemoteTuxDom MBean` セクションをコンフィグレーションします。
3. `WTCImport MBean` をコンフィグレーションします。
  - `ResourceName` を「`//domain_id`」に設定します。「`domain_id`」は、オブジェクトがデプロイされるリモート Tuxedo ドメインの Tuxedo UBBCONFIG ファイルで指定されている `DOMAINID` です。CORBA ドメインのこのユニークな識別子の最大長は、`//` を含めて 15 文字です。
  - `Set LocalAccessPoint` を `WTCRemoteTuxDom MBean` の `LocalAccessPoint` 属性の値に設定します。
  - `RemoteAccessPointList` を `WTCRemoteTuxDom MBean` の `AccessPointId` 属性の値に設定します。

WebLogic Server EJB を使用して Tuxedo CORBA サービスを呼び出すクライアント アプリケーションの開発方法については、『WebLogic Tuxedo Connector プログラマーズ ガイド』を参照してください。

## WTCServer MBean および Tuxedo UBB ファイルの例

次の `WTCServer MBean` では、Tuxedo CORBA サーバに対する `WTCImport MBean` のコンフィグレーション方法の例を示します。

### コード リスト 5-1 CORBA サーバアプリケーションの `WTCServer MBean` の例

---

```
<WTCServer Name="WTCsimpappCNS"
  <WTCImport LocalAccessPoint="examples"
    Name="myImportedResources" RemoteAccessPointList="TUXDOM"
    ResourceName="//simpapp"/>
  <WTCLocalTuxDom AccessPoint="examples" AccessPointId="examples"
    ConnectionPolicy="ON_DEMAND" NWAddr="//123.123.123.123:5678"
    Name="myLoclTuxDom" Security="NONE" />
  <WTCRemoteTuxDom AccessPoint="TUXDOM" AccessPointId="TUXDOM"
    LocalAccessPoint="examples" NWAddr="//123.123.123.123:1234"
```

```
Name="myRTuxDom" />
</WTCServer>
```

---

以下のサンプル Tuxedo UBB コンフィグレーション ファイルには、simpapp という DOMAINID 名が定義されています。DOMAINID 名は、WTCImport MBean の ResourceName 属性で使用されます。

### コード リスト 5-2 CORBA サーバアプリケーションのサンプル Tuxedo UBB ファイル

---

```
*RESOURCES
  IPCKEY      55432
  DOMAINID   simpapp
  MASTER     SITE1
  MODEL      SHM
  LDBAL      N
*MACHINES
  "YODA"
  LMID=SITE1
  APPDIR="your APPDIR"
  TUXCONFIG="APPDIR\tuxconfig"
  TUXDIR="your TUXDIR"
  MAXWSCLIENTS=10
*GROUPS
  SYS_GRP
    LMID=SITE1
    GRPNO=1
  APP_GRP
    LMID=SITE1
    GRPNO=2
*SERVERS
  DEFAULT:
    RESTART=Y
    MAXGEN=5
  TMSYSEVT
    SRVGRP=SYS_GRP
    SRVID=1
  TMFFNAME
    SRVGRP=SYS_GRP
    SRVID=2
    CLOPT=" -A -- -N -M"
  TMFFNAME
    SRVGRP=SYS_GRP
    SRVID=3
    CLOPT=" -A -- -N"
  TMFFNAME
    SRVGRP=SYS_GRP
    SRVID=4
    CLOPT=" -A -- -F"
```

```
ISL
    SRVGRP=SYS_GRP
    SRVID=5
    CLOPT="-A -- -n <// 自分の tux マシン : 2468>"

cns
    SRVGRP=SYS_GRP
    SRVID=6
    CLOPT="-A --"
DMADM SRVGRP=SYS_GRP SRVID=7
GWADM SRVGRP=SYS_GRP SRVID=8
GWTDOMAIN SRVGRP=SYS_GRP SRVID=9
simple_server
    SRVGRP=APP_GRP
    SRVID=1
    RESTART = N

*SERVICES
```

---

## 着信 RMI-IIOP の WebLogic Tuxedo Connector を管理およびコンフィグレーションする方法

この節では、アプリケーション環境の管理方法および Tuxedo CORBA オブジェクトが RMI-IIOP API を使用して WebLogic Server にデプロイした EJB を呼び出すことを可能にする WTCServer MBean のコンフィグレーション方法について説明します。

- WTCServer MBean のコンフィグレーション
- Tuxedo アプリケーション環境の管理

### WTCServer MBean のコンフィグレーション

**注意：** WebLogic Tuxedo Connector MBean のコンフィグレーション方法の詳細については、2-3 ページの「アプリケーション用の WebLogic Tuxedo Connector のコンフィグレーション」を参照してください。

環境に合わせて、`WTCLocalTuxDom MBean` および `WTCRemoteTuxDom MBean` をコンフィグレーションします。RMI-IIOP API を使用して、Tuxedo CORBA オブジェクトが WebLogic Server にデプロイした EJB を呼び出すことを可能にするために必要な特別な管理手順は必要ありません。

## Tuxedo アプリケーション環境の管理

**注意：** Tuxedo アプリケーション環境のコンフィグレーション方法の詳細については、「管理」ページを参照してください。

Tuxedo アプリケーション環境をコンフィグレーションする場合は、さらに次の手順を実行する必要があります。

1. 環境の `TOBJADDR` を設定します。

例：`//<hostname>:2468`

2. 次のコマンドを入力して、Tuxedo ドメインの `CosNaming` ネームスペースに WebLogic Server (WLS) ネーミング サービスを登録します。

```
cnbind -o ior.txt your_bind_name
```

*your\_bind\_name* は、使用する Tuxedo アプリケーションから得た `CosNaming` サービス オブジェクト名です。

`ior.txt` ファイルには、WebLogic Server のドメイン ネーミング サービスが含まれています。

### コード リスト 5-3 `iiop.ejb.stateless.server.tux` Tuxedo クライアントの `ior.txt` ファイルの例

---

```
corbaloc:tgioop:myServer/NameService
```

---

*myServer* は、サーバ名です。

3. Tuxedo ドメイン コンフィグレーション ファイルの `*DM_REMOTE_SERVICES` を変更します。それまで `DOMAINID` だった **WebLogic Server** サービス名を使用する **WebLogic Server** の名前に置き換えます。

### コード リスト 5-4 ドメイン コンフィグレーション ファイル

---

```
*DM_RESOURCES
  VERSION=U22

*DM_LOCAL_DOMAINS
  TDOM1 GWGRP=SYS_GRP
  TYPE=TDOMAIN
  DOMAINID="TDOM1"
  BLOCKTIME=20
  MAXDATALEN=56
  MAXRDOM=89

*DM_REMOTE_DOMAINS
  TDOM2 TYPE=TDOMAIN
  DOMAINID="TDOM2"

*DM_TDOMAIN
  TDOM1 NWADDR="< Tuxedo ドメインのネットワーク アドレス : ポート >"
  TDOM2 NWADDR="< WTC ドメインのネットワーク アドレス : ポート >"

*DM_REMOTE_SERVICES
  "//myServer"
```

---

`myServer` は、サーバ名です。

4. `dmloadcf` を使用して、修正したドメイン コンフィグレーション ファイルをロードします。

## オブジェクト参照としてのサーバ名使用に関するガイドライン

この節では、オブジェクト参照として使用するサーバ名を作成する際に、覚えておく必要のあるガイドラインを示します。

- `*DM_REMOTE_SERVICES` セクションのなかで、Tuxedo が受け入れる最大フィールド長は、//を含めて 15 文字です。たとえば、サーバ名が



examplesServer だとすると、\*DM\_REMOTE\_SERVICES オブジェクト参照は //examplesServe です。

- 複数のサーバが必要な場合、サーバ名は最初の 13 文字が一意にならなければなりません。
- ior.txt ファイルにあるサーバ名の完全な名前が 13 文字を超えている場合には、それを使用できません。例  
:corbaloc:tgiiop:examplesServer/NameService

# 送信 RMI-IIOP の WebLogic Tuxedo Connector をコンフィグレーションする方法

**注意：** WebLogic Tuxedo Connector MBean のコンフィグレーション方法の詳細については、2-3 ページの「アプリケーション用の WebLogic Tuxedo Connector のコンフィグレーション」を参照してください。

この節では、RMI-IIOP API を使用して Tuxedo CORBA オブジェクトが WebLogic Server にデプロイした EJB を呼び出すことを可能にする方法について説明します。以下の手順で、WTCServer MBean を変更します。

1. ローカル WLS ドメイン用の WTCLocalTuxDom MBean をコンフィグレーションします。
2. リモート Tuxedo ドメイン用の WTCRemoteTuxDom MBean をコンフィグレーションします。送信 RMI-IIOP では、FederationURL および FederationName という 2 つの要素を追加する必要があります。
  - FederationURL に、JNDI に結合する外部ネームサービスに対する URL を設定します。これは、EJB がリモート Tuxedo CORBA オブジェクトへのアクセスに使用する初期コンテキストを取得するために使用する URL と同じでなければなりません。
  - FederationName に、結合ポイントの識別名を設定します。
3. WTCImport MBean をコンフィグレーションします。

- ResourceName を「//domain\_id」に設定します。「domain\_id」は、オブジェクトがデプロイされるリモート Tuxedo ドメインの Tuxedo UBBCONFIG ファイルで指定されている DOMAINID です。この CORBA ドメインのユニークな識別子の最大長は、//を含めて 15 文字です。
- LocalAccessPoint を、WTCRemoteTuxDom MBean の LocalAccessPoint 要素の値に設定します。
- RemoteAccessPointList を、WTCRemoteTuxDom MBean の AccessPointId 要素の値に設定します。

RMI-IIOP を使用して WebLogic Server EJB を使用する Tuxedo サービスを呼び出すアプリケーションを開発する方法については、『WebLogic Tuxedo Connector プログラマーズガイド』を参照してください。

## 送信 RMI-IIOP コンフィグレーション例

以下の WTCServer MBean は、送信 RMI-IIOP のための WebLogic Tuxedo Connector のコンフィグレーション例を示したものです。

### コード リスト 5-5 送信 RMI-IIOP の WTCServer MBean の例

---

```
<WTCServer Name="WTCTrader">
  <WTCImport LocalAccessPoint="TDOM2" Name="myImportedResources"
  RemoteAccessPointList="TDOM1" ResourceName="//simpapp"/>
  <WTCLocalTuxDom AccessPoint="TDOM2" AccessPointId="TDOM2"
  ConnectionPolicy="ON_DEMAND" NWAddr="//123.123.123.123:5678"
  Name="myLocalTuxDom" Security="NONE"/>
  <WTCRemoteTuxDom AccessPoint="TDOM1" AccessPointId="TDOM1"
  FederationName="tuxedo.corba.remote"
  FederationURL="corbaloc:tglop:simpapp/NameService"
  LocalAccessPoint="TDOM2" NWAddr="//123.123.123.123:1234"
  Name="myRTuxDom"/>
</WTCServer>
```

---

---

## 6 tBridge メッセージ インタフェースのコンフィグレーション

以下の節では、tBridge の機能とコンフィグレーションについて説明します。

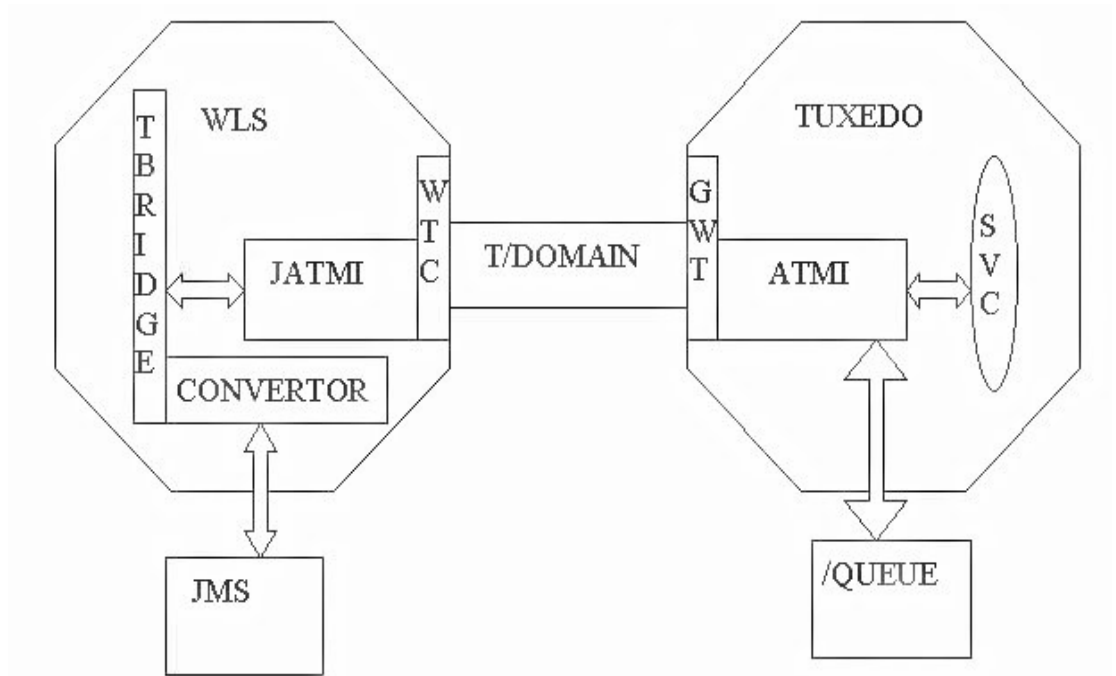
- tBridge の概要
- tBridge 用 WTCServer MBean コンフィグレーション
- tBridge の接続性
- 接続タイプのコンフィグレーションの例
- 優先度のマッピング
- エラー キュー

### tBridge の概要

tBridge は WebLogic Tuxedo Connector の一部で、Tuxedo アプリケーション環境と通信する WebLogic Server アプリケーションの双方向 JMS インタフェースを提供します。環境間でのメッセージングの転送は、クライアントアプリケーションの代わりにサービスを呼び出すのに使用するテキスト、Byte、または XML データ ストリームを含む JMS ベースのメッセージで構成されます。

tBridge の機能は以下のように決定されます。

- 接続性は、WTCServer MBean の WTCtBridgeGlobal MBean および WTCtBridgeRedirect MBean にある属性のコンフィグレーションによって決定されます。
- tBridge は、Java Messaging Service (JMS) を使用して Tuxedo /Q または Tuxedo サービスへのインタフェースを提供します。



- tBridge は、既存の Tuxedo システムに接続性を提供するために、XML と FML32 間においてシンプルな変換を提供します。

## tBridge による JMS から Tuxedo への接続

**注意：** すべてのメッセージは、確認応答されるまで JMS キューに保持されます。

この節では、JMS のメッセージが tBridge を介して Tuxedo キューおよびサービスに渡されるまでの流れを説明します。

1. Web が有効化された WLPI アプリケーションなどの JMS クライアントにより、Tuxedo で処理されるメッセージが JMS キューに配置されます。このメッセージがトランザクションの一部だった場合、トランザクションがコミットされます。
2. メッセージが JMS キューから移動し、tBridge コンバータで処理されます。

3. tBridge コンバータによりメッセージタイプがチェックされ、サポートされている JMS メッセージタイプが JATMI のバッファタイプに変換されます。
  - BytesMessage、TextMessage、XML が、それぞれ TypedCArray、TypedString、TypedFML32 に変換されます。TranslateFML 属性の値に従って、XML/FML 変換が実行されます。
  - 変換エラーが発生した場合、エラーは wlsServerErrorDestination キューに送られ、メッセージは JMS セッションで確認応答されます。
  - 認識されない JMS メッセージを受信した場合、適宜エラーメッセージがログされ、そのメッセージは確認応答された後、破棄されます。これはコンフィグレーションエラーと解釈され、tBridge はメッセージをエラーキューにリダイレクトしません。
4. 変換されたメッセージが、T/Domain ゲートウェイを使用して Tuxedo に送られます。
  - JmsQ2TuxQ のリダイレクトセットを利用し、JATMI の tpenqueue を使用して、メッセージが Tuxedo キューに送られます。
  - JmsQ2TuxS のリダイレクトセットを利用し、JATMI の tpcall を使用して、メッセージが Tuxedo キューに送られます。
5. tpenqueue または tpcall が成功した場合、戻り値が ReplyQ に置かれます。メッセージが JMS セッションで確認応答されます。
  - tpenqueue または tpcall が失敗した場合、メッセージは tBridge により wlsServerErrorDestination キューに送られ、JMS セッションで確認応答されます。wlsServerErrorDestination キューがコンフィグレーションされていない場合、メッセージは破棄され、tBridge は応答確認されていない次のメッセージを処理します。

## tBridge による Tuxedo から JMS への接続

**注意：** tBridge はトランザクションを使用して、Tuxedo/Q から JMS キューへの転送中にメッセージが失われないようにします。

この節では、TuxQ2JmsQ リダイレクトを利用して、Tuxedo のメッセージが tBridge を介して JMS キューに渡されるまでの流れを説明します。

1. tBridge により、Tuxedo キューの使用可能なメッセージがポーリングされます。
2. Tuxedo サービスにより、メッセージが Tuxedo キューに配置されます。
3. tBridge では、JATMI の tpdequeue を使用して、Tuxedo からメッセージを転送し、JMS キューに配置します。
  - 何らかの理由で、指定されたリトライ回数を使い果たしてもメッセージが JMS キューにリダイレクトされない場合、メッセージは Tuxedo キューと同じキュー スペースにある tuxErrorDestination キューに配置されます。
  - tBridge がメッセージを tuxErrorDestination キューに配置できなかった場合は、エラー ログが記録されて、メッセージは失われます。
  - tuxErrorDestination キューが指定されていない場合、メッセージは失われます。

## tBridge の制限

tBridge には次のような制限があります。

- JMS からメッセージを取り出して Tuxedo キューに配置する場合や、Tuxedo サービスを呼び出す場合には、トランザクションは使用されません。
- tBridge はスレッドインテンシブです。JMS キューから Tuxedo へメッセージが転送されるたびに、スレッドが使用されます。コンフィグレーションされている Tuxedo キューをモニタするには、ポーリング スレッドが必要になります。
- XML/FML 変換は、簡単なメッセージ構造の構築のために行われます。XML から FML への変換の詳細については、7-7 ページの「FML32 の考慮事項」を参照してください。

# tBridge 用 WTCServer MBean コンフィグレーション

WebLogic Tuxedo Connector の tBridge 接続性は、Tuxedo への接続を確立するために必要な情報を含む `WTCTBridgeGlobal MBean` および `WTCTBridgeRedirect MBean` のコンフィグレーションによって決定されます。

## tBridge の起動

tBridge は、WTCServer MBean の `WTCTBridgeGlobal MBean` および `WTCTBridgeRedirect MBean` がコンフィグレーションされていて、かつ、WTCServer MBean が対象サーバにデプロイされている場合は、WebLogic Server アプリケーション環境の一部として起動されます。tBridge が起動できないようなコンフィグレーション条件が発生した場合は、エラーがログされます。

## エラーのロギング

WebLogic Tuxedo Connector エラーは、WebLogic Server エラー ログにログされます。

## tBridge の接続性

**注意：** `MapMessage`、`ObjectMessage`、`StreamMessage` の JMS メッセージタイプは、WebLogic Tuxedo Connector では無効です。これらのメッセージタイプの 1 つが tBridge によって受信された場合は、サポートされていないタイプであることを示すログ エントリが作成され、メッセージが破棄されます。

tBridge は、JMS キューと Tuxedo /Q または JMS キューと Tuxedo サービスのインスタンス間で、一方向のデータ接続を確立します。この接続は、**WTCtBridgeRedirect MBean** で指定され、識別されたポイント間に 1 対 1 の接続を提供します。3 つのタイプの接続がコンフィグレーションできます。次に、各接続タイプについて説明します。

- **JmsQ2TuxQ**: 特定の JMS キューから読み取りを行い、指定された Tuxedo /Q にメッセージを転送します。
- **TuxQ2JmsQ**: Tuxedo /Q から読み取りを行い、JMS にメッセージを転送します。
- **JmsQ2TuxS**: 特定の JMS キューから読み取りを行い、同時に指定された Tuxedo サービスを呼び出し、指定された JMS キューに応答を返します。

## 接続タイプのコンフィグレーションの例

以下の節では、各接続タイプのコンフィグレーションの例を紹介します。

### JmsQ2TuxQ コンフィグレーションの例

JMS キューから読み取りを行い、Tuxedo /Q に送信するコードの例を以下に示します。

```
<WTCtBridgeRedirect
  Direction="JmsQ2TuxQ"
  Name="redir0"
  ReplyQ="RPLYQ"
  SourceName="weblogic.jms.Jms2TuxQueue"
  TargetAccessPoint="TDOM2"
  TargetName="STRING"
  TargetQspace="QSPACE"
  TranslateFML="NO"/>
```

次に、JmsQ2TuxQ コンフィグレーションの各コンポーネントを説明します。

- **Direction** 接続タイプは *JmsQ2TuxQ* です。
- **SourceName** は、読み取る JMS キューの名前が *weblogic.jms.Jms2TuxQueue* であることを指定します。tBridge は、



CLIENT\_ACKNOWLEDGE セマンティクスを使用して、このキューに JMS クライアントセッションを確立します。

- TargetAccessPoint は、アクセスポイントの名前が *TDOM2* であることを指定します。
- TargetQspace は、Qspace の名前が *Qspace* であることを指定します。
- TargetName は、キューの名前が *STRING* であることを指定します。
- ReplyQ は、JMS 応答キューの名前が *RPLYQ* であることを指定します。このキューを使用すると、tpenqueue は TMFORWARD 機能を提供します。
- TranslateFML が NO になっていると、tBridge がデータ変換を提供しないことを指定します。

以下の表に、JmsQtoTuxQ メッセージマッピングの情報を示します。

マッピング元 : JMS メッセージタイプ	マッピング先 : WebLogic Tuxedo Connector JATMI (Tuxedo)
BytesMessage	TypedCArray
TextMessage (translateFML = NONE)	TypedString
TextMessage (translateFML = FLAT)	TypedFML32

## TuxQ2JmsQ コンフィグレーションの例

以下に、Tuxedo /Q キューから読み取りを行い、JMS キューに送信するコードの例を示します。

```
<WTctBridgeRedirect
  Direction="TuxQ2JmsQ"
  Name="redirl"
  SourceAccessPoint="TDOM2"
  SourceName="STRING"
  SourceQspace="QSPACE"
  TargetName="weblogic.jms.Tux2JmsQueue"
  TranslateFML="NO"/>
```

次に、TuxQ2JmsQ コンフィグレーションの各コンポーネントを説明します。

- Direction 接続タイプは *TuxQ2JmsQ* です。

- `TargetName` は、読み取る JMS キューの名前が `weblogic.jms.Tux2JmsQueue` であることを指定します。
- `SourceAccessPoint` は、アクセス ポイントの名前が `TDOM2` であることを指定します。
- `SourceQspace` は、`Qspace` の名前が `Qspace` であることを指定します。
- `SourceName` は、キューの名前が `STRING` であることを指定します。
- `TranslateFML` が `NO` になっていると、`tBridge` がデータ変換を提供しないことを指定します。
- `TranslateFML` が `Flat` になっていると、`tBridge` がそのデータを FML から XML に変換することを指定します。

以下の表に、`TuxQ2JmsQ` メッセージ マッピングの情報を示します。

マッピング元 : WebLogic Tuxedo Connector JATMI (Tuxedo)	マッピング先 : JMS メッセージ タイプ
<code>TypedCArray</code>	<code>BytesMessage</code>
<code>TypedString</code> ( <code>translateFML = NO</code> )	<code>TextMessage</code>
<code>TypedFML32</code> ( <code>translateFML = FLAT</code> )	<code>TextMessage</code>
<code>TypedFML</code> ( <code>translateFML = FLAT</code> )	<code>TextMessage</code>
<code>TypedXML</code>	<code>TextMessage</code>

## JmsQ2TuxS コンフィグレーションの例

**注意：** XML/FML 変換の詳細については、第 7 章「WebLogic Tuxedo Connector での FML の使用」を参照してください。

次に、JMS キューから読み取りを行い、Tuxedo サービスを呼び出して、結果を JMS キューに返すコードの例を示します。

```
<WTCTBridgeRedirect
  Direction="JmsQ2TuxS"
  Name="redir0"
  ReplyQ="weblogic.jms.Tux2JmsQueue"
```

```

SourceName="weblogic.jms.Jms2TuxQueue"
TargetAccessPoint="TDOM2"
TargetName="TOUPPER"
TranslateFML="FLAT"/>

```

次に、JmsQ2TuxS コンフィギュレーションの各コンポーネントを説明します。

- Direction 接続タイプは *JmsQ2TuxS* です。
- SourceName は、読み取る JMS キューの名前が *weblogic.jms.Jms2TuxQueue* であることを指定します。
- TargetAccessPoint は、アクセスポイントの名前が *TDOM2* であることを指定します。
- TargetName は、キューの名前が *TOUPPER* であることを指定します。
- ReplyQ は、JMS 応答キューの名前が *weblogic.jms.Tux2JmsQueue* であることを指定します。
- TranslateFML が *FLAT* になっていると、JMS メッセージの受信時に、メッセージが XML フォーマットで、対応する FML32 データバッファに変換されることを指定します。メッセージは、引数 *TDOM2* および *TOUPPER* とともに *tpcall1* に配置されます。メッセージは FML32 から XML に変換され、*weblogic.jms.Tux2JmsQueue* に配置されます。

以下の表に、JMSQ2TuxX メッセージマッピングの情報を示します。

JMS メッセージタイプ	WebLogic Tuxedo Connector JATMI (Tuxedo)	JMS メッセージタイプ
BytesMessage	TypedCArray	BytesMessage
TextMessage (translateFML = NONE)	TypedString	TextMessage
TextMessage (translateFML = FLAT)	TypedFML32	TextMessage

## 優先度のマッピング

WebLogic Tuxedo Connector では、複数の tBridge リダイレクトインスタンスがサポートされています。リダイレクトインスタンスを複数使用すると、多くの環境下でアプリケーションのスケラビリティおよびパフォーマンスが著しく向上します。ただし、その場合メッセージが処理される順序がランダムになります。優先度のマッピングをしても順序は保証されませんが、割り当てられた重要度に従ってメッセージに対応するメカニズムが提供されます。配信の順序を確実にしたい場合は、tBridge リダイレクトインスタンスを 1 つだけ使用してください。

JMS および Tuxedo 間で優先度をマップするには、`priorityMapping` を使用します。

- JMS には 10 の優先度があります (0 ~ 9)。
- Tuxedo/Q には 100 の優先度があります (1 ~ 100)。

この節では、Tuxedo および JMS サブシステム間で優先度をマップするメカニズムについて説明します。2 つのマッピング方向があります。

- JmstoTux
- TuxtoJms

次の `value:range` の組み合わせで示すように、すべての値にデフォルトが提供されています。

- `value` は入力優先度を指定します。
- `range` は結果出力優先度のシーケンシャル グループを指定します。

JmstoTux- 0:1 | 1:12 | 2:23 | 3:34 | 4:45 | 5:56 | 6:67 | 7:78 | 8:89 | 9:100

TuxtoJms- 1-10:0 | 11-20:1 | 21-30:2 | 31-40:3 | 41-50:4 | 51-60:5 | 61-70:6 | 71-80:7 | 81-90:8 | 91-100:9

このコンフィグレーションでは、優先度 7 の JMS メッセージが Tuxedo /Q の優先度 78 に割り当てられます。優先度 47 の Tuxedo /Q は、優先度 4 の JMS に割り当てられます。

# エラー キュー

tBridge が Tuxedo キューまたは JMS キューからメッセージを受信したときに問題を検出した場合、再試行間隔の後に次の処理が行われます。

- 情報がログされます。
- コンフィグレーションされている場合は、メッセージがエラー キューに保存されます。

## wlsServerErrorDestination

wlsErrorDestination キューは、JMS メッセージが Tuxedo エラーまたは変換エラーにより適切に送信できない場合に使用されます。

## サポートされていないメッセージタイプ

認識できない JMS メッセージを受信した場合、適切なエラーメッセージがログされ、そのメッセージは破棄されます。これはコンフィグレーションエラーと解釈され、tBridge はメッセージをエラー キューにリダイレクトしません。

## tuxErrorQueue

tuxErrorQueue は、TuxQ2JmsQ のリダイレクト中の JATMI プリミティブ tpdequeue のエラー キューです。

## 制限

tBridge エラー キューには、次のような制限があります。

- `TuxErrorDestination` は一度だけ指定できます。`ErrorDestination` に関連するエラー キュー名は、すべての `QSPACE` が使用可能な同じエラー キュー名を持っていることを示します。
- エラーが発生した場合、メッセージはソース `QSPACE` に返されます。`QSPACE` が破損または一杯と判断されると、後続のメッセージは消失します。
- エラー時にメッセージを破棄するよう指定する方法はありません。すべてのメッセージが受信されるか、まったく受信されません。
- エラーに関する情報は、サーバ ログでのみ提供されます。

---

# 7 WebLogic Tuxedo Connector での FML の使用

この章では、フィールド操作言語 (FML) および WebLogic Tuxedo Connector が FML を使用する方法について説明します。

- FML の概要
- WebLogic Tuxedo Connector FML API
- FML フィールド テーブルの管理
- tBridge XML/FML32 変換

## FML の概要

**注意：** FML の使い方の詳細については、『FML を使用した BEA Tuxedo アプリケーションのプログラミング』の「FML プログラミング入門」を参照してください。

FML は、フィールド バッファと呼ばれる記憶構造を定義および操作する Java 言語機能のセットです。各フィールド バッファには、フィールドに属性および値の組み合わせが含まれています。各フィールドは次のように構成されます。

- 属性はフィールドの識別子である。
- 関連する値はフィールドのデータ内容を示す。
- オカレンス番号。

FML には、2 つのタイプがあります。

- フィールド長および識別子の 16 ビット値に基づく FML16。ユニークなフィールドは 8191 個、個々のフィールド長は 64K バイト、フィールド バッファ サイズの合計は 64K バイトに制限されています。

- フィールド長および識別子の 32 ビット値に基づく FML32。およそ 3 千万個のフィールドが使用でき、フィールドおよびバッファの長さはおよそ 20 億バイトまで許可されています。

# WebLogic Tuxedo Connector FML API

**注意：** WebLogic Tuxedo Connector は、FML 機能のサブセットを実装していません。たとえば、views はサポートされていません。

FML アプリケーション プログラム インタフェース (API) は、WebLogic Server クラスの Javadoc に含まれている `weblogic.wtc.jatmi` パッケージにドキュメント化されています。

## FML フィールド テーブルの管理

フィールドテーブルは、Tuxedo フィールド テーブルに似た方法で生成されます。フィールドテーブルは、2 つのシステム間で共通のフィールド名定義、フィールドタイプ、および識別番号を提供するテキスト ファイルです。FML を使用して Tuxedo システムと相互運用するには、次の手順を実行する必要があります。

1. Tuxedo システムから WebLogic Tuxedo Connector 環境に、フィールド テーブルをコピーします。

次に例を示す。Tuxedo 配布キットには、`bankapp` という銀行アプリケーション サンプルが含まれています。このアプリケーションには、次の構造を持つ `bankflds` というファイルがあります。

```
#Copyright (c) 1990 Unix System Laboratories, Inc.
#All rights reserved
#ident "@(#) apps/bankapp/bankflds      $Revision: 1.3 $"
# Fields for database bankdb

# name                number  type    flags  comments
ACCOUNT_ID            110    long   -      -
ACCT_TYPE              112    char   -      -
```



```
ADDRESS                109      string  -      -
.
.
.
```

- フィールド テーブル定義を Java ソース ファイルに変換します。

weblogic.wtc.jatmi パッケージの mkfldclass ユーティリティを使用します。このクラスは、FML32 フィールド テーブルを読み取るユーティリティ機能で、FldTbl インタフェースを実装する Java ファイルを生成します。このユーティリティには、次のような 2 つのインスタンスがあります。

```
- mkfldclass
- mkfldclass32
```

コマンドの正しいインスタンスを使用して、bankflds フィールド テーブルを FML32 Java ソースに変換します。次は、mkfldclass を使用した例です。

```
java weblogic.wtc.jatmi.mkfldclass bankflds
```

作成されたファイルは bankapp.java となり、次のような構造を持ちます。

```
import java.io.*;
import java.lang.*;
import java.util.*;
import weblogic.wtc.jatmi.*;

public final class bankflds
    implements weblogic.wtc.jatmi.FldTbl
{
    /** number:110 type:long */
    public final static int ACCOUNT_ID = 33554542;
    /** number:112 type:char */
    public final static int ACCT_TYPE = 67108976;
    /** number:109 type:string */
    public final static int ADDRESS = 167772269;
    /** number:117 type:float */
```

- 次のコマンドを使用して、作成した bankflds ファイルをコンパイルします。

```
javac bankflds.java
```

コンパイルすると、bankflds.class ファイルとなります。ロード時、WebLogic Tuxedo Connector はクラス ファイルを使用して FML32 フィールドからフィールド エントリを追加、検索、および削除します。

4. フィールド テーブルのクラス ファイルをアプリケーション CLASSPATH に追加します。
5. WTCServer MBean を更新します。
  - フィールド テーブルのクラス ファイルの完全に修飾された場所を反映するため、WTCResources MBean を更新します。
  - fml16 または fml32 の FML バッファ タイプを記述するために必要なキーワードを使用します。
  - カンマ区切りのリストに、複数のフィールド テーブル クラスを入力できません。

例：

```
<WTCResources
  FldTbl16Classes="my.bankflds,your.bankflds,more.bankflds"
  Name="BankappResources" />
```

6. フィールド テーブルのクラス定義をロードするために、WebLogic Server を再起動します。

## mkfldclass32 クラスに対する DynRdHdr プロパティの使用

WebLogic Tuxedo Connector には、FML テーブルを別の方法でコンパイルするためのプロパティが提供されています。以下の場合には、DynRdHdr ユーティリティを使用する必要があります。

- 非常に大きな FML テーブルを使用しているので、mkfldclass32 クラスで作成された .java メソッドが、単一のクラスまたはインタフェースの複雑さ全体に関する内部 Java 仮想マシンの制限を越える。
- 非常に大きな FML テーブルを使用しているので、.java メソッドをコンパイルして作成されたクラスをロードできない。

FML テーブルをコンパイルするときには、次の手順で DynRdHdr プロパティを使用します。

1. フィールド テーブル定義を Java ソース ファイルに変換します。

```
java -DDynRdHdr=Path_to_Your_FML_Table weblogic.wtc.jatmi.mkfld
class32 userTable
```

このコマンドの引数の定義は次のとおりです。

属性	説明
<code>-DDynRdHdr</code>	FML テーブルをコンパイルするための WebLogic Tuxedo Connector プロパティ
<code>Path_to_Your_FML_Table</code>	FML テーブルの絶対パスと完全修飾名
<code>weblogic.wtc.jatmi.mkfldclass32</code>	このクラスは、FML32 フィールドテーブルを読み取るユーティリティ機能で、FldTbl インタフェースを実装する Java ファイルを生成する。
<code>userTable</code>	mkfldclass32 クラスによって作成される .java メソッドの名前

2. 次のコマンドを使用して、`userTable` ファイルをコンパイルします。

```
javac userTable.java
```

3. `userTable.class` ファイルをアプリケーション CLASSPATH に追加します。
4. `userTable.class` ファイルの完全に修飾された場所を反映するため、WTCResources MBean を更新します。
5. WTCServer を対象とします。WTCServer サービスが開始されるときに、`userTable.class` がロードされます。

`userTable.class` ファイルの作成後は、`userTable.class` を手動で更新しなくても FML テーブルを変更してその変更をデプロイすることができます。

WTCServer の起動時に、Weblogic Tuxedo Connector は、WTCResources MBean で指定された場所に従って FML テーブルをロードします。

`Path_to_Your_FML_Table` 属性を変更した場合、前の手順に従って

`userTable.java` および `userTable.class` ファイルを更新する必要があります。

## tBridge XML/FML32 変換

**注意:** 指定するデータ型は、FLAT または NO にする必要があります。他のデータ型を指定すると、リダイレクションは失敗します。

WTCTBridgeRedirect MBean の TranslateFML 要素は、FML32 変換がメッセージペイロードで実行されることを示すために使用します。FML32 変換には、FLAT および NO の 2 つのタイプがあります。

### FLAT

メッセージペイロードは、WebLogic Tuxedo Connector の内部 FML32/XML トランスレータを使用して変換されます。フィールドは、メッセージ構造 (階層構造) を意識せず、グループ化を反復せずに、フィールドごとの値に変換されます。

FML32 バッファを XML に変換するため、tBridge は FML32 バッファにある各フィールドの各インスタンスを抽出して文字列に変換し、それをフィールド名で構成されるタグ内に配置します。これらすべてのフィールドは、サービス名で構成されるタグ内に配置されます。たとえば、次のフィールドで構成される FML32 バッファがあるとします。

```

NAME          JOE
ADDRESS       CENTRAL CITY
PRODUCTNAME   BOLT
PRICE         1.95
PRODUCTNAME   SCREW
PRICE         2.50

```

変換された XML バッファは次のようになります。

```

<FML32>
  <NAME>JOE</NAME>
  <ADDRESS>CENTRAL CITY</ADDRESS>
  <PRODUCTNAME>BOLT</PRODUCTNAME>
  <PRODUCTNAME>SCREW</PRODUCTNAME>
  <PRICE>1.95</PRICE>
  <PRICE>2.50</PRICE>
</FML32>

```

# NO

変換は使用されません。tBridge は、リダイレクションの方向に応じて、JMS `TextMessage` を Tuxedo `TypedBuffer (TypedString)` に (またはその逆に) マップします。JMS `BytesMessage` は、Tuxedo `TypedBuffer (TypedCarray)` に (またはその逆に) マップされます。

## FML32 の考慮事項

FML32 を使用する場合は、以下の情報について考慮する必要があります。

- XML 入力では、ルート要素が必要ですが無視されます。
- XML 出力では、ルート要素は常に <FML32> です。
- フィールド テーブル名は、7-2 ページの「FML フィールド テーブルの管理」で説明されているようにロードする必要があります。
- tBridge トランスレータは、「flat」またはリニアなグループ化でのみ有効です。これは FML32 順を記述する情報が管理されないために、繰り返しデータを含むバッファが予期しない形式で示される場合があります。たとえば、部品とそれに関連付けられた価格のリストを含む FML32 バッファがあると仮定します。ここでは PART A、PRICE A、PART B、PRICE B などが予想されますが、tBridge 内に構造的なグループ情報がないため、変換された XML は PART A、PART B など、PRICE A、PRICE B などになります。
- XML を FML32 に変換する場合、トランスレータは空白値を無視します。たとえば、<STRING></STRING> は、結果的に FML32 バッファではスキップされます。
- 組み込み FML は、このリリースではサポートされません。
- FML から XML への変換では、TypedCArray はサポートされません。以下のサポートされているフィールド型から選択してください。
  - SHORT
  - LONG
  - CHAR
  - FLOAT

- DOUBLE
- STRING
- INT (FML32)
- DECIMAL (FML32)
- FML 内に `TypedCArray` が存在する場合、`TypedString` にエンコードし、XML を `TypedCArray` にデコードします。
- バイナリ データを渡す必要がある場合、選択したフィールド型へのコード化を行い、受け取り側で XML をデコードします。

---

## 8 WebLogic Process Integrator と Tuxedo アプリケーションの接続

**注意：** アプリケーションの統合方法の詳細については、「BEA WebLogic Integration」を参照してください。

WebLogic Tuxedo Connector tBridge は、Tuxedo アプリケーションをビジネスワークフローに統合するために必要なインフラストラクチャを WebLogic Process Integrator ユーザに提供します。以下の節では、WebLogic Tuxedo Connector を使用して WebLogic Process Integrator と Tuxedo を統合する方法について説明します。

- WebLogic Process Integrator と Tuxedo の同期接続
- WebLogic Process Integrator と Tuxedo の同期非ブロッキング接続
- WebLogic Process Integrator と Tuxedo の非同期接続
- 非同期 Tuxedo /Q と WebLogic Process Integrator の接続
- Tuxedo と WebLogic Process Integrator の双方向非同期接続

### WebLogic Process Integrator と Tuxedo の同期接続

WebLogic Process Integrator は、JATMI EJB を使用して Tuxedo に対してブロッキング呼び出しを実行します。このプロセスは、次の 3 つの部分で構成されます。

- WebLogic Process Integrator ビジネス オペレーションの定義
- eLink アダプタの呼び出し

- WebLogic Process Integrator 例外ハンドラの定義

### ビジネス オペレーションの定義

使用する JATMI メソッドの WebLogic Process Integrator ビジネス オペレーションを定義します。

- TypedFML32 バッファ操作メソッド。
- JATMI `tpcall()` メソッド。

例 : `out_buffer = tpcall (service_name, in_buffer, flags)`

### eLink アダプタの呼び出し

WebLogic Process Integrator プロセス フローから eLink アダプタを呼び出します。

- 定義されたビジネス オペレーションを使用して、TypedFML32 リクエスト バッファを構築します。
- 定義されたビジネス オペレーションを使用して、サービス名を指定する JATMI `tpcall()` メソッドを呼び出します。
- 定義されたビジネス オペレーションを使用して、TypedFML32 応答バッファを処理します。

### 例外ハンドラの定義

例外を処理するために、WebLogic Process Integrator 例外ハンドラを定義します。

## WebLogic Process Integrator と Tuxedo の同期非ブロッキング接続

WebLogic Process Integrator は、Tuxedo サービスを同期的に呼び出すためにメッセージを送信します。



- JMS キューと Tuxedo サービスの呼び出し間の 1 対 1 の関係。
- Tuxedo サービスからの応答と JMS キュー間の 1 対 1 の関係。
- WebLogic Process Integrator は、メッセージを JMS キューに書き込みます。
- メッセージが JMS キューに入ると、tBridge はターゲットの Tuxedo サービスにメッセージを移動します。
- メッセージは、XML と FML32 間で変換されます。
- 応答は、指定された JMS 応答キューに書き込まれます。
- WebLogic Process Integrator イベント ノードは、応答メッセージの応答キューを待機します。

# WebLogic Process Integrator と Tuxedo の非同期接続

WebLogic Process Integrator は、Tuxedo /Q に保証された非同期メッセージを送信します。

- JMS キューと Tuxedo /Q 間の 1 対 1 の関係。
- WebLogic Process Integrator は、メッセージを JMS キューに書き込みます。
- メッセージが JMS キューに入ると、tBridge はメッセージごとにターゲットの Tuxedo /Q にメッセージを移動します。
- 次のようなエラー メッセージは、指定した JMS エラー キューに転送されません。
  - インフラストラクチャ エラー。
  - XML/FML32 変換エラー。

## 非同期 Tuxedo /Q と WebLogic Process Integrator の接続

Tuxedo /Q は、WebLogic Process Integrator に保証された非同期メッセージを送信します。

- JMS キューと Tuxedo /Q 間の 1 対 1 の関係。
- Tuxedo は、メッセージを Tuxedo /Q に書き込みます。
- メッセージが Tuxedo /Q でコミットされると、そのメッセージは Tuxedo /T Domain Gateway を介して WebLogic Tuxedo Connector tBridge およびターゲット JMS キューに転送されます。
- Tuxedo から転送できないメッセージは、Tuxedo /Q エラー キューにエンキューされます。
- 次のようなエラーのメッセージは、指定した Tuxedo /Q エラー キューに転送されます。
  - インフラストラクチャ エラー。
  - FML32/XML 変換エラー。
- JMS キューのメッセージを待機するワークフローが作成されます。これは、既存のワークフロー インスタンスの Start ワークフロー ノードまたは Event ノードで定義されます。

## Tuxedo と WebLogic Process Integrator の双方向非同期接続

Tuxedo は、WebLogic Process Integrator プロセス フローのブロッキング呼び出しを実行します。JMS から Tuxedo /Q へ、および Tuxedo /Q から JMS へ接続するには、2 つの非同期インスタンスを使用します。

---

## 9 WebLogic Tuxedo Connector の トラブルシューティング

以下の節では、WebLogic Tuxedo Connector のトラブルシューティング情報について説明します。

- WebLogic Tuxedo Connector のモニタ
- よくある質問

### WebLogic Tuxedo Connector のモニタ

WebLogic Tuxedo Connector は、WebLogic Server ログ ファイルを使用してログ情報を記録します。ログ情報を記録するには、次の作業を行ってください。

- トレースレベルの設定
- デバッグ モードの有効化

### トレースレベルの設定

**注意：** WebLogic Server のプロパティ設定の詳細については、2-8 ページの「WebLogic Tuxedo Connector プロパティの設定方法」を参照してください。

トレースを有効にするには、サーバ起動スクリプトにある `JAVA_OPTIONS` 変数を必要なレベルに変更します。

例：

```
JAVA_OPTIONS=-Dweblogic.wtc.TraceLevel=100000
```

TraceLevel を設定するには、次の値を使用します。

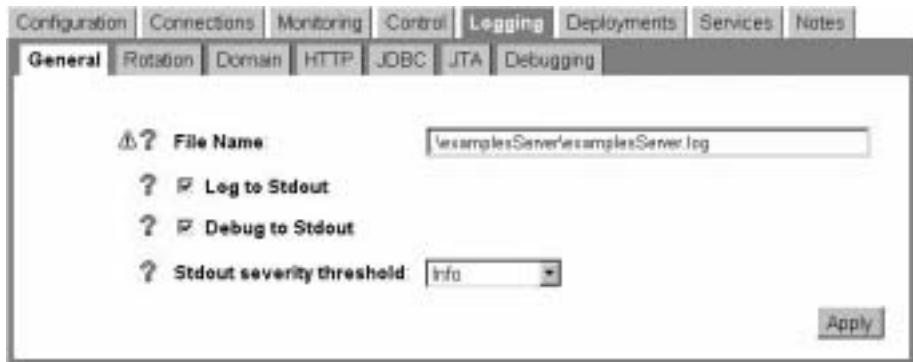
値	トレースするコンポーネント	説明
10000	TBRIDGE_IO	tBridge の入力および出力
15000	TBRIDGE_EX	詳細な tBridge 情報
20000	GWT_IO	ATMI パープなどのゲートウェイの入力と出力
25000	GWT_EX	詳細なゲートウェイ情報
50000	JAMTI_IO	低レベルの JAMTI 呼び出しなどの JAMTI の入力と出力
55000	JAMTI_EX	詳細な JAMTI 情報
60000	CORBA_IO	CORBA の入力および出力
65000	CORBA_EX	詳細な CORBA 情報
100000	すべてのコンポーネント	すべての WebLogic Tuxedo Connector コンポーネントの情報

## デバッグ モードの有効化

ログ ファイルにトレース情報を書き込む指定は、以下の手順で行います。

1. 左ペイン内で [サーバ] のノードをクリックします。
2. 左ペイン内で使用するサーバを選択します。
3. [ログ] タブを選択します。
4. [一般] タブをクリックします。
  - a. [Stdout ヘデバッグ情報出力] をクリックします。
  - b. [Stdout 重大度しきい値] を [Info] に設定します。

図 9-1 デバッグ モードの設定



## よくある質問

この節では、よくあるユーザからの質問に対する解決策について説明します。

### この EJB デプロイメント メッセージはどのような意味ですか？

simpsserv サンプルを構築すると、次のようなエラーが表示されます。

```
<date> <Error> <EJB> <EJB Deployment: Tolower has a class  
weblogic.wtc.jatmi.tpserviceHome which is in the classpath. This class should only  
be located in the ejb-jar file.>
```

このエラー メッセージは、このリリースの WebLogic Tuxedo Connector では無視できません。EJB は、EJB jar ファイルの EJB 呼び出しにすべてのインタフェースを要求します。しかし、WebLogic Tuxedo Connector のインタフェースのいくつかは CLASSPATH で実装されるため、コンパイラが例外が発生します。EJB をデプロイするとき、コンパイラはクラスのいくつかを CLASSPATH で検出されるために再デプロイできないことを示します。

## コネクタはどう起動するのですか？

コネクタの以前のリリースでは、WebLogic Tuxedo Connector セッションの開始に WebLogic Server の起動クラスを、またセッションの終了に WebLogic Server の停止クラスを使用していました。WebLogic Server 7.0 では、WebLogic Tuxedo Connector は起動または停止クラスを使用しません。WebLogic Tuxedo Connector セッションは WTCServer MBean を使用して管理されます。

- WebLogic Tuxedo Connector セッションは、コンフィグレーション済みの WTCServer MBean が、選択されたサーバに割り当てられたときに開始します。
- WebLogic Tuxedo Connector セッションは、WTCServer MBean が WebLogic Server から削除されたとき、または WebLogic Server が停止されたときに終了します。

## tBridge はどのように起動するのですか？

tBridge は、WTCServer MBean の WTCTBridgeGlobal MBean および WTCTBridgeRedirect MBean がコンフィグレーションされ、WTCServer MBean が選択したサーバに割り当てられると起動されます。

## WTCServer MBean をサーバに割り当てるにはどのようにするのですか？

WTCServer MBean をサーバに割り当てようとするとき、コンソールに例外メッセージが表示されます。どうしたらよいでしょうか？

コンフィグレーション済みの有効な WTCServer MBean があることを確認します。サーバに割り当てる前に、各 WTCServer MBean には、コンフィグレーション済みのローカル WLS ドメインが 1 つ以上必要です。サーバのログに、次のように表示されます。

```
<Apr 22, 2002 4:21:35 PM EDT> <Error> <WTC> <180101> <At least one local domain has to be defined.>
```

## 接続の問題はどのように解決するのですか？

WebLogic Tuxedo Connector と Tuxedo 間で接続の確立中に問題が発生します。どうしたらよいのでしょうか？

- Tuxedo サーバが起動されていることを確認します。
- TraceLevel を設定し、デバッグ モードを有効にします。接続テストを再度行い、WebLogic Tuxedo Connector と Tuxedo ログ ファイルにエラー メッセージがないかチェックします。
- マシン名や localhost の使用は避けます。ネットワークの位置を指定する場合には、必ず IP アドレスを使用します。
- AclPolicy 属性と CredentialPolicy 属性をチェックします。AclPolicy が LOCAL なら、リモート ドメイン DOMAINID を WebLogic Server ユーザとして登録する必要があります。詳細については、3-11 ページの「ユーザ認証」を参照してください。
- WebLogic Server 6.x からの移行で、使用していたアプリケーションでセキュリティ機能を使用していた場合、PasswordKey を WebLogic Server プロパティとして設定する必要があります。詳細については、2-8 ページの「WebLogic Tuxedo Connector プロパティの設定方法」を参照してください。
- Tuxedo リモート ドメインに対する WebLogic Tuxedo Connector コンフィグレーションをチェックします。リモート ドメインは、WebLogic Tuxedo Connector でコンフィグレーションされたリモート ドメインの名前と一致している必要があります。

例：Tuxedo DMCONFIG \*DM\_LOCAL\_DOMAINS セクションで simpapp という名前がコンフィグレーションされている場合、この名前は WTCRemoteTuxDom MBean の AccessPointId 属性にある名前と一致している必要があります。
- BEA カスタマ サポートに問い合わせます。

## 旧リリースからの移行はどのように行いますか？

WebLogic Tuxedo Connector 6.x アプリケーション (WebLogic Tuxedo Connector 1.0 も含めて) を、WebLogic Server 7.0 で使用するには、何箇所か変更しなければなりません。詳細については、「WebLogic Server 6.x からバージョン 7.0 へのアップグレード」を参照してください。



---

# 10 WebLogic Tuxedo Connector MBean 属性

MBean を使用して、WebLogic Tuxedo Connector 内のドメインをコンフィグレーションするための情報を提供します。コンフィグレーション属性は、Tuxedo ドメインで使用される DM\_MIB 属性とクラスに類似しています。以下の節には、WebLogic Tuxedo Connector のコンフィグレーションに使用される MBean のリファレンス情報があります。

- WTCServerMBean
- WTCLocalTuxDomMBean
- WTCRemoteTuxDomMBean
- WTCEXportMBean
- WTCImportMbean
- WTCPasswordMbean
- WTCResourcesMBean
- WTCtBridgeGlobalMBean
- WTCtBridgeRedirectMBean

## WTCServerMBean

WebLogic Server と Tuxedo との接続に必要な相互運用属性を含む WebLogic Tuxedo Connector の親 MBean です。

属性	説明
DeploymentOrder	デプロイメント順を指定する。順序付けは、EJB のクラスや Web アプリケーションクラスなど、同一クラス内のデプロイ可能なユニットと比較される。 DeploymentOrder の最も低いものが、一番先にデプロイされる。
Exports	WTCEXportMBean を使用して、ローカル ドメインによってエクスポートされるサービスに関する情報を提供する。
Imports	WTCTImportMBean を使用して、ローカル ドメインによってインポートされるリモート サービスに関する情報を提供する。
LocalTuxDoms	WTCLocalTuxDomMBean を使用して、リモート ドメインで認識されるローカル ドメインの情報を提供する。
Name	WTCTServer MBean の名前。
Passwords	WTCTPasswordMBean を使用して、相互ドメイン認証に関するコンフィグレーション情報を提供する。
RemoteTuxDoms	WTCTRemoteTuxDomMBean を使用して、ローカル ドメインで認識されるリモート ドメインの情報を提供する。
Resource	WTCTResourcesMBean を使用して、ドメインのグローバル フィールドテーブルクラス、ビューテーブルクラス、アプリケーション パスワードに関する情報を提供する。
Targets	WTCTServer MBean がデプロイされる選択した対象サーバの名前。
Type	WebLogic Server の MBean のタイプ : WTCTServer
tBridgeGlobal	WTCTtBridgeGlobalMBean を使用して、WebLogic Server と Tuxedo の間のメッセージ転送に関するグローバル コンフィグレーション情報を指定する。

属性	説明
tBridgeRedirects	WTCTBridgeRedirectMBean を使用して、WebLogic Server と Tuxedo の間のメッセージのソース、ターゲット、方向および転送を指定する。

## WTCLocalTuxDomMBean

WTCLocalTuxDom MBean は、他のドメインで認識されるローカルドメインのビューを提供します。以下の属性で、MBean タイプ WTCLocalTuxDom を記述します。

属性	説明
AccessPoint	<p>必須。WTCTServer MBean でドメインを識別するために使用する名前。この名前は WTCTServer 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。</p> <p>例: TDOM2</p>
AccessPointId	<p>必須。別のドメインへの接続を確立するときに、ドメインを識別するための接続プリンシパル名を指定する。</p> <p>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。</p> <p>例: TDOM2</p>

属性	説明
BlockTime	<p>省略可能。ブロッキング呼び出しに許可される最大待機時間 (単位: 秒) を指定する。</p> <ul style="list-style-type: none"> <li>■ 最小値: 0</li> <li>■ 最大値: 2147483647</li> <li>■ デフォルト値: 60</li> </ul>
CmpLimit	<p>省略可能。リモート ドメインへのデータ送信時に使用する圧縮しきい値を指定する。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> <ul style="list-style-type: none"> <li>■ 最小値: 0</li> <li>■ 最大値: 2147483647</li> <li>■ デフォルト値: 2,147,483,647 バイト</li> </ul>
ConnPrincipalName	<p><b>注意:</b> ConnPrincipalName は、このリリースの WebLogic Server ではサポートされていない。</p> <p>省略可能。接続プリンシパル名の識別子を指定する。これは、別のドメインへの接続を確立するときに、ドメインを識別するためのプリンシパル名。</p> <ul style="list-style-type: none"> <li>■ 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。</li> <li>■ この要素を指定しない場合、接続プリンシパル名はこのドメインの AccessPointID 要素にデフォルト設定される。</li> </ul>

---

属性	説明
ConnectionPolicy	<p data-bbox="575 256 1166 315">省略可能。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指定する。</p> <p data-bbox="575 331 1166 389">ローカル ドメインに対する有効値は、ON_DEMAND、ON_STARTUP、または INCOMING_ONLY。</p> <ul style="list-style-type: none"><li data-bbox="575 415 982 441">■ デフォルト設定は ON_DEMAND。</li><li data-bbox="575 467 1174 584">■ 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。</li><li data-bbox="575 610 1180 1019">■ 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 60 秒ごとに失敗した接続を再試行する。アプリケーション固有の値を指定するには、MaxRetry および RetryInterval 要素を使用する。</li><li data-bbox="575 1045 1180 1328">■ 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時接続に使用可能で、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時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。</li></ul>

---

属性	説明
Interoperate	<p><b>注意：</b> Tuxedo 6.5 は、セキュリティ マッピングをサポートするために必要なセキュリティ インフラストラクチャを持っていない。セキュリティ機能が必要で、かつ WebLogic Tuxedo Connector を使用する場合は、Tuxedo 7.1 以上にアップグレードする必要がある。</p> <p>省略可能。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。このパラメータの有効な値は、Yes または No。</p> <ul style="list-style-type: none"><li>■ Yes : Tuxedo 6.5 と相互運用する。</li><li>■ No : Tuxedo 7.1 以降のドメインで機能する。</li><li>■ デフォルト値 : No</li></ul>
MaxEncryptBits	<p><b>注意：</b> MaxEncryptBits 値 40 は、BEA Tuxedo 7.1 以降のソフトウェアを実行しているドメインにのみ適用される。</p> <p>省略可能。このドメインのネットワーク リンクの確立時に使用する最大レベルの暗号キー長 (単位: ビット) を指定する。このパラメータの有効な値は、0、40、56、および 128。</p> <ul style="list-style-type: none"><li>■ デフォルト値 : 128 ビット</li><li>■ 値 0 : 暗号化なし</li></ul>

属性	説明
MaxRetries	<p>省略可能。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。<b>ConnectionPolicy</b> が <b>ON_STARTUP</b> に設定されているときだけ使用する。</p> <ul style="list-style-type: none"> <li>■ 最小値: 0</li> <li>■ 最大値: 2147483647</li> <li>■ デフォルト値: 2147483647</li> <li>■ <b>MaxEncryptBits</b> 属性の値は、<b>MinEncryptBits</b> 属性の値と同じかそれ以上でなければならない。</li> </ul> <p>最大値は、接続が確立されるまで処理を再試行する場合に使用する。最小値は、自動再試行メカニズムを無効にする場合に使用する。</p>
MinEncryptBits	<p><b>注意:</b> <b>MinEncryptBits</b> 値 40 は、BEA Tuxedo 7.1 以降のソフトウェアを実行しているドメインにのみ適用される。</p> <p>省略可能。このドメインのネットワーク リンクの確立時に使用する最小レベルの暗号キー長(単位:ビット)を指定する。このパラメータの有効な値は、0、40、56、および128。</p> <ul style="list-style-type: none"> <li>■ デフォルト値: 0 ビット</li> <li>■ 値 0: 暗号化なし</li> <li>■ <b>MinEncryptBits</b> 属性の値は、<b>MaxEncryptBits</b> 属性の値と同じかそれ以下でなければならない。</li> <li>■ この暗号の最小レベルが一致しない場合、ネットワーク リンクは失敗する。</li> </ul>

属性	説明
NWAddr	<p>必須。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。次のフォーマットのうちの 1 つを使用して、TCP/IP アドレスを指定する。</p> <ul style="list-style-type: none"> <li>■ //hostname:port_number</li> <li>■ //#.##.##.##:port_number</li> </ul> <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 DNS) を使用して、ホスト名のアドレスを検索する。ドットで区切った数値のフォーマットを使用する場合、各 # は 0 ~ 255 までの数値にする。このドットで区切った数値は、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><b>注意:</b> T_DM_LOCAL_DOMAIN の NWAddr をコンフィグレーションする場合、使用するポート番号は、他の WebLogic Server プロセスに割り当てられたポート番号とは異なる必要がある。例: WebLogic Server のリスンポートが //mymachine:7001 に割り当てられている場合、NWAddr を //mymachine:7001 に設定すると無効になる。</p>
Name	WTCLocalTuxDom MBean の名前。
RetryInterval	<p>省略可能。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行われる接続の時間間隔 (単位: 秒)。ConnectionPolicy が ON_STARTUP に設定されているときだけ使用する。</p> <ul style="list-style-type: none"> <li>■ 最小値: 0</li> <li>■ 最大値: 2147483647</li> <li>■ デフォルト設定: 60</li> </ul>



属性	説明
Security	<p><b>注意:</b> Tuxedo 6.5 ユーザはセキュリティ パラメータを NONE に設定する必要がある。</p> <p>省略可能。実行するアプリケーションセキュリティのタイプを指定する。このパラメータの有効な値は、NONE、APP_PW、または DM_PW。NONE がデフォルト値。</p> <ul style="list-style-type: none"> <li>■ NONE : セキュリティは使用されない。</li> <li>■ 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。WTCResources MBean にアプリケーション パスワードが定義される。</li> <li>■ 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。WTCPassword MBean にドメインパスワードが定義される。</li> </ul>
Type	WebLogic Server の MBean のタイプ : WTCLocalTuxDom

## WTCRemoteTuxDomMBean

WTCRemoteTuxDom MBean は、ローカル ドメインで認識されるリモート ドメインのビューを提供します。以下の属性で、MBean タイプ WTCRemoteTuxDom を記述します。

属性	説明
AccessPoint	<p>必須。WTCServer MBean でドメインを識別するために使用する名前。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。</p> <p>例 : TDOM2</p>

属性	説明
AccessPointId	<p>必須。別のドメインへの接続を確立するときに、ドメインを識別するための接続プリンシパル名を指定する。</p> <p>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。</p> <p>例 : TDOM2</p>
AclPolicy	<p><b>注意 :</b> Interoperate パラメータが Yes に設定されている場合、AclPolicy は無視される。詳細については、10-9 ページの「WTCRemoteTuxDomMBean」を参照。</p> <p>省略可能。リモート ドメインからのリクエストに対して、着信アクセス制御リスト (ACL) ポリシーを指定する。このパラメータの有効な値は、LOCAL または GLOBAL。</p> <ul style="list-style-type: none"><li>■ <b>LOCAL :</b> 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。</li><li>■ <b>GLOBAL :</b> 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。</li><li>■ <b>デフォルト値 :</b> LOCAL</li></ul>
CmpLimit	<p>省略可能。リモート ドメインへのデータ送信時に使用する圧縮しきい値を指定する。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> <ul style="list-style-type: none"><li>■ <b>最小値 :</b> 0</li><li>■ <b>最大値 :</b> 2147483647</li><li>■ <b>デフォルト値 :</b> 2,147,483,647 バイト</li></ul>

---

属性	説明
ConnPrincipalName	<p data-bbox="575 253 1184 313"><b>注意：</b> ConnPrincipalName は、このリリースの WebLogic Server ではサポートされていない。</p> <p data-bbox="575 326 1184 410">省略可能。接続プリンシパル名の識別子を指定する。これは、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プリンシパル名。</p> <ul data-bbox="575 435 1184 634" style="list-style-type: none"><li data-bbox="575 435 1184 524">■ 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 <b>TDOMAIN</b> のドメインにのみ適用される。</li><li data-bbox="575 548 1184 634">■ この要素を指定しない場合、接続プリンシパル名はこのドメインの <b>AccessPoint</b> 要素にデフォルト設定される。</li></ul>

---

---

属性	説明
ConnectionPolicy	<p data-bbox="649 253 1237 315">省略可能。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指定する。</p> <p data-bbox="649 329 1231 386">ローカル ドメインに対する有効値は、ON_DEMAND、ON_STARTUP、または INCOMING_ONLY。</p> <ul style="list-style-type: none"><li data-bbox="649 415 1053 438">■ デフォルト設定は ON_DEMAND。</li><li data-bbox="649 467 1247 586">■ 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。</li><li data-bbox="649 610 1251 1019">■ 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 60 秒ごとに失敗した接続を再試行する。アプリケーション固有の値を指定するには、MaxRetry および RetryInterval 要素を使用する。</li><li data-bbox="649 1044 1251 1328">■ 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時接続に使用可能で、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時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。</li></ul>

---

属性	説明
CredentialPolicy	<p><b>注意:</b> Interoperate パラメータが Yes に設定されている場合、CredentialPolicy は無視される。詳細については、10-9 ページの「WTCRemoteTuxDomMBean」を参照。</p> <p>省略可能。リモート ドメインへのリクエストに対して、発信アクセス制御リスト (ACL) ポリシーを指定する。このパラメータの有効な値は、LOCAL または GLOBAL。</p> <ul style="list-style-type: none"> <li>■ 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。</li> <li>■ 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。</li> <li>■ デフォルト値: LOCAL</li> </ul>
FederationName	<p>外部ネーム サービスと結合するコンテキスト。省略すると、結合ポイントは <i>tuxedo.domains</i> になる。</p>
FederationURL	<p>JNDI に結合される外部ネーム サービスの URL。</p> <ul style="list-style-type: none"> <li>■ 省略すると、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、TGIIOP を使用して CosNaming サーバと結合する。CORBA 以外のサービス プロバイダとの結合が可能。</li> </ul>
LocalAccessPoint	<p>必須。リモート ドメインがアクセスするローカル ドメイン名。</p> <p>例: TDOM2</p>

属性	説明
MaxEncryptBits	<p><b>注意：</b> MaxEncryptBits 値 40 は、BEA Tuxedo 7.1 以降のソフトウェアを実行しているドメインにのみ適用される。</p> <p>省略可能。このドメインのネットワーク リンクの確立時に使用する最大レベルの暗号キー長（単位：ビット）を指定する。このパラメータの有効な値は、0、40、56、および 128。</p> <ul style="list-style-type: none"> <li>■ デフォルト値：128 ビット</li> <li>■ 値 0：暗号化なし</li> <li>■ MaxEncryptBits 属性の値は、MinEncryptBits 属性の値と同じかそれ以上でなければならない。</li> </ul>
MaxRetries	<p>省略可能。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。ConnectionPolicy が ON_STARTUP に設定されているときだけ使用する。</p> <ul style="list-style-type: none"> <li>■ 最小値：0</li> <li>■ 最大値：2147483647</li> <li>■ デフォルト値：2147483647</li> <li>■ ローカル WLS Domain のドメイン値に対するデフォルトとして 1 を使用。</li> </ul> <p>最大値は、接続が確立されるまで処理を再試行する場合に使用する。最小値は、自動再試行メカニズムを無効にする場合に使用する。</p>

属性	説明
MinEncryptBits	<p><b>注意:</b> MinEncryptBits 値 40 は、BEA Tuxedo 7.1 以降のソフトウェアを実行しているドメインにのみ適用される。</p> <p>省略可能。このドメインのネットワーク リンクの確立時に使用する最小レベルの暗号キー長 (単位: ビット) を指定する。このパラメータの有効な値は、0、40、56、および 128。</p> <ul style="list-style-type: none"> <li>■ デフォルト値: 0 ビット</li> <li>■ 値 0: 暗号化なし</li> <li>■ MinEncryptBits 属性の値は、MaxEncryptBits 属性の値と同じかそれ以下でなければならない。</li> <li>■ この暗号の最小レベルが一致しない場合、ネットワーク リンクは失敗する。</li> </ul>
NWAddr	<p>必須。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。次のフォーマットのうちの 1 つを使用して、TCP/IP アドレスを指定する。</p> <ul style="list-style-type: none"> <li>■ //hostname:port_number</li> <li>■ //#.##.##:port_number</li> </ul> <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 DNS) を使用して、ホスト名のアドレスを検索する。ドットで区切った数値のフォーマットを使用する場合、各 # は 0 ~ 255 までの数値にする。このドットで区切った数値は、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>
Name	WTCRemoteTuxDom MBean の名前。

属性	説明
RetryInterval	<p>省略可能。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行われる接続の時間間隔 (単位: 秒)。ConnectionPolicy が ON_STARTUP に設定されているときだけ使用する。</p> <ul style="list-style-type: none"> <li>■ 最小値: 0</li> <li>■ 最大値: 2147483647</li> <li>■ デフォルト設定: 60</li> <li>■ ローカル WLS Domain のドメイン値に対するデフォルトとして 1 を使用。</li> </ul>
TpUsrFile	<p>省略可能。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。これは、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。正しい認可、認証、および監査を行うために、このファイルにはユーザ名、uid および gid 情報が含まれ、有効になっている必要がある。</p>
Type	<p>WebLogic Server の MBean のタイプ: WTCRemoteTuxDom</p>

## WTCEXportMBean

WTCEXportMBean は、ローカル ドメインによってエクスポートされるサービスに関する情報を提供します。

- 指定しない場合、すべてのローカル ドメインは、デフォルト JNDI ルックアップ ルールに従ってすべてのサービスへのリクエストを受け付けます。
- セクションを定義する場合は、それをリモート ドメインからリクエストされたローカル サービスのセットを制限するために使用します。



属性	説明
EJBName	省略可能。サービスの呼び出し時に使用する EJB ホームインタフェースの完全な名前。 この要素を指定しない場合、使用されるデフォルトインタフェースは <code>tuxedo.services.servicenameHome</code> となる。 例：呼び出されるサービスが <code>TOUPPER</code> で、EJBName 属性を指定しない場合、JNDI でロ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。
LocalAccessPoint	必須。ローカルアクセス ポイント名。例：TDOM2
Name	WTCEXport MBean の名前。
RemoteName	省略可能。サービスのリモート名。 指定しない場合、ResourceName 属性が使用される。
ResourceName	ResourceName 属性は、エクスポートされたサービス エントリを記述する。
Type	WebLogic Server の MBean のタイプ：WTCEXport

## WTCImportMbean

WTCImportMBean は、インポートされたサービスおよびリモート ドメインで使用可能なサービスに関する情報を提供します。

属性	説明
LocalAccessPoint	サービスが提供されるローカルアクセス ポイントを指定する。例：TDOM2
Name	WTCImport MBean の名前。

属性	説明
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、カンマで区切られたフェイルオーバーリスト。 例 : TDOM3,TDOM4,TDOM5
RemoteName	このリソースの RemoteName。RemoteName とは、サービスのリモート名。指定しない場合、ResourceName 属性がデフォルトとなる。
ResourceName	ResourceName 属性は、インポートされたサービス エントリを記述する。 ResourceName、LocalAccessPoint、および RemoteAccessPointList 属性の組み合わせは、このタイプのすべてのオブジェクト間でユニークでなければならない。 例 : //simpapp
Type	WebLogic Server の MBean のタイプ : WTCImport

## WTCPasswordMbean

WTCPasswordMBean は、タイプ TDOMAIN のアクセス ポイントの相互ドメイン認証に関する情報を提供します。

属性	説明
LocalAccessPoint	パスワードが適用されるローカル ドメイン アクセス ポイントの名前。 例 : TDOM2

属性	説明
LocalPassword	genpasswd ユーティリティから返される暗号化されたローカルパスワード。このパスワードは、 <b>LocalAccessPoint</b> 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、 <b>RemoteAccessPoint</b> 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。
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。
Name	WTCPasswd MBean の名前。
RemoteAccessPoint	パスワードが適用されるリモートドメインアクセスポイントの名前。 例: TDOM3
RemotePassword	genpasswd ユーティリティから返される暗号化されたリモートパスワード。このパスワードは、 <b>LocalAccessPoint</b> 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、 <b>RemoteAccessPoint</b> 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。
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。
Type	WebLogic Server の MBean のタイプ: WTCPasswords

## WTCResourcesMBean

ドメインのグローバルフィールドテーブルクラス、ビューテーブルクラスおよびアプリケーションパスワードを指定するために使用します。

属性	説明
AppPassword	このリソースのアプリケーションパスワード。 <b>genpasswd</b> ユーティリティから返されたままのアプリケーションパスワード。この <b>Tuxedo</b> アプリケーションパスワードは、暗号化されたパスワードで、接続の認証に使われる。
AppPassword IV	グローバ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。 <b>AppPassword</b> を使って <b>genpasswd</b> ユーティリティから返されたもの。
FldTbl classes	クラスローダーによってロードされ、 <b>FldTbl</b> 配列に追加される <b>FldTbl16</b> クラスの名前。使用されるクラス名は、必要なクラスの完全修飾名。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。
FldTbl32 classes	クラスローダーによってロードされ、 <b>FldTbl</b> 配列に追加される <b>FldTbl32</b> クラスの名前。使用されるクラス名は、必要なクラスの完全修飾名。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。
Name	<b>WTCResources MBean</b> の名前。
Type	<b>WebLogic Server</b> の <b>MBean</b> のタイプ : <b>WTCResources</b>
ViewTbl classes	クラスローダーによってロードされ、 <b>ViewTbl</b> 配列に追加される <b>ViewTbl</b> クラスの名前。使用されるクラス名は、必要なクラスの完全修飾名。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。
ViewTbl32 classes	クラスローダーによってロードされ、 <b>ViewTbl32</b> 配列に追加される <b>ViewTbl32</b> クラスの名前。使用されるクラス名は、必要なクラスの完全修飾名。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。

---

# WTCtBridgeGlobalMBean

**注意：** tBridge は、定義された各リダイレクションの新しいスレッドを起動することによって、1つまたは複数のリダイレクションを処理します。最低でも1つのリダイレクションを指定しないと、tBridge は失敗し、エラーログが記録されます。

WTCtBridgeMBean は、WebLogic Server と Tuxedo の間のメッセージ転送に関するグローバル コンフィグレーション情報を提供します。

---

属性	説明
allowNonStandardTypes	<p>省略可能。非標準データ型が tBridge の通過を許可されている場合に指定する。このパラメータの有効な値は、NO または YES。</p> <ul style="list-style-type: none"><li>■ NO : 非標準データ型は拒否され、指定したエラー位置に配置される。</li><li>■ YES : 非標準データ型は、元のタイプを示すタグと共に、BLOB としてターゲット位置に配置される。</li></ul> <p>標準データ型は次のとおり。</p> <ul style="list-style-type: none"><li>■ ASCII テキスト (TextMessage、STRING)</li><li>■ BLOB (BytesMessage、CARRAY)</li></ul>

---

属性	説明
defaultReplyDeliveryMode	<p><b>注意：</b> defaultReplyDeliveryMode が指定されていない場合、または 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> <p>省略可能。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られている応答配信モードを指定する。</p> <p>JMS_BEA_TuxGtway_Tuxedo_ReplyDeliveryMode がメッセージに設定されていない場合、JMS から Tuxedo/Q にリダイレクトされるメッセージに対して、この要素を使用する。このパラメータの有効な値は、PERSIST、NONPERSIST、または DEFAULT。</p>
deliveryModeOverride	<p><b>注意：</b> deliveryModeOverride を指定しない場合、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> <p>省略可能。ターゲット位置にメッセージを配置するときに使用する配信モードを指定する。</p> <p>deliveryModeOverride 値は、メッセージに関連付けられている任意の配信モードをオーバーライドする。このパラメータの有効な値は、PERSIST または NONPERSIST。</p>
jmsFactory	<p>必須。JMS 接続ファクトリの名前。</p> <p>例: <code>weblogic.jms.ConnectionFactory</code></p>
JmstoTuxPriorityMap	<p>必須。JMS から Tuxedo /Q メッセージ優先度への優先度マッピング方向を指定するのに使用する。</p>
JndiFactory	<p>必須。JNDI ルックアップ ファクトリの名前。</p> <p>例: <code>weblogic.jndi.WLInitialContextFactory</code></p>
Name	<p>WTCTBridgeGlobal MBean の名前。</p>

属性	説明
retries	<p>省略可能。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指定する。</p> <p>デフォルト値は 0。</p> <p>値は 0 または正の整数でなければならない。</p>
retryDelay	<p><code>retryDelay</code> 中、メッセージを処理するスレッドは他のメッセージをリダイレクトできない。</p> <p>省略可能。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 (単位: 秒) を指定する。</p> <p>デフォルト値は 10。</p> <p>値は正の整数でなければならない。</p>
timeout	<p>ソース位置からメッセージを検索するのに使用するトランザクションのタイムアウトは、タイムアウトよりも長くなる。このパラメータは、タイムアウトの効果的な長さをリダイレクション全体に反映する。</p> <p>省略可能。ターゲット位置にメッセージを配置するために使用するトランザクションタイムアウト値 (単位: 秒) を指定する。<code>transactional</code> パラメータが <b>YES</b> に設定された場合は必須。</p> <p>デフォルト値は 60。</p> <p>値は 0 または正の整数でなければならない。</p> <p>0 は無限に待機することを示す。</p>
transactional	<p><b>注意：</b> この要素は、現在サポートされていない。</p> <p>省略可能。ソース位置からのメッセージの検索時およびターゲット位置へのメッセージの配置時に、トランザクションの使用を指定する。有効なパラメータ値は、<b>YES</b> または <b>NO</b>。</p> <p><b>YES：</b> トランザクションは、両方の処理で使用される。</p> <p><b>NO：</b> トランザクションは、いずれの処理でも使用されない。</p>

属性	説明
tuxErrorQueue	<p>省略可能。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。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> <p>tuxErrorDestination を指定しない場合、リダイレクトできないメッセージはすべて失われる。</p> <p>何らかの理由により 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>
tuxFactory	<p>必須。Tuxedo 接続ファクトリの名前。</p> <p>例: <code>tuxedo.services.TuxedoConnection</code></p>
TuxtoJmsPriorityMap	<p>必須。Tuxedo /Q から JMS メッセージ優先度への優先度マッピング方向を指定するのに使用する。</p>
Type	<p>WebLogic Server の MBean のタイプ: WTCTBridgeGlobal</p>
userID	<p>省略可能。セキュリティ オプションがコンフィグレーションされている場合、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 を指定する。</p> <p>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。</p> <p>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>



属性	説明
wlsErrorDestination	<p>省略可能。メッセージが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する場所の名前。</p> <p>wlsErrorDestination を指定しない場合、リダイレクトできないメッセージはすべて失われる。</p> <p>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>

## WTcBridgeRedirectMBean

ソース、ターゲット、方向、およびメッセージの転送を指定するために使用されます。

属性	説明
direction	<p>必須。データ フローの方向を指定する。有効なパラメータ値は、JmsQ2TuxQ、TuxQ2JmsQ、JmsQ2TuxS。</p> <ul style="list-style-type: none"> <li>■ JmsQ2TuxQ : JMS から TUXEDO /Q</li> <li>■ TuxQ2JmsQ : TUXEDO /Q から JMS</li> <li>■ JmsQ2TuxS : JMS から JMS への Tuxedo サービス応答</li> </ul>
metadataFile	<p><b>注意：</b> この要素は、現在サポートされていない。</p> <p>省略可能。メタデータ ファイルの URL。</p>
Name	WTcBridgeRedirect MBean の名前。
ReplyQ	<p>省略可能。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。応答は JMS ReplyQ に返される。</p>

属性	説明
SourceAccessPoint	ドメイン コンフィグレーションのローカルまたはリモートのエントリ名の範囲内でユニークな識別子。そこにソースが置かれる。
SourceName	必須。JMS キュー名、Tuxedo キュー名、または Tuxedo サービス名の名前。
SourceQspace	省略可能。ソース位置の Qspace の名前。
TargetAccessPoint	ドメイン コンフィグレーションのローカルまたはリモートのエントリ名の範囲内でユニークな識別子。そこにターゲットが置かれる。
TargetName	必須。JMS キュー名、Tuxedo キュー名、または Tuxedo サービス名の名前。
TargetQspace	省略可能。ターゲット位置の Qspace の名前。
translateFML	<p><b>注意：</b> WLXT パラメータ値は、現在サポートされていない。</p> <p>必須。XML/FML 変換のタイプを指定する。有効なパラメータ値は、NONE、FLAT、または WLXT。</p> <ul style="list-style-type: none"> <li>■ 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、リダイレクションは失敗する。</li> <li>■ FLAT: メッセージペイロードは、WebLogic Tuxedo Connector の組み込みトランスレータを使用して変換される。</li> <li>■ WLXT: 変換は、XML-to-nonXML WL XML Translator (WLXT) によって実行される。指定された metadataFile URL は、変換を実行する WLXT 外部メソッドを呼び出すために渡される。</li> <li>■ デフォルト値: NO</li> </ul>
Type	WebLogic Server の MBean のタイプ: WTCTBridgeRedirect