

# WebOTX 運用編(コンフィグレーション)

WebOTX 運用編

バージョン: 7.1

版数: 第9版

リリース: 2011年2月

Copyright (C) 1998 - 2011 NEC Corporation. All rights reserved.

# 目次

1. はじめに.....	1
2. 各サービスの属性の設定.....	2
2.1. サービス一覧.....	2
2.2. 統合運用管理ツールを利用時の設定.....	2
2.3. 運用管理コマンドを利用時の設定.....	2
2.4. JMX APIを利用した設定.....	2
3. エージェントに関する設定.....	3
3.1. エージェント設定項目・設定方法.....	3
3.2. エージェント設定項目一覧.....	3
4. HTTPサーバに関する設定.....	10
4.1. HTTPサーバ設定方法.....	10
4.2. HTTPサーバに関する設定.....	10
4.3. Javaベース内蔵Webサーバ設定方法.....	12
4.3.1. SSL通信の設定方法.....	12
4.4. 外部Webサーバ設定方法.....	13
5. Webコンテナに関する設定.....	14
5.1. Webコンテナ設定項目・設定方法.....	14
5.2. Webコンテナ設定項目一覧.....	14
5.3. HTTPのみのセッションレプリケーション.....	23
5.3.1. HTTPのみのセッションレプリケーションの設定方法.....	23
5.4. Webアプリケーションのコンテキスト名を動的反映しない設定方法.....	24
5.4.1. 動的反映しない設定方法.....	24
5.5. Webサーバプラグイン設定項目一覧.....	25
5.6. IIOPプラグイン定義ファイルの設定方法.....	38
5.6.1. IIOPプラグイン定義ファイルの設定手順.....	38
5.6.2. IIOPプラグインの設定項目一覧.....	38
5.6.3. IIOPプラグインの設定例.....	43
6. EJBコンテナに関する設定.....	45
6.1. EJBコンテナ設定項目・設定方法.....	45
6.2. EJBコンテナ設定項目一覧.....	45
6.3. タイマーBeanの使用方法について.....	47
6.4. リソースのプロセス単位のロード設定(Standard/Enterprise Editionの場合).....	47
7. JNDIサービスに関する設定.....	49
7.1. JNDI設定項目・設定方法.....	49
7.2. JNDIサービス設定項目一覧.....	49
7.3. JNDIサーバの通信リスナの設定.....	50

7.4. JNDIサーバを二重化するための設定 .....	50
8. JMSに関する設定 .....	52
8.1. JMS設定項目・設定方法 .....	52
8.1.1. JMSサーバに対する設定 .....	52
8.1.2. コネクションファクトリに対する設定 .....	53
8.1.3. 送信先に対する設定 .....	53
8.2. JMS設定項目一覧 .....	53
8.2.1. MOの属性一覧 .....	53
8.2.2. JMSのプロパティ／属性一覧 .....	55
8.3. その他の設定項目・設定方法 .....	69
8.3.1. ファイルストアを利用するための設定 .....	69
8.3.2. JDBCストアを利用するための設定 .....	69
8.3.3. ユーザ管理の設定 .....	71
8.3.4. アクセス制御機能を利用するための設定 .....	72
8.3.5. SSL通信を利用するための設定 .....	74
8.3.6. パスワードファイルの使用 .....	75
8.3.7. Standard/Enterprise Editionで動作させる場合の設定 .....	75
8.3.8. ユーザ管理方式の設定 .....	77
8.3.9. メッセージ再配信に関する設定 .....	77
8.3.10. 破棄メッセージの転送に関する設定 .....	79
8.3.11. JMSクライアントのログ出力 .....	80
8.3.12. 分散トランザクションへの参加 .....	80
8.3.13. JMSサーバクラスタを利用するための設定 .....	80
8.3.14. 接続先分散のための設定 .....	82
8.3.15. 再配信メッセージの順序保証のための設定 .....	82
9. JDBCデータソースに関する設定 .....	84
9.1. JDBCデータソース設定項目・設定方法 .....	84
9.2. JDBCデータソース設定項目一覧 .....	84
9.3. Oracleでの設定内容 .....	90
9.4. DB2 での設定内容 .....	91
9.5. Microsoft SQL Serverでの設定内容 .....	92
9.6. Sybaseでの設定内容 .....	93
9.7. SequeLinkでの設定内容 .....	95
9.8. Cloudscapeでの設定内容 .....	96
9.9. PostgreSQLでの設定内容 .....	97
9.10. Apache Derby での設定内容 .....	98
9.11. JDBC での設定内容 .....	99
9.12. クラスパス設定 .....	100
9.13. その他の設定 .....	101
10. Transactionサービスに関する設定 .....	102
10.1. Transactionサービス全般の設定項目・設定方法 .....	102

10.2. トランザクション情報の表示 .....	102
10.3. Transactionサービスに関する設定 .....	102
10.4. リソースを管理するためのコマンド .....	105
11. Object Brokerに関する設定 .....	109
11.1. Object Broker設定項目・設定方法 .....	109
11.1.1. Object Broker設定項目一覧 .....	109
11.2. WebOTX Object Broker Java™ におけるORBのプロパティ定義 .....	120
11.3. WebOTX Object Broker C++ における環境設定 .....	133
12. TPモニタに関する設定 .....	143
12.1. エージェント設定項目・設定方法 .....	143
12.2. TPモニタ設定項目一覧 .....	143
13. OLF/TP Adapterに関する設定 .....	147
13.1. OLF/TP Adapterに関する設定 .....	147
13.2. 通信環境定義(OLFAdapter.ini) .....	147
13.3. 端末定義(OLFAdapter.xml) .....	149
13.4. 配備記述子(ra.xml) .....	151
14. Working Domain Coordinatorに関する設定 .....	154
14.1. Working Domain Coordinator全般の設定項目・設定方法 .....	154
14.2. Working Domain Coordinatorに関する設定 .....	154
14.3. ビジネスロジックグループに関する設定 .....	155
14.4. ビジネスロジックに関する設定 .....	156
14.5. ロードバランサに関する設定 .....	156
14.6. 制御対象サーバに関する設定 .....	157
14.7. 制御対象ドメインに関する設定 .....	158
15. データベースコントローラに関する設定 .....	160
15.1. データベースコントローラ全般の設定項目・設定方法 .....	160
15.2. データベースコントローラに関する設定 .....	160

# 1.はじめに

本書は WebOTX 実行環境を運用するための運用操作法について概要や具体的な設定項目や設定方法について記載していません。

## 対象読者

このマニュアルは WebOTX Application Server Web Edition、Standard-J Edition、Standard Edition、Enterprise Edition を使って運用環境を構築するシステムエンジニア、日々の運用を行うオペレータを対象としています。

## 表記について

### パス名表記

本書ではパス名の表記については特に OS を限定しない限りセパレータはスラッシュ '/' で統一しています。Windows 環境においては '¥' に置き換えてください。

### 環境変数表記

インストールディレクトリやドメインルートディレクトリなど環境によって値の異なるものについては環境変数を用いて表します。

`$(env)` または `$(env)` で表しています。

例)

`$(AS_INSTALL)`: インストールディレクトリ

`$(INSTANCE_ROOT)`: ドメインルートディレクトリ

### コマンド操作について

本書中では運用操作に用いるコマンドの詳細についての説明は省略しています。

コマンドの詳細は「運用管理コマンド」、「運用管理コマンドリファレンス」を参照してください。

## 2.各サービスの属性の設定

WebOTX の各サービスで提供している属性の設定方法について説明します。属性は MO の属性として提供しています。属性値の取得・設定を行なうには次の方法があります。

- 統合運用管理ツールより設定
- 運用管理コマンド(otxadmin)により設定
- JMX API を利用したプログラムより設定

### 2.1.サービス一覧

WebOTX で提供しているサービス一覧を以下に示します。なお各サービスの属性については各サービスの設定項目についてを参照ください。

- 運用管理エージェント
- HTTP サーバ
- Web コンテナ
- EJB コンテナ
- JNDI サービス
- JMS
- JDBC データソース
- Transaction サービス
- Object Broker
- TP モニタ
- OLF/TP Adapter
- Working Domain Coordinator

### 2.2.統合運用管理ツールを利用しての設定

WebOTXAdministratorで提供しているWebOTX統合運用管理ツールよりGUI画面で設定を行なうことができます。統合運用管理ツールの使用方法は運用編「[統合運用管理ツール](#)」を参照してください。

### 2.3.運用管理コマンドを利用しての設定

WebOTX 実行環境(Web,Standard-J,Standard,Enterprise の各 Edition)および WebOTX Administrator で提供している運用管理コマンドよりコマンドベースで設定を行なうことができます。設定方法の例については運用編「運用管理コマンド」の「運用管理コマンドの使用例」を参照してください。

### 2.4.JMX APIを利用した設定

JMX API を利用した Java プログラムより設定を行なうことができます。なお JMX を利用したプログラミングについては「WebOTX アプリケーション開発ガイド」→「第 3 部 チュートリアル」→「JMX アプリケーション」を参照ください。また各 MO については運用編「MO 定義リファレンス」を参照ください。

# 3. エージェントに関する設定

エージェント本体に関する設定について説明します。

## 3.1. エージェント設定項目・設定方法

エージェントの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

## 3.2. エージェント設定項目一覧

Dottedname : domain

属性名 (attribute-name)	説明	既定値
application-root	アプリケーション配備パスを表します	<code>\${INSTANCE_ROOT}</code> <code>/applications</code>
backup-option	バックアップの対象とするファイルを指定します。 "config" を指定した場合、 <code>\${INSTANCE_ROOT}/config</code> フォルダ、 "all" を指定した場合、 <code>\${INSTANCE_ROOT}/</code> 配下にある各種リソースファイルを対象とします。	config
num-config-backup-generations	リソースのバックアップ世代数	3
hostname	ホスト名を表します。	localhost
ipv6-enable	IPv6 を優先的に使用するか指定します。 既定値は IPv4 優先 (false) です。 jvm-option の <code>java.net.preferIPv6Addresses</code> の設定に 影響します。	false

Dottedname : server.J2EEserver

属性名 (attribute-name)	説明	既定値
name	エージェントのサーバ名 server 固定です。	server
services-alive-check-enabled	サービス内部で動作しているサービスに対するア ライブチェックの有無	true
services-alive-check-interval-in-seconds	サービスに対するアライブチェックの間隔 (秒)	10
waiting-shutdown-timeout-in-seconds	サーバシャットダウン処理を待機する時間 (秒)	60

Dottedname : server.admin-service

属性名 (attribute-name)	説明	既定値
system-jmx-connector-name	JMX コネクタ名	jmx_connector
admin-realm	管理ユーザの認証で使用するレルム名	file

Dottedname : server.admin-service.das-config

属性名 (attribute-name)	説明	既定値
admin-session-timeout-in-minutes	運用管理コンソールのタイムアウト値	60
autodeploy-dir	autodeploy 用のディレクトリ	`\${INSTANCE_ROOT}` /autodeploy
autodeploy-enabled	true のとき、autodeploy が可能になります。ドメインの再起動や個別の配備操作を行わずに、起動しているアプリケーションサーバにアプリケーションやモジュールをすばやく配備することができます。	true
autodeploy-jsp-precompilation-enabled	true のとき、JSP ページは autodeploy 中に事前コンパイルされます。	false
autodeploy-polling-interval-in-seconds	Autodeploy のポーリング間隔(秒単位)を指定します。	2
autodeploy-verifier-enabled	true のとき、autodeploy 前に確認が行われます。失敗すると配備は行われません。	false
deploy-xml-validation	配備時に配備記述子の XML 妥当性チェックの種類を設定します。  full - XML 妥当性チェックが失敗したときは配備も失敗します。  parsing - XML 妥当性チェック時のエラーは記録されますが、配備は行われます。  none - XML 妥当性チェックは行われません。	full
dynamic-reload-enabled	true の場合、動的リロードを実行するためにすべてのモジュールおよびアプリケーションディレクトリ階層の .reload ファイル上のタイムスタンプをチェックします。	true (Standard Edition、および Enterprise Edition では、false に設定されます。)
dynamic-reload-poll-interval-in-seconds	動的リロードのポーリング間隔(秒単位)を指定します。	2
deployable-space-validation	配備可能容量の検証を行うかどうかを設定します。	true



Dottedname : server.admin-service.jmx-connector.jmx\_connector

属性名 (attribute-name)	説明	既定値
access-log-enabled	アクセスログ採取の有無を指定します。 設定を反映させるにはドメインの再起動が必要です。	false
name	JMX コネクタ名。	Jmx_connector
port	JMX Remote で使用するポート番号。	各 domain の port 番号
protocol	JMX Remote で使用するプロトコル。	jmxmp

Dottedname : server.internal-lifecycle-module.service-name

属性名 (attribute-name)	説明	既定値
class-name	ライフサイクルモジュールクラス名	
dependence	依存関係のあるライフサイクルモジュール	
description	説明	
enabled	true: ドメイン起動時にサービスを起動する false: ドメイン起動時にサービスを起動しない	true
is-failure-fatal	true: サービス起動に失敗した場合はドメイン起動に失敗する false: サービス起動に失敗してもドメイン起動を行う	false
load-order	サービスの実行順番	
Name	サービス名	
State-order	モジュール(サービス)の最終起動状態(どの状態まで起動させるか)を表す	

Dottedname : server.java-config

属性名 (attribute-name)	説明	既定値
bytecode-preprocessors	コンマによって分けられた、 "com.nec.webotx.appserv.BytecodePreprocessor" インタフェースをインプリメントしたクラスのリスト。 指定された各プリプロセッサクラスは指定された順に呼ばれます。	
classpath-prefix	システムクラスパスのプレフィクスを指定します。XML パーサクラスのようなシステムクラスをオーバーライドしたいときは単にシステムクラスパスの前につけるべきです。この属性は注意して使用してください。 なお、クラスパスを複数指定する場合は、OSによって指定方法が異なります。 Windows 系 OS : <Path1>;<Path2>;...;<PathX> UNIX 系 OS : <Path1>:<Path2>:...:<PathX>	

	パスの区切り文字が“;”(セミコロン)と“:”(コロン)とで異なりますので注意して下さい。	
classpath-suffix	システムクラスパスのサフィックスを指定します。 なお、クラスパスを複数指定する場合は、OSによって指定方法が異なります。 Windows 系 OS : <Path1>;<Path2>;...;<PathX> UNIX 系 OS : <Path1>:<Path2>:...:<PathX> パスの区切り文字が“;”(セミコロン)と“:”(コロン)とで異なりますので注意して下さい。	
debug-enabled	true の場合、サーバは JPDA ベースのデバッガを利用したデバッグモードで起動します。	false
debug-options	JPDA (Java Platform Debugger Architecture)オプションを指定します。利用可能なデバッグオプションは以下にあります。 <a href="http://java.sun.com/products/jpda/doc/conninv.html#Invocation">http://java.sun.com/products/jpda/doc/conninv.html#Invocation</a>	-Xdebug -Xrunjwdp: transport=dt_socket, server=y,suspend=n , address=4004
env-classpath-ignored	False のとき、CLASSPATH 環境変数は WebOTX アプリケーションサーバクラスパスに読み込まれ追加されます。CLASSPATH 環境変数はクラスパスの最後に追加されます。 開発環境では、この値は false に設定してください。本番環境では、環境変数の副作用を防ぐため true に設定してください。	true
java-home	Java インストールディレクトリ	\${com.nec.webotx.javaRoot}
javac-options	アプリケーション配備時に Java コンパイラに渡すオプションを指定します。	-g
jvm-options	JVM オプションを指定します。	(*1)を参照してください
native-library-path-prefix	ネイティブライブラリパスのプレフィックスを指定します。 ネイティブライブラリパスはそのネイティブ共通ライブラリ、標準 JRE ネイティブライブラリパス、シェル環境変数設定 (UNIX の LD_LIBRARY_PATH)、profiler 要素に設定されたパスに対してアプリケーションサーバのインストール相対パスをもとに自動的に構築されます。パスが作られた後は、サーバ設定内で明示的に現れません。 なお、パスを複数指定する場合は、OS によって指定方法が異なります。 Windows 系 OS : <Path1>;<Path2>;...;<PathX>	

	<p>UNIX 系 OS : &lt;Path1&gt;:&lt;Path2&gt;:...:&lt;PathX&gt;</p> <p>パスの区切り文字が“;”(セミコロン)と“:”(コロン)とで異なりますので注意して下さい。</p>	
native-library-path-suffix	<p>ネイティブライブラリパスのサフィックスを指定します。</p> <p>なお、パスを複数指定する場合は、OS によって指定方法が異なります。</p> <p>Windows 系 OS : &lt;Path1&gt;;&lt;Path2&gt;;...;&lt;PathX&gt;</p> <p>UNIX 系 OS : &lt;Path1&gt;:&lt;Path2&gt;:...:&lt;PathX&gt;</p> <p>パスの区切り文字が“;”(セミコロン)と“:”(コロン)とで異なりますので注意して下さい。</p>	
rmic-command-maxlength	<p>EJB の配備の際に実行される RMI コンパイラの java コマンドラインの最大長を指定します。これは Windows の場合のみ有効です。</p> <p>rmic コンパイラにはアプリケーションに含まれる EJB の全ホーム、リモートインタフェースクラス名がパラメータとして指定されますが、インタフェース数が多い場合、コマンドラインの長さが非常に長くなり、Windows の場合に RMI コンパイラを実行する java の起動が失敗することがあります。このため rmic を起動するコマンドラインの長さがこの設定値を超える場合、rmic コンパイルするインタフェースの数を減らしてコマンドラインの長さが設定値に収まるように rmic の実行を分割して行ないます。</p>	32000
rmic-java-options	<p>EJB の配備の際に実行される RMI コンパイラの java コマンドに指定されるオプションを指定します。例えば rmic の実行が OutOfMemoryError で失敗する場合は、-Xmx オプションでヒープサイズを拡張します。</p>	
rmic-options	<p>アプリケーション配備時に RMI コンパイラに渡されるオプションを指定します。-keepgenerated オプションはスタブや tie の生成されたソースを保存します。</p> <p>rmic コマンドの詳細は以下を参照してください。</p> <p><a href="http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/rmic.html">http://java.sun.com/j2se/1.4.2/docs/tooldocs/solaris/rmic.html</a></p>	-g
rmic-enable	<p>RMIC を実行すると Dynamic RMI-IIOP が無効化されます。</p>	false
server-classpath	<p>サーバのクラスパスを指定します。</p> <p>このクラスパスは System.getProperty("java.class.path") を使用してアクセスします。</p> <p>なお、クラスパスを複数指定する場合は、OS によって指定方法が異なります。</p> <p>Windows 系 OS : &lt;Path1&gt;;&lt;Path2&gt;;...;&lt;PathX&gt;</p>	

	UNIX 系 OS : <Path1>:<Path2>:...:<PathX> パスの区切り文字が“;”(セミコロン)と“:”(コロン)とで異なりますので注意して下さい。	
--	--	--

(\*1) jvm-options の既定値

```

-server
-Xms64m
-Xmx512m
-XX:MaxPermSize=128m
-XX:NewRatio=2
-Djava.endorsed.dirs=${com.nec.webotx.installRoot}${file.separator}lib${file.separator}endorsed
-Djava.security.policy=${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}server.policy
-Djava.security.auth.login.config=${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}login.conf
-Dsun.rmi.dgc.server.gcInterval=3600000
-Dsun.rmi.dgc.client.gcInterval=3600000
-Djavax.net.ssl.keyStore=${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}keystore.jks
-Djavax.net.ssl.trustStore=${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}cacerts.jks
-Djava.ext.dirs=${com.nec.webotx.javaRoot}${file.separator}jre${file.separator}lib${file.separator}ext${path.separator}${com.nec.webotx.instanceRoot}${file.separator}lib${file.separator}ext
-Dcom.nec.webotx.enterprise.taglibs=wosv-jstl.jar,jstl-impl.jar
-Dlog4j.configuration=file:///${com.nec.webotx.instanceRoot}${file.separator}config${file.separator}log4j.xml
-Djava.net.preferIPv6Addresses=false

```

Dottedname : server.log-service

属性名 (attribute-name)	説明	既定値
log-rotation-generations	ログ(server.log)のローテーション数	3
log-service.log-rotation-limit-in-bytes	ログ(server.log)のローテーションサイズ(単位:byte)	10000000

Dottedname : server.monitoring-service.module-monitoring-levels

属性名 (attribute-name)	説明	既定値
connector-connection-pool	コネクションプールのモニタリングレベル OFF: モニタリングしない ON: モニタリングする	OFF
ejb-container	EJB コンテナのモニタリングレベル OFF: モニタリングしない LOW: 概要レベルでモニタリングする	OFF

	HIGH: 詳細レベルでモニタリングする	
http-service	HTTP サービスのモニタリングレベル OFF: モニタリングしない ON: モニタリングする	OFF
jdbc-datasource	JDBC データソースのモニタリングレベル OFF: モニタリングしない ON: モニタリングする	OFF
thread-pool	スレッドプールデータソースのモニタリングレベル OFF: モニタリングしない ON: モニタリングする	OFF
transaction-service	トランザクションサービスのモニタリングレベル OFF: モニタリングしない LOW: 概要レベルでモニタリングする HIGH: 詳細レベルでモニタリングする	OFF
web-container	Web コンテナのモニタリングレベル OFF: モニタリングしない LOW: 概要レベルでモニタリングする HIGH: 詳細レベルでモニタリングする	OFF
jms-service	jms service のモニタリングレベル OFF: モニタリングしない LOW: 概要レベルでモニタリングする HIGH: 詳細レベルでモニタリングする	OFF

## 4.HTTPサーバに関する設定

WebOTX では、利用可能な Web サーバとして、Java ベースの内蔵 Web サーバと IIS/Apache (WebOTX Web Server)/Sun ONE などの外部 Web サーバをサポートしています。インストール時のデフォルト設定は、内蔵 Web サーバを利用します。この場合、インストール後に設定する項目は特にありません。外部 Web サーバを利用する場合には、インストール後に「環境設定ツール」を起動し、外部 Web サーバとの連携設定を行う必要があります。

WebOTX V7.1 では、Java ベースの内蔵 Web サーバのほかに、Apache ベースの Web サーバ(以降「WebOTX Web サーバ」と記載します)をバンドルしています。WebOTX Web サーバは、インストール時の指定により Ver1.3 (Apache HTTP Server 1.3 ベース)と Ver2.0 (Apache HTTP Server 2.0 ベース)を選択してインストールすることができます。

ここでは、WebOTXで利用するWebサーバとして、「WebOTX Webサーバ」を利用する場合の設定方法について記載します。その他の外部Webサーバを利用する場合については、「4.4 外部Webサーバ設定方法」項目を参照してください。

WebOTX の運用管理コマンドでは、WebOTX Web サーバの定義情報の参照および Web サーバの起動/停止ができます。定義情報を設定するには、従来どおり定義ファイル (httpd.conf)をエディタで編集してください。

なお、Javaベースの内蔵Webサーバの設定については、「5 Webコンテナに関する設定」を参照してください。

### 4.1.HTTPサーバ設定方法

#### 項目の設定

HTTP サーバの定義情報を設定(変更)するには、HTTP サーバの定義情報ファイル(httpd.conf)を編集する必要があります。

**`#{INSTANCE_ROOT}/config/WebServer/httpd.conf`**

設定項目の詳細については、Apache のマニュアルまたは Web サイトを参照してください。

例えば、HTTP サーバが利用するポート番号を 80 番から 8080 番に変更する場合には、HTTP サーバの定義情報ファイル (httpd.conf)において、次の変更を行います。HTTP サーバのバージョンにより設定項目が異なる場合がありますので、注意してください。また、ドメイン生成後にポート番号を変更した場合に限り、domain.xml の設定内容も変更する必要があります。

(httpd.conf)

```
#Port 80
Port 8080
```

(domain.xml)

```
....
<webserver conf-file="xxxx" port="8080" .... />
....
```

#### SSL(HTTPS)通信用の設定について

HTTPS 通信を行うには、SSL 通信用のモジュールのインストールと、次節の MO 設定において SSL 通信を行うかどうかの設定 (security\_enabled)を、「有効」(ture)に設定する必要があります。

なお、SSL(HTTPS)通信に関する定義情報を変更する場合は、

**`#{INSTANCE_ROOT}/config/WebServer/ssl.conf`**

の設定項目を変更する必要があります。

詳細については、マニュアルの「注意制限事項」「Web サーバ」「SSL(HTTPS)通信について」を参照してください。

### 4.2.HTTPサーバに関する設定

#### MOから参照可能な定義情報項目

以下は MO から参照可能な Web サーバの定義情報項目の一覧です。

各定義情報を変更する場合には、`#{INSTANCE_ROOT}/config/WebServer/httpd.conf` または `#{INSTANCE_ROOT}/config/WebServer/ssl.conf` を直接編集してください。

Dottedname : server.WebServer

属性名 (attribute-name)	説明	httpd.conf の 指示子	既定値
serverroot	HTTP サーバが動作するルートディレクトリ 参照のみ可能です	ServerRoot	\${INSTANCE_ROOT}
conf_file	定義情報ファイル (httpd.conf) のパス 参照のみ可能です	(MO でのみ参照)	\${INSTANCE_ROOT}/config/ WebServer/httpd.conf
port	HTTP サーバが利用するポート番号 参照のみ可能です	Port/Listen	80
Version	HTTP サーバのバージョン情報 参照のみ可能です	(MO でのみ参照)	1.3.37 / 2.0.59
ServerName	HTTP サーバのサーバ名 参照のみ可能です	ServerName	localhost
DocumentRoot	ブラウザが表示するルートディレクトリ 参照のみ可能です	DocumentRoot	\${INSTANCE_ROOT}/docroot
ErrorLog	HTTP サーバのエラーログ 参照のみ可能です	ErrorLog	\${INSTANCE_ROOT}/logs/ WebServer/error.log
LogLevel	HTTP サーバのエラーログレベル 参照のみ可能です	LogLevel	Warn
Maxclients	最大同時接続クライアント数 参照のみ可能です	(Windows)	(Win-V1.3) 50
		ThreadsPerChild	(Win-V2.0) 250
		(UNIX) MaxClients	(UNIX) 150
security_enabled	SSL (HTTPS) 通信を行うかどうか	(MO でのみ設定)	false
ssl_conf_file	SSL 用の定義ファイル 参照のみ可能です	(MO でのみ参照)	\${INSTANCE_ROOT}/config/ WebServer/ssl.conf
ssl_port	SSL 用のポート 参照のみ可能です	Port/Listen (ssl.conf)	443

## 上記以外の定義情報項目

上記以外の定義情報の変更する場合には、\${INSTANCE\_ROOT}/config/WebServer/httpd.conf または  
\${INSTANCE\_ROOT}/config/WebServer/ssl.conf を直接編集してください。  
各定義情報の詳細については、次のWebサイトを参照してください。

<http://httpd.apache.org/docs/1.3/>

<http://httpd.apache.org/docs/2.0/ja/>

## Webサーバ起動待ち合わせ時間の設定

Webサーバのライブチェックモニタ処理は、Webサーバの起動処理が完了してから開始しますが、システムの状態やバンドルするWebサーバのモジュール起動タイミングによっては、Webサーバが起動しているにもかかわらず、「Webサーバが起動していない」というメッセージが出力される場合があります。これを回避するために、次の設定により、Webサーバ起動後にライブチェックモニタ処理を開始する時間を秒単位で指定します。

ここでは、Webサーバの起動が完了してから10秒後にライブチェックモニタを開始する場合の設定例を記載します。

```
otxadmin>create-jvm-options -Dwebotx.webserver.startup_wait_count=10
```

## 4.3.Javaベース内蔵Webサーバ設定方法

### 項目の設定

Javaベースの内蔵Webサーバの設定方法については、次の2通りがあります。

### Webコンテナ管理コンソールによる設定

Webコンテナの運用管理コンソールの「コネクタ」項目で、Javaベースの内蔵Webサーバの設定ができます。コネクタの「種類」が「HTTP」と表示されているものが内蔵Webサーバです。詳細は「5Webコンテナに関する設定」を参照してください。

### MOによる参照と設定

Javaベースの内蔵Webサーバの設定は、WebコンテナのMOを介してもアクセス可能です。対象となるMOのDottednameは「server.http-service.http-listener.http-listener-name」です。属性 protocol の値が「HTTP/1.1」となっているものが内蔵Webサーバです。詳細は「5Webコンテナに関する設定」を参照してください。

### 4.3.1.SSL通信の設定方法

#### 設定手順

#### ① キーストアへのサーバ証明書の追加

WebOTXのキーストアにサーバ証明書を追加します。

WebOTXのキーストアは、\${INSTANCE\_ROOT}/config/keystore.jks です。このファイルにサーバ証明書を追加します。

J2SE 5.0 Development Kit (JDK 5.0) に付属の keytool を利用した例を次に記述します。サーバ証明書は、server.cer とします。

キーエントリの追加例

```
keytool -genkey -alias server -keyalg RSA -storepass changeit -keystore keystore.jks
```

キーエントリ追加後、証明書署名要求(CSR)を作成し、証明書の発行依頼を行います。取得した証明書をインポートします。

証明書署名要求の作成例

```
keytool -certreq -alias server -file server.csr -keystore ./keystore.jks -storepass changeit
```

証明書のインポート例

```
keytool -import -v -trustcacerts -alias server -file server.cer -keystore ./keystore.jks -storepass changeit
```

#### ② https設定でのサーバ証明書の追加

httpsの処理で使用するサーバ証明書を指定します。

JavaVMのオプションの設定は、運用管理コマンド(otxadmin)を利用して、次のように行います。サーバ証明書の別名はserver とします。運用管理コマンドのユーザ名、パスワードはデフォルトでdomain1 を作成した場合の例です。

```
otxadmin>create-ssl --type http-listener --certname server --clientauthenabled=false http-listener-2
```



### ③ パスワードの変更設定

パスワードを変更した場合は、JavaVM のオプションの追加とJMXのコネクタのパスワードの変更が必要になります。  
JavaVMのオプションの設定は、運用管理コマンド(otxadmin)を利用して、次のように行います。

```
otxadmin> create-jvm-options -Djavax.net.ssl.keyStorePassword=password
```

JMXのコネクタのパスワードの変更は、\${INSTANCE\_ROOT}/config/domain.xmlの次の定義(keystore-password)を変更します。

[ domain.xml]

```
<jmx-connector ...  
  keystore-password="changeit"  
  ...  
</>
```

## 4.4.外部Webサーバ設定方法

### 項目の設定

外部 Web サーバを利用する場合は、事前に環境設定ツールを使って連携のための設定が必要です。設定方法は、「セットアップガイド」-「インストール」-「インストール後の作業」を参照してください。

外部 Web サーバの設定方法については次の2通りがあります。なお、下記は外部 Web サーバと連携するためのコネクタに関する設定です。Web サーバ自体の設定については、それぞれのマニュアルを参照してください。

### Webコンテナ管理コンソールによる設定

Webコンテナの運用管理コンソールの「コネクタ」項目で、外部のWebサーバと連携するためのコネクタの設定ができます。コネクタの「種類」が“AJP”と表示されているものが外部Webサーバ連携用のコネクタです。詳細は「5Webコンテナに関する設定」を参照してください。

### MOによる参照と設定

外部Webサーバ連携用コネクタの設定は、WebコンテナのMOを介してもアクセス可能です。対象となる MO の Dottedname は「server.http-service.http-listener.http-listener-name」です。属性 protocol の値が“AJP/1.3”となっているものが外部Webサーバ連携用コネクタです。詳細は「5Webコンテナに関する設定」を参照してください。

# 5.Webコンテナに関する設定

Web コンテナに関する設定について説明します。

## 5.1.Webコンテナ設定項目・設定方法

Web コンテナの設定項目の設定方法については次の 2 通りがあります。

### Webコンテナ管理コンソールによる設定

Web コンテナの運用管理コンソールでは、情報の設定ができます。なお Web コンテナ運用管理コンソールの使い方と詳細については、運用編「運用管理コンソール」を参照してください。

### MOによる参照と設定

Webコンテナの運用管理コンソールで設定できる項目は、JMXのMOとしてアクセス可能です。項目の設定方法については「2各サービスの属性の設定」を参照してください。

## 5.2.Webコンテナ設定項目一覧

### Webコンテナ管理コンソールで設定可能な項目一覧

操作項目	項目	説明
ログ	アクセスログ取得	クライアントからのアクセスのログを取得するか、しないかを設定します。
	Web Application 統計情報のログ	Web Application の統計情報ログを取得するか、しないかを設定します。
	JavaVM メモリ情報のログ取得	JavaVM の統計情報ログを取得するか、しないかを設定します。
JavaVM の統計情報	収集間隔(分)	統計情報を採取する間隔を分単位で設定します。
	起動時から収集を行う	Web コンテナ起動時から、JavaVM の統計情報を収集するかどうかを設定します。
	収集開始/終了	統計情報の収集を開始、または終了の指示ができます。
	表示の更新	最新の統計情報を表示します。
Web Application の統計情報	収集間隔	統計情報を採取する間隔を分単位で設定します。
	起動時から収集を行う	Web コンテナ起動時から、Web Application の統計情報を収集するかどうかを設定します。
	収集開始/終了	Web Application の統計情報の収集を開始、または終了の指示ができます。
	表示の更新	Web Application の最新の統計情報を指示します。
	グラフ表示	選択した Web Application の統計情報をグラフ表示します。
コネクタ	同時接続数	リクエスト受け付け用ソケットのバックログ値(待ち行列)です。
	接続タイムアウト	クライアントとの接続がタイムアウトするまでの時間を設定します。
	バッファサイズ	リクエストを処理する入力ストリームのバッファサイズを設定します。
	DNS のルックアップ	リモートホスト名の名前解決に DNS を使用するかどうかを設定します。

	IP アドレス	複数アドレスを持つサーバで、特定のアドレスに対するリクエストのみ許可したい場合に指定します。
	ポート番号	連携する Web サーバとの通信用ポート番号、もしくは HTTP リクエストを受け付けるポート番号を表示します。
	最小プロセッサ数	Web コンテナ起動時に生成するリクエスト受け付けプロセッサの数を設定します。
	最大プロセッサ数	Web コンテナが生成できるリクエスト受け付けプロセッサの最大数を設定します。
	プロセッサ警告のしきい値	Web コンテナは、アクティブなリクエスト受け付けプロセッサの数が最大数に近づいた時に運用管理コンソールに警告を通知します。その閾値を設定します。
コンテキスト	セッション ID の管理	セッション ID の管理に、cookie を利用するか、URLRewriting を利用するかを選択します。
	クロスコンテキスト	同じ仮想ホスト上で動作している他の Web Application のコンテキスト情報を参照可能にするかどうかの指定をします。
	再リロード可能	Web Application に含まれるサーブレットが更新された時に自動的にリロードを行うかどうかを設定します。
	プロセッサ処理タイムアウト (秒)	リクエスト受け付けプロセッサが、開始したリクエスト処理 (Web Application) がストールしたと判断するまでの時間を設定します。
ローダ	再ロード可能	Web Application に含まれるクラスが更新された時に自動的にリロードを行うかどうかを表示します。
セッションマネージャ	セッション ID 初期化文字列	セッション ID の元ネタとなる乱数の生成に使用するタネとなる文字列を表示します。
	最大有効セッション数	Web Application で作成できるセッションオブジェクトの最大数を表示します。-1 は制限が無いことを示しています。
Web Application の管理	起動	停止している Web Application を起動します。
	停止	実行中の Web Application を停止させ、利用できなくします。
	配備解除	Web Application を停止し、サーバから削除します。
Web Application の管理 (配備) サーバ上に配置されたディレクトリまたは WAR ファイルの配備	WAR ファイルまたはディレクトリの URL	配備する WAR ファイル、またはディレクトリの URL を設定します。
	Application Name	Web Application の名前を設定します。この名前は、Web コンテナ内で Web Application を一意に識別するために使用します。
	Context Root	Web ブラウザからアクセスする際に指定する Web Application の URI を設定します。
	Virtual Server	Web コンテナ内に存在する Virtual Server のうち、関連付けたい Virtual Server 名を設定します。
	配備	配備指示を実行します。

Web Application の管理 (配備) WAR ファイルの配備アップロード	アップロードする WAR ファイルの選択	アップロードする WAR ファイルを設定します。
	Application Name	Web Application の名前を設定します。この名前は、Web コンテナ内で Web Application を一意に識別するために使用します。
	Context Root	Web ブラウザからアクセスする際に指定する Web Application の URI を設定します。
	Virtual Server	Web コンテナ内に存在する Virtual Server のうち、関連付けたい Virtual Server 名を設定します。
	配備	配備指示を実行します。

## MOで設定可能な項目一覧

Dottedname : server.web-container

属性名 (attribute-name)	説明	既定値
log-available-accesslog	クライアントからのアクセスのログを取得するか、しないかを設定します。	true
log-available-webap-statisticslog	Web Application の統計情報ログを取得するか、しないかを設定します。	false
log-available-javavm-statisticslog	JavaVM の統計情報ログを取得するか、しないかを設定します。	false
stat-vm-collection-interval	統計情報を採取する間隔を分単位で設定します。	5
stat-vm-collection-startup	Web コンテナ起動時から、JavaVM の統計情報を収集するかどうかを設定します。	false
stat-webap-collection-interval	Web Application の統計情報を採取する間隔を分単位で設定します。	5
stat-webap-collection-startup	Web コンテナ起動時から、Web Application の統計情報を収集するかどうかを設定します。	false
session-replication-jndi-url	HTTP セッションのみのレプリケーションを、レプリケーション先の JNDI サーバを列挙して設定します。  例: rmiiop://<ホスト 1>:<JNDI サーバポート 1>,<ホスト 2>:<JNDI サーバポート 2>...	<null>

Dottedname : server.web-container.property

属性名 (attribute-name)	説明	既定値
default-buffer-size	Servlet や JSP のレスポンスバッファサイズのデフォルト値を設定します。次のように、property の name、および value 属性として指定します。  otxadmin> set server.web-container.property.default-buffer-size=10kb  単位はキロバイトで "10kb" のように単位を指定する必要があります (単位は小文字です)。なお、次の指定があった場合は、それぞれの指定を優先します。  ・ Servlet で setBufferSize() を呼び出し。 ・ JSP で page ディレクティブの buffer 属性を指定。	—
default-encoding	Servlet や JSP のレスポンス文字エンコーディングのデフォルト値を設定しま	—

	<p>す。次のように、property の name、および value 属性として指定します。</p> <pre>otxadmin&gt; set server.web-container.property.default-encoding=Shift_JIS</pre> <p>なお、次の指定があった場合は、それぞれの指定を優先します。</p> <ul style="list-style-type: none"> <li>• Servlet で setContentTypes () を呼び出し。</li> <li>• JSP で page ディレクティブの contentType 属性を指定。</li> </ul>	
j2ee-enabled	<p>認証情報の EJB コンテナへの伝播、および J2EE のトランザクションサービスとの連携を ON/OFF します。“false” を指定すると、これらの機能が利用できなくなります。次のように、property の name、および value 属性として指定します。</p> <pre>otxadmin&gt; set server.web-container.property.j2ee-enabled=true</pre>	<p>Web Edition</p> <p>… false</p> <p>その他の Edition</p> <p>… true</p>

Dottedname : server.http-service.virtual-server.virtual-server-name

属性名 (attribute-name)	説明	既定値
default-web-module	コンテキスト名を省略した時に動作する、デフォルトの Web モジュールです。	—
hosts	仮想サーバを選択するためのホストリクエストヘッダの中で有効な値のカンマ(,)区切りのリストです。	—
http-listeners	仮想サーバに関連付けされている HTTP リスナのカンマ(,)区切りのリストです。	—
Id	仮想サーバの識別名です。	—

Dottedname : server.http-service.virtual-server.virtual-server-name.property

属性名 (attribute-name)	説明	既定値
valve	<p>Valve を実装したクラスを指定します。次のように、property の name、および value 属性として指定します。</p> <pre>otxadmin&gt; set server.http-service.virtual-server.server.property.valve =org.apache.catalina.valves.RequestDispatchTestValve</pre>	—

Dottedname : server.http-service.http-listener.http-listener-name

属性名 (attribute-name)	説明	既定値
accept-count	リクエスト受け付け用ソケットのバックログ値(待ち行列)です。	10
address	複数アドレスを持つサーバで、特定のアドレスに対するリクエストのみ許可したい場合に指定します。	0.0.0.0
buffer-size	リクエストを処理する入力ストリームのバッファサイズを設定します。	2048
connection-timeout	クライアントとの接続がタイムアウトするまでの時間を設定します。	60000
default-virtual-server	リスナに関連付けされた仮想サーバ名です。	—
enable-lookups	リモートホスト名の名前解決に DNS を使用するかどうかを設定します。	false
enabled	リスナの活性状態を表します。	true
Id	リスナの識別用名です。	—
limit-processors	Web コンテナは、アクティブなリクエスト受け付けプロセッサの数が最大数に近づ	15

	いた時に運用管理コンソールに警告を通知します。その閾値を設定します。	
max-processors	Web コンテナが生成できるリクエスト受け付けプロセッサの最大数を設定します。	20
min-processors	Web コンテナ起動時に生成するリクエスト受け付けプロセッサの数を設定します。	5
port	連携する Web サーバとの通信用ポート番号、もしくは HTTP リクエストを受け付けるポート番号を表示します。	—
protocol	リスナが扱うプロトコルです。HTTP/1.1(内蔵 Web サーバ) と AJP/1.3(外部 Web サーバ連携) があります。	HTTP/1.1
redirect-port	SSL リクエストが通常ポートに来た時のリダイレクト先ポート番号です。	—
security-enabled	SSL を利用したセキュアな通信を行うかどうかを表します。	false
server-name	クライアントに送信する URL のホスト名に設定されるサーバ名です。	—
xpowered-by	レスポンスに X-Powered-By ヘッダを付加するかどうかを表します。	false

Dottedname : server.http-service.http-listener.*http-listener-name*.property

属性名 (attribute-name)	説明	既定値
max-post-size	POST で扱える最大データサイズを指定します。指定する単位はバイトです。次のように、property の name、および value 属性として指定します。  otxadmin> set server.http-service.http-listener.http-listener-1  .property.max-post-size=4194304	2048000
uri-encoding	URI のエンコードに使う文字コードを設定します。次のように、property の name、および value 属性として指定します。  otxadmin> set server.http-service.http-listener.http-listener-1  .property.uri-encoding=Windows-31J	—
use-body-encoding-for-uri	GET メソッドで SetCharacterEncoding() で指定したエンコーディングで URI のパラメータをエンコードするかどうかを指定します。SetCharacterEncoding() のエンコーディング指定を有効にするには、"true"を指定します。次のように、property の name、および value 属性として指定します。  otxadmin> set server.http-service.http-listener.http-listener-1  .property.use-body-encoding-for-uri=true	false

Dottedname : server.applications.web-module.*web-module-name*

属性名 (attribute-name)	説明	既定値
context-root	コンテキストルートの名前です。	—
cookies	セッション ID の管理に、cookie を利用するか、URLRewriting を利用するかを選択します。	true
cross-context	同じ仮想ホスト上で動作している他の Web Application のコンテキスト情報を参照可能にするかどうかの指定をします。	false

description	Web Application の説明です。	—
enabled	Web Application の活性状態を表します。	true
location	配備先のパス名です。	—
name	Web Application の名前です。	—
object-type	Web Application のタイプです。	—
privileged	内部クラスへのアクセスを許可するかどうかを表します。	false
reloadable	Web Application に含まれるサーブレットが更新された時に自動的にリロードを行うかどうかを設定します。	false
webap-alive-timeout	リクエスト受け付けプロセッサが、開始したリクエスト処理 (Web Application) がストールしたと判断するまでの時間を設定します。'0' では監視しません。	0

※コンテキストパス (context-root) に “/” を含めると一部機能が利用できなくなります。詳しくは [注意制限事項 > 3.Web コンテナ > 3.1.Web コンテナの注意事項] を参照してください。

### JavaVMオプションで設定可能な項目一覧

WebOTXは、Servlet/JSPコンテナにさまざまな機能があり、それらの機能の制御を行えるJava VM オプションを説明します。

下記に動作モード(シングルプロセスモード/マルチプロセスモード)別の設定方法を記載します。

#### ■シングルプロセスモードの場合

・運用管理コンソールでの設定方法(推奨)

運用管理コンソールにログインし、下記の箇所に設定を記述した後、ドメインを再起動してください。

<ドメイン名>

-「アプリケーションサーバ」

-「JVM構成」

の順にクリックしていき、「属性」タブの「JVMオプション」の「システムJVMオプション」の欄に“-D”の後に続いて設定を記述してください。その後、ドメインを再起動することで反映されます。

設定例:

```
-Dwebotx.webcont.jndidelete=480
```

・domain.xml 直接編集での設定方法

ドメインを停止し、下記の例のように対象ファイルの要素にタグを記述した後“-D”の後に続いて設定を記述してください。その後、ドメインを起動することで設定が反映されます。

設定例:<WebOTX インストールディレクトリ>/domains/domain1/config/domain.xml

```
<java-config>
```

```
<system-jvm-options>-Dwebotx.webcont.jndidelete=480</system-jvm-options>  
</java-config>
```

## ■マルチプロセスモードの場合

### ・運用管理コンソールでの設定方法

運用管理コンソールにログインし、下記の箇所に設定を記述した後、TP システムを再起動してください。

<ドメイン名>

-「TP システム」

-「アプリケーショングループ」

-<アプリケーショングループ名>

-「プロセスグループ」

-<プロセスグループ名>

の順にクリックしていき、「属性」タブの「Java システムプロパティ」の欄に設定を記述してください。

「名前」に項目名を記述し、「値」に指定する値を記述してください

設定例:

名前	値
webotx.webcont.jndidelete	480

### 設定項目一覧

項目	説明
webotx.webcont.jndidelete	JNDI に格納したセッション情報をタイムアウト後、何秒で削除するかを指定します。 デフォルト値は 480 秒です。
webotx.webcontainer.serverconfig.Append	JK プラグインが利用する連携のための設定ファイルを出力するか否かを指定します。 デフォルト値は true です。
webotx.webcontainer.serverconfig.ForwardAll	JK プラグインに対するコンテキストのリクエストを全て転送するか否かを指定します。 デフォルト値は true です。
webotx.webcontainer.serverconfig.ForwardKeyInfo	Web サーバの持つ認証情報を Web コンテナに転送するか否かを指定します。 デフォルト値は true です。
webotx.webcontainer.serverconfig.NoRoot	JK プラグインに対するリクエストのうち、ルートコンテキスト(/)を転送するか否かを指定します。 デフォルト値は true です。
webotx.webcontainer.serverconfig.Omlistener.OutputConfAuto	設定ファイルを出力するかどうかを指定します。ただし、ファイルが存在しない場合はこの値に関わらず出力します。



	デフォルト値は true です。
com.nec.webotx.enterprise.web.connector.checkLastProcessTimeInterval	IIOIP プラグインと Web コンテナとの通信状況のチェック間隔を指定します。 デフォルト値は 1800 です。
com.nec.webotx.enterprise.web.connector.fileSwitchSize	IIOIP プラグインが扱うメッセージのサイズが大きい場合にテンポラリファイルを利用しますが、そのしきい値を指定します。 デフォルト値は 128(Kbyte)です。
com.nec.webotx.enterprise.web.connector.iioipDriveMode	fileSwitchSize で指定したサイズを超えた場合に利用する一時媒体にメモリを利用するかファイルを利用するかどうかを指定します。 デフォルト値はファイルです。
com.nec.webotx.enterprise.web.connector.iioipSendMaxSize	IIOIP プラグインと Web コンテナ間でやりとりするメッセージを分割送信する場合のサイズを指定します。 デフォルト値は 128(Kbyte)です。
com.nec.webotx.enterprise.web.connector.poolBufferSize	IIOIP プラグインからのリクエストを処理する時に使用するバッファ領域の初期個数を指定します。 デフォルト値は 5 です。
com.nec.webotx.enterprise.web.connector.securityEnabled	マルチプロセスモード時、Cookie に登録したセッション情報にセキュアオプションを付加するか否かを指定します。 デフォルト値は false です。
com.nec.webotx.enterprise.web.connector.useBodyEncodingForURI	ContentType 又は Request.setCharacterEncoding メソッドで指定されたエンコーディングを URL のデコードに使用するか否かを指定します。 デフォルト値は false です。
com.nec.webotx.enterprise.web.connector.URIEncoding	「http://contents?a=%12%34」形式で記述された URL をデコードする際の文字エンコーディングを指定します。 デフォルト値は「ISO-8859-1」です。
com.nec.webotx.enterprise.overrideablejavaxpackages	Web アプリケーション内に配置したクラスのうちパッケージ名が javax、sun、org 等で始まるクラスは参照することができません。参照を許可するパッケージをここで指定します。複数設定する場合はカンマで区切って指定します。 ("javax.sub.org.sub"等) デフォルト値はありません。
com.nec.webotx.enterprise.web.InvalidOverridableJavaxPackages	Webapp クラスローダが自身でロードするクラスから除外するパッケージを指定します。 デフォルト値はありません。
com.nec.webotx.enterprise.registerRequest	リクエスト毎に発生する MO 登録処理を行うかどうかを指定します。 デフォルト値は false です。
jvmRoute	ロードバランサを使用して、サーバ固定で振り分けたい場合に jsessionid に付加させる任意の文字列を指定します。プラグインによる Sticky セッションの負荷分散を使用する場合には値にワーカ名 (ajp13 など) を指定してください。 デフォルト値はありません。

<p>com.nec.webotx.enterprise.web.enablePipelineRequestResponseChain</p>	<p>Valve で生成した Request オブジェクトや Response オブジェクトをサーブレットまで届けるかどうかを指定します。</p> <p>デフォルト値は false です。</p>
<p>com.nec.webotx.enterprise.checkURLPatternAfterMatching</p>	<p>認証対象外の URL に ":" (コロン) を含む場合、URL を異常とするかどうかを指定します。</p> <p>true :異常とします false:正常とします</p> <p>デフォルト値は false です。</p>
<p>user.language</p>	<p>Web コンテナが動作する言語ロケールを指定します。</p> <p>デフォルト値は動作環境に依存します。</p>
<p>org.apache.catalina.session.useConcurrentHashMap</p>	<p>セッションオブジェクトの属性の管理に ConcurrentHashMap を使用するかどうかを指定します。</p> <p>デフォルト値は false です。</p>
<p>org.apache.jasper.compiler.DefaultSessionEnabled</p>	<p>JSP の page ディレクティブの session 属性のデフォルトを指定します。</p> <p>デフォルト値は true です。</p>

## 5.3.HTTPのみのセッションレプリケーション

WebOTX V7.1 より、HTTP のみセッションレプリケーションを利用できるようになりました。従来、レプリケーションを行うと JDBC データソース、JMS リソースも同じ内容となってしまう、ドメインごとに異なる設定内容にできませんでした。これにより、複数の Web コンテナおよび EJB コンテナそれぞれに HTTP セッションの共有が容易にできるようになります。

HTTP のみセッションレプリケーションを利用するには、`session-replication-jndi-url` に、つぎのように設定します。

```
rmiiop://<ホスト 1>:<JNDI サーバポート 1>,<ホスト 2>:<JNDI サーバポート 2>...
```

ホスト名にはレプリケーションさせる JNDI サーバが動作しているホスト名または IP アドレスを記述します。

JNDI サーバポート番号は JNDI サーバが動作している WebOTX ドメインの組み込み IIOP リスナのポート番号となります。インストール時に作成される domain1 のポート番号は 7780 です。省略したときは自ドメインの組み込み IIOP リスナのポート番号が使用されます。

複数の WebOTX ドメイン間で HTTP セッションをレプリケートするときはレプリケートする全ての WebOTX ドメインの JNDI サーバのホスト名、ポート番号を列挙した指定を各 WebOTX ドメインに対して設定します。列挙する順番は意識する必要はありません。

例えば二つの WebOTX ドメイン domain1 と domain2 で HTTP セッションレプリケーションを行うためには

```
host1: domain1 のホスト名
port1: domain1 の組み込み IIOP リスナのポート番号
host2: domain2 のホスト名
port2: domain2 の組み込み IIOP リスナのポート番号
```

のように設定されている場合、

```
rmiiop://host1:port1, post2:port2
```

を domain1 と domain2 の `session-replication-jndi-url` 属性に設定します。

### 5.3.1.HTTPのみのセッションレプリケーションの設定方法

#### 設定手順

#### ①運用管理コマンドによる設定

次の運用管理コマンドで設定します。

```
otxadmin> set server.web-container.session-replication-jndi-url= rmiiop://<ホスト 1>:<JNDI サーバポート 1>,<ホスト 2>:<JNDI
サーバポート 2>...
```

<例>

```
otxadmin> set server.web-container.session-replication-jndi-url= rmiiop://host1:7780,host2:7780
```

#### ②設定の反映

ドメインを再起動して、設定を反映します。

## 5.4.Webアプリケーションのコンテキスト名を動的反映しない設定方法

WebOTX には、Web アプリケーションが配備されたことを自動的に WebOTX Web Server (Apache) などの外部 Web サーバに反映させる仕組みがあり、デフォルトではこの機能が有効になっています。この機能により、ドメインを再起動することなく新規に配備された Web アプリケーションの存在を外部 Web サーバに認識させています。動的反映の処理には、ある程度負荷がかかるため、Web アプリケーションの更新のないシステムでは不要な負荷がかかることになります。そこで、この動的反映をしない設定をすることによって負荷を低減することができます。

※この設定を行った場合、新規に配備された Web アプリケーションを外部 Web サーバに認識させるには、外部 Web サーバおよびドメインを再起動する必要があります。

※環境設定ツールで設定を行うと、以下の設定は初期化されデフォルトの「動的反映を行う」状態になります。環境設定を行った場合は、再度、下記の設定を行ってください。

### 5.4.1.動的反映しない設定方法

#### 設定手順

- ①外部Webサーバの停止
- ②workers.propertiesの設定

`\${INSTANCE\_ROOT}/config/WebCont/workers.properties` に次の名称でワーカ名を追加してください。

```
ajp13_original
```

[ workers.properties の例]

```
#----- DEFAULT worket list -----  
#-----  
  
ps=/  
  
#  
# The workers that your plugins should create and work with  
#  
worker.list=ajp13, ajp13_original  
  
#  
#----- DEFAULT ajp13 WORKER DEFINITION -----  
#-----  
#  
#  
# Defining a worker named ajp13 and of type ajp13  
# Note that the name and the type do not have to match.  
#  
# TM_PLUGIN_PORT-start  
worker.ajp13.port=8009  
# TM_PLUGIN_PORT-end  
worker.ajp13.host=localhost  
worker.ajp13.type=ajp13  
worker.ajp13.cachesize=150
```

- ③外部Webサーバの起動

外部 Web サーバを起動し、設定内容を反映します。

## 5.5.Webサーバプラグイン設定項目一覧

以下は JK プラグイン(シングルプロセスモード時に使用)、IIOP プラグイン(マルチプロセスモード時に使用) 共通の設定項目一覧です。

連携設定項目 : workers.properties (JK)、ior\_workers.properties (IIOP)

プロパティ名	説明、書式	既定値
worker.list	<p>ワーカ名のリストを指定します。一つ以上のワーカを指定する必要があり、複数指定する場合はコンマ(,)区切りで記述します。</p> <p>書式: worker.list=ajp13</p> <p>※負荷分散機能使用時のワーカタイプの"lb"は1つだけ指定できます。</p>	ajp13 (JK) otxiop (IIOP)
worker.maintain	<p>再利用のために接続を再構築する時間(秒単位)を指定します。負荷分散時には指定した間隔でリクエストカウンタが初期化されます。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式: worker.maintain=60</p>	60
type	<p>ワーカのタイプを指定します。負荷分散機能を利用する場合は lb を指定してください。</p> <p>デフォルト値は ajp13 (JK)、otxiop (IIOP) です、</p> <p>書式: worker.ajp13.type=ajp13</p>	ajp13 (JK) otxiop (IIOP)
host	<p>ワーカのホスト名、または IP アドレスを指定します。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式: worker.ajp13.host=localhost</p>	localhost
port	<p>ワーカの使用するポート番号を指定します。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式: worker.ajp13.port=8009</p>	8009

socket_timeout	<p>リモートホストとコネクタ間におけるソケット通信のタイムアウト時間(秒単位)を指定します。</p> <p>指定した時間以上が経過すると、エラーを返し、リトライ処理を行います。</p> <p>デフォルト値である0を指定すると、ソケット操作の待ち時間は無制限となります。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式:</p> <pre>worker.ajp13.socket_timeout=0</pre>	0
socket_keepalive	<p>Web サーバとWebOTX 間に、アクティブでないコネクションは切断するファイアウォールを設置している場合はこの指定を true にする必要があります。</p> <p>この指定を行うと、OS にアクティブでないコネクションに対しての KEEP_ALIVE メッセージを送ることができます。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式:</p> <pre>worker.ajp13.socket_keepalive=false</pre>	false
connection_pool_size または cachesize	<p>Web サーバが Web コンテナに対して張るコネクションのプールサイズの上限值を指定します。</p> <p>コネクタはこの値の数だけ同時にコネクションを張ることができ、この値を超えたリクエストはエラーとなります。</p> <p>※“cachesize”による指定も可能ですが、旧バージョン表記のため、“connection_pool_size”を指定してください。</p> <p>書式:</p> <pre>worker.ajp13.connection_pool_size=150</pre>	150
connection_pool_minsize	<p>Web サーバが Web コンテナに対して張るコネクションのプールサイズの下限值を指定します。</p> <p>書式:</p> <pre>worker.ajp13.connection_pool_minsize=75</pre>	("connection_pool_size" + 1) / 2
connection_pool_timeout	<p>Web サーバが Web コンテナに対して張るコネクションのプールを維持する時間(秒単位)を指定します。</p> <p>これは WebOTX 上で動作するスレッド数を抑えるために使用します。</p> <p>デフォルト値は 0 であり、タイムアウトは発生しません。</p>	0

	<p>※この設定は JK プラグインでのみ設定可能です。</p> <p>書式:</p> <pre>worker.ajp13.connection_pool_timeout=0</pre>	
balance_workers または balanced_workers	<p>負荷分散対象となるワーカ名のリストを指定します。複数指定する場合は、コンマ(,)区切りで記述します。</p> <p>※“balanced_workers”による指定も可能ですが、旧バージョンの表記のため、“balance_workers”を指定してください。</p> <p>書式:</p> <pre>worker.ajp13.balanced_workers=ajp13_1,ajp13_2</pre>	なし
sticky_session	<p>セッション固定でのサーバの振り分けを行う場合はこの値を true に指定します。</p> <p>書式:</p> <pre>worker.ajp13.sticky_session=true</pre>	true
sticky_session_force	<p>障害発生先のドメインにリクエストを送った場合等に起こるワーカビジー状態の時にセッション ID が付いたリクエストは拒否するかどうかを指定します。</p> <p>この値が true の場合は、セッション ID が付いたリクエストにはステータスコード 503 が返ります。</p> <p>false の場合は新規にセッションを払い出します。</p> <p>書式:</p> <pre>worker.ajp13.sticky_session_force=false</pre>	false
method	<p>負荷分散機能利用時にワーカの選択形式を指定します。</p> <p>sticky_session を利用した負荷分散でセッション数が少ない時や振り分け先のサーバが多い時にはこの指定は必要ありません。</p> <p>“R”を指定すると、リクエスト数の少ないワーカを使用します。(デフォルト)</p> <p>“S”を指定すると、払い出されたセッション数の少ないワーカを使用します。</p> <p>“T”を指定すると、ネットワークトラフィックが最も少ないワーカを使用します。</p> <p>“B”を指定すると、処理中のワーカが少ないワーカを使用します。</p> <p>書式:</p>	R

	worker.ajp13.method=R	
lock	<p>ロードバランサが共有メモリ内のデータをロックするかどうかを指定します。</p> <p>“O”を指定すると、ロックを行いません。</p> <p>“P”を指定すると、ロックを行いますが、そのため性能は劣化します。</p> <p>書式:</p> <p>worker.ajp13.lock=O</p>	0
retries	<p>正常な応答が返ってこなかった場合に行うリトライ処理を含めた総リクエスト回数を指定します。下限値は 1 であり、それより小さな値を指定しても、1 が設定されます。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式:</p> <p>worker.ajp13.retries=3</p>	3
recovery_options	<p>WebOTX がファイルオーバーした際にどのようなリトライ処理を行うかをビットマスクで指定します。</p> <p>設定可能な値は以下になります</p> <p>1: WebOTX からリクエストを受け取った後の場合、リトライしません。</p> <p>2: リクエストヘッダをクライアントに送った後の場合、リトライしません。</p> <p>4: ブラウザに表示するレスポンスを作成中の場合、WebOTX との接続を切断します。</p>	0
fail_on_status	<p>WebOTX のサブレットコンテナが一時的に意図しないエラーコードを返してしまうような場合にこの指定を行ってください。</p> <p>ここに指定したステータスコードが WebOTX サブレットコンテナから返ってきた場合、503 に変換され、クライアントに返されます。</p> <p>書式:</p> <p>worker.ajp13.fail_on_status=404,500</p>	0
max_packet_size	<p>AJP パケットの最大バイト数を指定します。最大値は 65535(Byte)です。</p> <p>この設定を変更する場合、WebOTX 側の“packetSize”の指定も変更する必要があります。</p>	8192



	<p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式: worker.ajp13.max_packet_size=8192</p>	
mount	<p>ワーカのマップ情報をスペース区切りで表示します。</p> <p>書式: worker.ajp13.mount=/WebAP1/*</p>	なし
secret	<p>秘密の文字列を登録することができます。</p> <p>この設定により WebOTX コンテナの“request.secret”で設定された文字列と同一の文字列が設定されたワーカからのリクエストのみを受け付けることができます。</p> <p>書式: worker.ajp13.secret=secret</p>	なし
max_reply_timeouts  ※V7.11.02 より使用可	<p>負荷分散環境においていくつかのワーカで“reply_timeout”で設定された値よりも長い時間を設定したい場合にこの指定を行います。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式: worker.ajp13.max_reply_timeouts=0</p>	0
recover_time	<p>負荷分散時にエラー状態となったワーカを復帰状態にするまでの待ち時間を指定します。</p> <p>この時間経過後にワーカは“recovering”の状態になり、リクエスト処理が再開されます。</p> <p>書式: worker.ajp13.recover_time=60</p>	60
activation	<p>負荷分散環境において、ワーカを“disabled”または“stopped”の状態に設定します。</p> <p>“disabled”のワーカはセッション付きのリクエストのみ受け付けます。</p> <p>“stopped”のワーカはすべてのリクエストを拒否します。</p> <p>設定には“d”または“D”、“s”または“S”を記述します。</p> <p>書式: worker.ajp13.activation=d</p>	なし

route	<p>負荷分散環境において、セッション固定でサーバを振り分けるための JVM ルートの設定を行います。</p> <p>書式： worker.ajp13.route=ajp13</p>	<ワーカ名と同一>
distance	<p>負荷分散環境において、ワーカ間の重み付けを設定します。この値の最も小さいワーカが“error”、“disabled”、“stopped”の状態の場合はより大きな値のワーカが選択されます。</p> <p>書式： worker.ajp13.distance=0</p>	0
domain	<p>負荷分散環境において、同じドメイン名を使用するワーカを独立したワーカとして使用する場合に指定します。</p> <p>sticky_session を利用している場合はセッションルートとして使用されます。</p> <p>書式： worker.ajp13.domain=domain1</p>	なし
redirect	<p>不正なセッション ID のためにリクエストがエラーとなると、この指定を行ったワーカを代わりに使用します。</p> <p>sticky_session を利用している場合はこの指定を推奨します。</p> <p>書式： worker.ajp13.redirect=ajp23</p>	なし

連携設定項目 : mod\_jk.conf-auto、mod\_jk-20.conf-auto (JK) mod\_jk\_om.conf-auto、mod\_jk-20\_om.conf-auto (IIOP)

※ドメインを再起動すると、mod\_jk.conf-auto、mod\_jk-20.conf-auto ファイルは再作成され、以前記述していた定義は削除されます。必要に応じてファイル名を変更してください。

プロパティ名	説明、書式	既・値
JkWorkersFile	<p>WebOTX コンテナと連携するために必要な設定ファイルのパスを指定します。これは Web サーバ起動時に一回のみ読み込まれます。</p> <p>“JkWorkerProperty”を設定していない場合は、これを設定する必要があります。</p> <p>書式： JkWorkersFile “C:/WebOTX/domains/domain1/config/Webcont/workers.properties”</p>	<WebOTX ドメインディレクトリ/config/ /WebCont/ workers.properties
JkWorkerProperty	<p>設定ファイルのパス指定を Apache の設定ファイル (httpd.conf) 内</p>	なし

	<p>で行います。</p> <p>書式:</p> <p>JkWorkerProperty</p> <p>“C:/WebOTX/domains/domain1/config/Webcont/workers.properties”</p>	
JkShmFile	<p>UNIX 系 OS でのみ使用する共有メモリファイルを指定します。</p> <p>この設定を行っていない場合、デフォルト箇所にファイルが作成され、Web サーバのログに下記が出力されます。</p> <p>[warn] No JkShmFile defined in httpd.conf. Using default /opt/WebOTX/domains/domain1/logs/jk-runtime-status</p> <p>書式:</p> <p>JkShmFile “/opt/WebOTX/domains/domain1/logs/jk_shm”</p>	<p>&lt;WebOTXドメインディレクトリ/logs</p> <p>/jk-runtime-status</p>
JkShmSize	<p>共有メモリファイルのサイズ(KB 単位)を指定します。</p> <p>書式:</p> <p>JkShmSize 64</p>	OS に依存 (通常 64KB 未満)
JkMountFile	<p>ワーカーのマッピング情報を記述するファイルパスを指定します。</p> <p>書式:</p> <p>JkMountFile</p> <p>“C:/WebOTX/domains/domain1/config/WebCont/mod_jk-20.conf-auto”</p>	なし
JkMountFileReload	<p>“JkMountFile”で指定したファイルを再読み込みする間隔(秒単位)を指定します。</p> <p>書式:</p> <p>JkMountFileReload 60</p>	60
JkMount	<p>ワーカーにマウントを行うコンテキストパスを指定します。</p> <p>書式:</p> <p>JkMount “/WebAP1/*”</p>	なし
JkUnMount	<p>ワーカーにマウントを行わないコンテキストパスを指定します。この指定があるコンテキストに対してのリクエストは WebOTX までフォワードされません。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式:</p> <p>JkUnMount “/WebAP1/*”</p>	なし

JkAutoAlias	<p>Web アプリケーションのコンテキストディレクトリを Apache のドキュメントベース内に置く場合にそのパスを指定します。</p> <p>書式: JkAutoAlias "C:¥¥WebOTX¥¥domains¥¥domain1¥¥docroot"</p>	なし
JkMountCopy ※V7.11.02 より使用可	<p>"On","Off","All"の設定が可能であり、"On"を指定されたバーチャルサーバにはメインサーバのマウントがコピーされます。(JkMount、JkUnMount の設定を含む)"Off"を指定するとコピーされません。"All"を指定するとすべてのバーチャルサーバに設定が反映されます。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>書式: JkMountCopy All</p>	ALL
JkWorkerIndicator	<p>ワーカー名を識別するための Apache 環境変数名を指定します。</p> <p>書式: JkWorkerIndicator</p>	JK_WORKER_NAME
JkLogFile	<p>プラグインのログファイルパスを指定します。</p> <p>書式: JkLogFile "C:/WebOTX/domains/domain1/logs/webcontainer/mod_jk-20.log"</p>	<WebOTXドメインディレクトリ/logs/mod_jk-20.log
JkLogLevel	<p>プラグインのログレベルを指定します。</p> <p>設定可能な値は info/debug/error/emerg です。</p> <p>書式: JkLogLevel error</p>	error
JkLogStampFormat ※V7.11.02 より使用可	<p>プラグインのログファイルの出力形式を指定します。</p> <p>設定可能な値は以下になります。</p> <p>※この設定は JK プラグインのみ設定可能です。</p> <p>%a 短縮された曜日名 %A 完全な曜日名 %b 短縮された月の名前 %B 完全な月の名前 %y 短縮された年 %Y 完全な年</p>	"%a %b %d %H: %M:%S.000 %Y"

	<p>%m 月</p> <p>%d 日付</p> <p>%H 時間</p> <p>%M 分</p> <p>%S 秒</p> <p>%Q ミリ秒</p> <p>%q マイクロ秒</p> <p>%l 12 時間表示での時間(01 から 12 まで)</p> <p>%j 年間での日付を 10 進数で表示 (001 から 366)</p> <p>%p 時刻表示における AM または PM の表示</p> <p>%U 年間で何番目の週であるかを表示(年間の最初の日曜を最初の週的第一天として数える)</p> <p>%W 年間で何番目の週であるかを表示(年間の最初の月曜を最初の週的第一天として数える)</p> <p>%w 10 進数で表示された曜日(日曜は 0)</p> <p>%x 月/日付/年</p> <p>%X 時間:分:秒</p> <p>%c 月/日付/年 時間:分:秒</p> <p>%Z タイムゾーンの表示</p> <p>%% 文字列“%”</p> <p>書式:  JkLogStampFormat “%a %b %d %H:%M:%S.000 %Y”</p>	
JkRequestLogFormat	<p>リクエストログの出力形式を指定します。  設定可能な値は以下になります。</p> <p>%b HTTP ヘッダを除いたリクエストバイト数(CLF 形式)</p> <p>%B HTTP ヘッダを除いたリクエストバイト数</p> <p>%H リクエストプロトコル</p> <p>%m リクエストメソッド</p> <p>%p リクエストを送る Web サーバのポート番号</p> <p>%q クエリ文字列</p> <p>%r リクエストの最初の行</p> <p>%s リクエスト時の HTTP ステータスコード</p> <p>%T リクエスト処理時間</p> <p>%U クエリ文字列を含まない URL</p> <p>%v Web サーバ名</p> <p>%V “UseCanonicalName”で設定された Web サーバ名</p>	なし(ログ出力されません)

	<p>%w WebOTX が認識するワーカ名 %R Web サーバが認識するワーカ名</p> <p>書式: JkRequestLogFormat "%b"</p>	
JkExtractSSL	<p>SSL 通信を許可するかどうかを指定します。</p> <p>書式: JkExtractSSL On</p>	On
JkHTTPSIndicator	<p>SSL を示す Apache 環境変数名を指定します。</p> <p>書式: JkHTTPSIndicator HTTPS</p>	HTTPS
JkCERTSIndicator	<p>SSL クライアント証明書を示す Apache 環境変数名を指定します。</p> <p>書式: JkCERTSIndicator SSL_CLIENT_CERT</p>	SSL_CLIENT_CERT
JkCIPHER Indicator	<p>SSL 暗号を示す Apache 環境変数名を指定します。</p> <p>書式: JkCIPHERIndicator SSL_CIPHER</p>	SSL_CIPHER
JkCERTCHAINPrefix	<p>SSL チェーン認証を示す Apache 環境変数名を指定します。</p> <p>書式: JkCERTCHAINPrefix SSL_CLIENT_CERT_CHAIN_</p>	SSL_CLIENT_CERT_CHAIN_
JkSESSIONIndicator	<p>SSL セッションを示す Apache 環境変数名を指定します。</p> <p>書式: JkSESSIONIndicator SSL_SESSION_ID</p>	SSL_SESSION_ID
JkKEYSIZEIndicator	<p>SSL 暗号鍵のサイズを示す Apache 環境変数名を指定します。</p> <p>書式: JkKEYSIZEIndicator SSL_CIPHER_USEKEYSIZE</p>	SSL_CIPHER_USEKEYSIZE
JkOptions	<p>プラグインのオプション設定を指定します。</p> <p>項目の前に"+"か"-を付けることで有効/無効を指定します。</p> <p>デフォルトでは ForwardURIProxy が有効であり、+ForwardKeySize の設定が記述されます。</p> <p>設定可能な値は以下になります。</p>	+ForwardURICompat

	<p>ForwardURISCompat: 受け取った URI をそのまま WebOTX へ転送します。</p> <p>ForwardURISCompatUnparsed: 受け取った URI を解釈せず、WebOTX へ転送します。</p> <p>ForwardURISEscaped: 受け取った URI についてセッション ID が付いているリクエストに対しては ForwardURISCompatUnparsed と同様の動作をし、その他は ForwardURISCompat と同様の動作をします。</p> <p>ForwardURISProxy: 受け取った URI にエンコード処理を行います。</p> <p>ForwardKeySize: Servlet API 2.3 で要求されている SSL キーサイズを転送します。</p> <p>書式: JkOptions +ForwardKeySize</p>	
JkEnvVar	<p>リクエスト項目として WebOTX コンテナに送られる独自の環境変数を指定します。</p> <p>WebOTX 側で request.getAttribute(Attributem 名)を呼び出すことでこの値が参照できますが、request.getAttributeNames()ではリスト表示されません。</p> <p>書式: JkEnvVar VARIABLE</p>	なし

以下は IIS 連携時の設定項目一覧です。

設定を反映させるには下記のレジストリファイルを編集、実行し、IIS を再起動してください。

プロパティ名	説明、書式	既定値
extension_uri	<p>WebOTXコンテナと連携するために必要な設定ファイルのパスを指定します。これは Web サーバ起動時に一回のみ読み込まれます。</p> <p>“JkWorkerProperty”を設定していない場合は、これを設定する必要があります。</p> <p>書式: JkWorkersFile “C:/WebOTX/domains/domain1/config/Webcont/workers.properties”</p>	“/webcont/isapi_redirect.dll”
log_file	<p>ログファイルを作成する場所を指定します。</p> <p>書式: “log_file”=“C:¥¥WebOTX¥¥domains¥¥domain1</p>	<p>&lt;WebOTXドメインディレクトリ &gt;¥¥logs¥¥webcontainer ¥¥isapi.log</p>

	¥¥logs¥¥webcontainer¥¥isapi.log”	
log_level	<p>ログレベルを指定します。 設定可能な値は debug/info/warn/error/trace です。</p> <p>書式: ”log_level”=”error”</p>	error
worker_file	<p>WebOTX コンテナと連携するために必要な設定ファイルのパスを指定します。これは Web サーバ起動時に一回のみ読み込まれます。</p> <p>書式: ”worker_file”=”C:¥¥WebOTX¥¥domains¥¥domain1¥¥config¥¥WebCont¥¥workers.properties”</p>	<p>&lt;WebOTXドメインディレクトリ &gt;¥¥config¥¥WebCont ¥¥workers.properties</p>
worker_mount_file	<p>uriworkermap.properties-auto ファイルへのパスを指定します。</p> <p>書式: ”worker_mount_file”=”C:¥¥WebOTX¥¥domains¥¥domain1¥¥config¥¥WebCont¥¥uriworkermap.properties-auto”</p>	<p>&lt;WebOTXドメインディレクトリ &gt;¥¥config¥¥WebCont ¥uriworkermap.properties-autp</p>
worker_mount_reload	<p>worker_mount_file で指定したファイルを再読み込みする間隔(秒単位)を指定します。</p> <p>書式: ”worker_mount_reload”=”60”</p>	なし
strip_session	<p>この設定を On にすると URL の中からセッション ID (jsessionid= で始まる文字列)を除き、リクエストを行います。</p> <p>書式: ”strip_session”=”false”</p>	false
auth_complete	<p>IIS5.1 との互換性を保つために必要となる場合があります。 デフォルト値は 1 であり、認証時に SF_NOTIFY_AUTH_COMPLETE を使用します。 0 を指定すると、SF_NOTIFY_PREPROC_HEADERS を使用し、HTTP の PUT メソッドを利用する時にこれが必要になる場合があります。</p> <p>書式: ”auth_complete”=”1”</p>	1
uri_select	URI の文字コード形式を指定し、	parsed



	<p>proxy/parsed/unparsed/escaped が指定可能です。</p> <p>セキュリティの観点からデフォルト値である“proxy”を推奨します。</p> <p>“parsed”を指定すると、“..”のような文字列も含みエンコード処理を行います。</p> <p>“unparsed”URL はエンコード処理を行いません。</p> <p>“escaped”を設定すると、セッション ID が付いているリクエストに対しては“unparsed”と同様の動作をし、その他は“parsed”と同様の動作をします。</p> <p>“proxy”を指定すると、“parsed”でエンコードされた URI の中で“..”のような不正なパスに再エンコード処理を行います。</p> <p>書式:</p> <p>“uri_select”=“proxy”</p>	<p>※V7.11.02 より</p> <p>proxy</p>
reject_unsafe	<p>この値を true に指定すると、“%”と“¥”を含む URL にはエラーを返します。</p> <p>ほとんどの Web アプリケーションではこのような URL は使用しないため、URL 操作を行う攻撃を防ぐことができます。</p> <p>書式:</p> <p>“reject_unsafe”=“false”</p>	false

## 5.6.IIOPプラグイン定義ファイルの設定方法

WebOTX Application Server Standard Edition/Enterprise Edition では、Web アプリケーションを動作させる Web コンテナを TP モニタ上でマルチプロセス動作する機能を提供します。このマルチコンテナ機能を利用するには、外部 Web サーバ(WebOTX Web Server(Apache))と Web コンテナを IIOP プラグインで連携する仕組みになっています。

IIOP プラグインの連携設定は、環境設定ツールを使用して連携設定を行います。また、さらに詳細な連携設定は、設定ファイルを直接編集して行います。ここでは、IIOP プラグインの連携の詳細設定、エラードキュメントを指定する方法を説明します。環境設定ツールを使用した連携設定については「セットアップガイド・インストール」を参照してください。

### 5.6.1.IIOPプラグイン定義ファイルの設定手順

外部Webサーバ(WebOTX Web Server(Apache))とWebコンテナの連携設定を行ったあとに、以下の設定をしてください。

#### 設定手順

- ①外部Webサーバの停止
- ②ior\_workers.propertiesの設定

設定ファイル `${INSTANCE_ROOT}/config/WebCont/ior_workers.properties` を編集し、保存します。

[ ior\_workers.properties の設定例 ]

```
# TM_PLUGIN_IOR-start
worker.list=otxiop
otx_domain=D:/WebOTX650/domains/domain1
worker.otxiop.type=otxiop
worker.otxiop.default_ior=localdomain
worker.otxiop.ior_def.localdomain=${otx_domain}/config/httpgateway.ior
worker.otxiop.cachesize=150
worker.otxiop.http_error.list=404,500
worker.otxiop.http_error.404.doc=file:///D:/temp/error/404.html
worker.otxiop.http_error.500.doc=file:///D:/temp/error/500.html
worker.otxiop.http_error.charset=Shift_JIS
# TM_PLUGIN_IOR-end
```

- ③外部Webサーバの起動

外部 Web サーバを起動し、設定内容を反映します。

### 5.6.2.IIOPプラグインの設定項目一覧

以下は、設定可能な IIOP プラグイン設定項目一覧です。

IIOP プラグインが自ドメイン(環境設定ツールで設定したドメイン)以外に配備されたコンテキストを呼び出すためコンテキストと IOR を対応付けする情報定義を行うことができます。

また、HTTP エラーおよび CORBA エラーに対応するエラードキュメントを指定することができます。IIOP プラグインの設定とともに、Apache のエラードキュメントの設定を行ってください。エラーの内容により有効になる設定箇所が変わります。

連携設定項目: ior\_workers.properties

プロパティ名	説明、書式	既定値
worker.otxiiop.type	<p>otxiiop のワーカータイプの指定。設定値は"otxiiop" 固定です。</p> <p>書式: worker.otxiiop.type=otxiiop</p>	otxiiop
worker.otxiiop.external_contexts	<p>他のドメインに存在するコンテキスト名を", " (カンマ) で区切り列挙します。ここで定義されていないコンテキストのリクエストが来た場合は "worker.otxiiop.default_ior" の IOR が使用されます。</p> <p>他のドメインに配備されたコンテキストにアクセスする場合に、ここに手動で設定、定義します。</p> <p>書式:</p> <pre>worker.otxiiop.external_contexts=&lt;コンテキスト名&gt;[, &lt;コンテキスト名&gt;]...</pre> <p>パラメータ説明:</p> <p>&lt;コンテキスト名&gt;: Web コンテキスト名 (先頭に"/"を付けない)</p>	なし
worker.otxiiop.context_ior	<p>コンテキストに対応する IOR を指定します。</p> <p>他のドメインに配備されたコンテキストにアクセスする場合に、手動で定義します。</p> <p>書式:</p> <pre>worker.otxiiop.context_ior.&lt;コンテキスト名&gt;=&lt;IOR 定義名&gt;</pre> <p>パラメータ説明:</p> <p>&lt;コンテキスト名&gt;: Web コンテキスト名 (先頭に"/"を付けない)</p> <p>&lt;IOR 定義名&gt; : OTXIorDef で定義されている IOR の名前。</p> <p>OTXIorDef に定義されていない場合はエラーログを出力します。</p>	なし
worker.otxiiop.default_ior	<p>worker.otxiiop.external_contexts にコンテキスト情報が存在しない場合に使用するデフォルト IOR を指定します。</p> <p>※必須項目、複数定義不可</p> <p>書式:</p> <pre>worker.otxiiop.default_ior=&lt;IOR 定義名&gt;</pre> <p>パラメータ説明:</p> <p>&lt;IOR 定義名&gt;: IOR を特定する名前</p> <p>IOR 定義名は"worker.otxiiop.ior_def.&lt;IOR 定義名&gt;"によって定義されている必要があります。</p>	なし
worker.otxiiop.iiop_retry_count	<p>IIOP メソッド呼び出すリトライ回数を指定します。</p> <p>書式:</p> <pre>worker.otxiiop.iiop_retry_count=&lt;リトライ回数&gt;</pre> <p>パラメータ説明</p> <p>&lt;リトライ回数&gt;: IIOP メソッド呼び出しのリトライ回数</p> <p>設定可能範囲は1以上です。</p>	3

worker.otxiiop.iiop_retry_interval	<p>IIOP メソッド呼び出しリトライ時のインターバル時間を指定します。</p> <p>書式:</p> <pre>worker.otxiiop.iiop_retry_interval=&lt;インターバル時間&gt;</pre> <p>パラメータ説明:</p> <p>&lt;インターバル時間&gt;: •IOP メソッド呼び出しリトライ時の待ち時間 (秒)</p>	5
worker.otxiiop.iiop_timeout	<p>IIOP メソッド呼び出しのタイムアウト時間を指定します。</p> <p>書式:</p> <pre>worker.otxiiop.iiop_timeout=&lt;タイムアウト時間&gt;</pre> <p>パラメータ説明</p> <p>&lt;タイムアウト時間&gt;: IIOP メソッド呼び出しがタイムアウトする時間 (秒)</p>	300
worker.otxiiop.iiop_max_send_size	<p>POST データ送信時のデータ分割単位を指定します。</p> <p>書式:</p> <pre>worker.otxiiop.iiop_max_send_size=&lt;パケット長&gt; (K M)</pre> <p>パラメータ説明</p> <p>&lt;パケット長&gt;: POST データの送信時のパケット分割単位</p> <p>最後に "K" がある場合は * 1024 の値になり</p> <p>最後に "M" がある場合は * 1024 * 1024 の値になる</p> <p>設定可能範囲は、32K から 9M です。</p>	128K

#### エラードキュメント指定設定項目

プロパティ名	説明、書式	既定値
worker.otxiiop.http_error.list	<p>エラードキュメントをデフォルトから変更する HTTP レスポンスの Status-Code の一覧を", "で区切り列挙します。 エラードキュメントを変更しない場合はこのプロパティを定義しない。</p> <p>書式:</p> <pre>worker.otxiiop.http_error.list=&lt;Status-Code&gt;[, &lt;Status-Code&gt;]...</pre> <p>パラメータ説明:</p> <p>&lt;Status-Code&gt;: Web コンテナから返ってくる HTTP レスポンスの Status-Code</p>	なし
worker.otxiiop.http_error.<Status-Code>.doc	<p>worker.otxiiop.http_error.list で指定された Status-Code に対応するエラードキュメント情報を定義します。</p> <p>書式:</p> <pre>worker.otxiiop.http_error.&lt;Status-Code&gt;.doc=&lt;File URL&gt;</pre> <p>パラメータ説明</p> <p>&lt;Status-Code&gt;: 前述を参照</p>	なし

	<p>&lt;File URL&gt;: "file://" で始まる URL</p> <p>例:</p> <pre>worker.otxiiop.http_error.404.doc= file:///C:/WebOTX/ WebServer2/error/404.html</pre>	
worker.otxiiop.http_error.charset	<p>エラードキュメントファイルのデフォルトのキャラクタセットを指定します。</p> <p>書式:</p> <pre>worker.otxiiop.http_error.charset=&lt;charset&gt;</pre> <p>パラメータ説明</p> <p>&lt;charset&gt;: エラードキュメントのデフォルトのファイルキャラクタセット</p>	ISO-8859-1
worker.otxiiop.http_error.<Status-Code>.charset	<p>指定したステータスコード用のエラードキュメントファイルのキャラクタセットを指定します。</p> <p>書式:</p> <pre>worker.otxiiop.http_error.&lt;Status-Code&gt;.charset=&lt;charset&gt;</pre> <p>パラメータ説明</p> <p>&lt;charset&gt;: ステータスコードのエラードキュメントのファイルキャラクタセット</p>	worker.otxiiop.http_error.charset の値
worker.otxiiop.corba_error.list	<p>エラードキュメントをデフォルトから変更する CORBA の Error-Name の一覧を", "で区切り列挙します。エラードキュメントを変更しない場合はこのプロパティを定義しません。</p> <p>書式:</p> <pre>worker.otxiiop.corba_error.list=&lt;Error-Name&gt;[, &lt;Error-Name&gt;]...</pre> <p>パラメータ説明:</p> <p>&lt;Error-Name&gt;: Web コンテナから返ってくる CORBA のエラーの名称</p> <p>例:</p> <pre>worker.otxiiop.corba_error.list=timeout</pre>	なし
worker.otxiiop.corba_error.<Error-Name>.err	<p>worker.otxiiop.corba_error.list で指定された Error-Name に対応する CORBA のエラー名と対応しているマイナーコードを定義します。</p> <p>書式:</p> <pre>worker.otxiiop.corba_error.&lt;Error-Name&gt;.err=&lt;エラー定義&gt;[, &lt;エラー定義&gt;]...</pre> <p>パラメータ説明:</p> <p>&lt;エラー定義&gt; : &lt;CORBA-ErrorName&gt; [( &lt;MinorCode-列挙&gt;   &lt;MinorCode-範囲指定&gt; ) ] ...</p> <p>&lt;Error-Name&gt; : 前述を参照</p> <p>&lt;CORBA-ErrorName&gt; : Web コンテナから返ってくる CORBA のエラー名</p>	なし

	<p>&lt;CORBA-MinorNum&gt; : Web コンテナから返ってくる CORBA のマイナーコード</p> <p>&lt;MinorCode-列挙&gt; :</p> <p>&lt;CORBA-MinorCode&gt; [&lt;CORBA-MinorCode&gt;] . . .</p> <p>&lt;MinorCode-範囲指定&gt; :</p> <p>&lt;CORBA-MinorCode&gt;-&lt;CORBA-MinorCode&gt;</p> <p>&lt;Error-Name&gt;: Web コンテナから返ってくる CORBA の Error-Name</p> <p>例 1:</p> <pre>worker.otxiiop.corba_error.timeout.err=CORBA::COM_FAILURE(3879;3883)</pre> <p>例 2:</p> <pre>worker.otxiiop.corba_error.transient.err=CORBA::TRANSIENT(3892-3895):</pre>	
<pre>worker.otxiiop.corba_error.&lt;Error-Name&gt;.doc</pre>	<p>worker.otxiiop.corba_error.list で指定された Error-Name に対応するエラードキュメント情報を定義します。</p> <p>書式:</p> <pre>worker.otxiiop.corba_error.&lt;Error-Name&gt;.doc=&lt;File URL&gt;</pre> <p>パラメータ説明</p> <p>&lt;Error-Name&gt;: 前述を参照</p> <p>&lt;File URL&gt;: "file://" で始まる URL (フルパスで指定します)</p> <p>例: Windows:</p> <pre>worker.otxiiop.corba_error.timeout.doc=file:///C:/WebOTX/WebServer2/error/timeout.html</pre> <p>Unix:</p> <pre>worker.otxiiop.corba_error.timeout.doc=file:///error/timeout.html</pre>	なし
<pre>worker.otxiiop.corba_error.charset</pre>	<p>エラードキュメントファイルのデフォルトのキャラクタセットを指定します。</p> <p>書式:</p> <pre>worker.otxiiop.corba_error.charset=&lt;charset&gt;</pre> <p>パラメータ説明</p> <p>&lt;charset&gt;: エラードキュメントのデフォルトのファイルキャラクタセット</p>	ISO-8859-1
<pre>worker.otxiiop.corba_error.&lt;Error-Name&gt;.charset</pre>	<p>指定した Corba エラー用のエラードキュメントファイルのキャラクタセットを指定します。</p> <p>書式:</p> <pre>worker.otxiiop.corba_error.&lt;Error-Name&gt;.charset=&lt;charset&gt;</pre> <p>パラメータ説明</p> <p>&lt;charset&gt;: Corba エラーのエラードキュメントのファイルキャラクタセット</p>	worker.otxiiop.corba_error.charset の値

### 5.6.3.IIOPプラグインの設定例

以下に IIOP プラグイン定義について設定例で説明します。

(1)環境設定を行うと `ior_workers.properties` は、以下の初期値に戻ります。

```
# TM_PLUGIN_IOR-start
worker.list=otxiop
otx_domain=D:/WebOTX/domains/domain1
worker.otxiop.type=otxiop
worker.otxiop.default_ior=localdomain
worker.otxiop.ior_def.localdomain=$(otx_domain)/config/httpgateway.ior
# TM_PLUGIN_IOR-end
```

※“`otx_domain`”はマクロ定義です。 `$(otx_domain)` として、パラメータに使用します。

(2) コンテキスト/`abc` を `hostA` の `domain1` に、/`xyz` を `hostB` の `domain2` に配備した場合

/`abc`、/`xyz` をそれぞれのドメインに振り分ける `ior_workers.properties` の設定内容の例は、次のとおりです。

- 1) `<Domain Dir>/config/iors` ディレクトリを作成する
- 2) `<Domain Dir>/config/iors/hostA_domain1` ディレクトリを作成する
- 3) `<Domain Dir>/config/iors/hostB_domain2` ディレクトリを作成する
- 4) `hostA` の `<WebOTX Dir>/domain1/config/httpgateway.ior` を 2)にコピーする
- 5) `hostB` の `<WebOTX Dir>/domain2/config/httpgateway.ior` を 3)にコピーする
- 6) 以下の記述を `ior_workers.properties` の “`TM_PLUGIN_IOR`” セクションの外に追加する

```
# TM_PLUGIN_IOR-start
...
# TM_PLUGIN_IOR-end
...
worker.otxiop.external_contexts=abc,xyz
worker.otxiop.context_ior.abc=hostA_domain1
worker.otxiop.context_ior.xyz=hostB_domain2
worker.otxiop.ior_def.hostA_domain1=$(otx_domain)/config/iors/hostA_domain1/httpgateway.ior
worker.otxiop.ior_def.hostB_domain2=$(otx_domain)/config/iors/hostB_domain2/httpgateway.ior
...
```

(3) IIOP プラグインの `loadbalancer` 機能を使い `hostA` の `domain1` と `hostB` の `domain2` に負荷分散させる場合、

`loadbalancer` 機能を使用するための `ior_workers.properties` の設定内容の例は、次のとおりです。

※本機能は単純なラウンドロビンによる負荷分散機能のみを提供しています。切り離しなど機能はありません。負荷分散装置による負荷分散を推奨します。

- 1) (2)の 1) - 5) を行う。
- 2) `ior_workers.properties` を、以下のとおり修正する。

```
# TM_PLUGIN_IOR-start
#worker.list=otxiop
otx_domain=D:/WebOTX/domains/domain1
#worker.otxiop.type=otxiop
#worker.otxiop.default_ior=localdomain
#worker.otxiop.ior_def.localdomain=$(otx_domain)/config/httpgateway.ior
# TM_PLUGIN_IOR-end
# ワーカ otxiop は worker.list の末尾に追加してください
worker.list=otxiop1,otxiop2,otxiop
worker.otxiop1.type=otxiop
worker.otxiop1.default_ior=hostA_domain1
worker.otxiop1.ior_def.hostA_domain1=$(otx_domain)/config/iors/hostA_domain1/httpgateway.ior

worker.otxiop2.type=otxiop
worker.otxiop2.default_ior=hostB_domain2
worker.otxiop2.ior_def.hostB_domain2=$(otx_domain)/config/iors/hostB_domain2/httpgateway.ior
worker.otxiop.type=lb
worker.otxiop.balanced_workers=otxiop1,otxiop2
```

3) プロセスグループの JavaVM オプション設定に以下の定義を追加する。

•hostA の domain1 のプロセスグループの JavaVM オプション

```
"-DjvmRoute=otxiop1"
```

•hostB の domain2 のプロセスグループの JavaVM オプション

```
"-DjvmRoute=otxiop2"
```



# 6.EJBコンテナに関する設定

EJB コンテナに関する設定について説明します。

## 6.1.EJBコンテナ設定項目・設定方法

EJB コンテナに関する設定項目には次のものがあります。

EJB インスタンスプール

EJB インスタンスキャッシュ

Entity Beanトランザクションコミットオプション

組み込み IIOP リスナ

それぞれ設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

### エディションによる違い

Standard/Enterprise Edition では EJB は TP モニタのプロセスグループ上で動作します。このため TP モニタの設定に影響されません。詳しくは TP モニタの設定の章を参照してください。

### 組み込みIIOPリスナについて

組み込み IIOP リスナとは、Standard-J Edition における EJB に対する要求や、JNDI サーバに対する RMI-IIOP のリクエストを処理するリスナです。これはエージェントプロセス内で動作します。EJB や JNDI サーバに対する通信ポート番号や SSL の使用の有無などを設定する場合は、この組み込み IIOP リスナの設定を変更してください。

Standard/Enterprise Edition における EJB の通信リスナは TP モニタの IIOP リスナとなりますのでそちらの設定に従います。このため、組み込み IIOP リスナの設定は JNDI サーバに対してのみ有効となります。

既定値では EJB コンテナと JNDI サーバに対する通信ポートは 7780 となります。

組み込み IIOP リスナの MO 名は「embedded-iiop-service」、それ以外の上記 3 つに対応する MO 名は「ejb-container」です。

## 6.2.EJBコンテナ設定項目一覧

以下は MO で設定可能な EJB コンテナ設定項目一覧です。

Dottedname : server.ejb-container

属性名 (attribute-name)	説明	既定値
steady-pool-size	Bean インスタンスの初期値を指定します。この値がプールで保持されます。0 から int 型の最大値を指定することができます。 (注:この項目は Standard-J Edition のみ設定可能です。)	0
pool-resize-quantity	サーバがプールを処理する時に作成または削除する Bean インスタンス数を指定します。0 から int 型の最大値を指定することができます。 (注:この項目は Standard-J Edition のみ設定可能です。)	8
max-pool-size	Bean インスタンスプールの最大値を指定します。0 から int 型の最大値を指定することができます。0 を指定するとプールサイズが無制限となります。	32

	(注:この項目は Standard-J Edition のみ設定可能です。)	
pool-idle-timeout-in-seconds	ステートレスセッション Bean や MDB がアイドル状態でプール内に存在できる最大時間を指定します。この時間を過ぎると Bean インスタンスは削除されます。	600 (s)
cache-resize-quantity	サーバがキャッシュを処理する時に作成または削除する Bean インスタンス数を指定します。0 から int 型の最大値を指定することができます。	32
max-cache-size	キャッシュに格納できる Bean インスタンスの最大値を指定します。0 を指定するとキャッシュできる Bean 数は無制限になります。	512
cache-idle-timeout-in-seconds	ステートフルセッション Bean や MDB がアイドル状態でキャッシュ内に存在できる最大時間を指定します。この時間を過ぎると Bean はバックアップストアで非活性化されます。	600 (s)
removal-timeout-in-seconds	ステートフルセッション Bean が非活性化される時間を指定します。この時間を過ぎてもアクセスされなかった場合、Bean は除されクライアントからアクセスできなくなります。	5400 (s)
victim-selection-policy	ステートフルセッション Bean のキャッシュから削除対象を特定するアルゴリズムを指定します。FIFO/LRU/NRU のいずれかを指定します。	nru
commit-option	エンティティ Bean のトランザクションコミットオプションの指定をします。B または C のいずれかを指定します。	B
session-store	非活性化された Bean がファイブとして保持される場所を指定します。	<code>/\${INSTANCE_ROOT}/session-store</code>

Dottedname : server.embedded-iiop-service

属性名 (attribute-name)	説明	既定値
enabled	組み込み IIOP リスナを有効にするかどうかを指定します。	true
port	暗号化しない通信を行うときに使用するポート番号を指定します。	7780 0 は自動割当を示す
ssl-port	SSL クライアント認証なしの通信を行うときに利用するポート番号を指定します。	null
ssl-auth-port	SSL クライアント認証ありの通信を行うときに利用するポート番号を指定します。	null
max-accept-number	同時クライアント接続数を指定します。	null (制限無し)
client-timeout	クライアント無通信監視タイマ時間(秒)を指定します。	0 (監視無し)
cert-key	サーバ側の SecureWare/セキュリティパックにて登録済みの鍵を特定する鍵 ID または JSSE で使用するサーバ証明書のファイル名を指定します。	null
cert-key-pass-phrase	JSSE で使用するサーバ証明書のパスワードを指定します。	null

cert-key-type	JSSE で使用するサーバ証明書の鍵の形式 (RSA, DSA など) を指定します。	null
cert-key-file-type	JSSE で使用するサーバ証明書のファイルの形式 (PKCS12 もしくは JKS) を指定します。	null
cert-alias	JSSE で使用するサーバ証明書の鍵の別名を指定します。	null
cert-alias-pass-phrase	JSSE で使用するサーバ証明書の鍵の別名のパスワードを指定します。	null
trusted-ca-certs	JSSE で使用する信頼する CA 証明書のファイル名を指定します。	null
trusted-ca-certs-pass-phrase	JSSE で使用する信頼する CA 証明書のファイルのパスワードを指定します。	null
trusted-ca-certs-file-type	JSSE で使用する信頼する CA 証明書のファイルの形式 (PKCS12 もしくは JKS) を指定します。	null

### 6.3. タイマー Bean の使用方法について

EJB 2.1 仕様ではコンテナ管理のタイマ機能が新たに追加されました。

WebOTX でこのタイマ機能を使用するためには、タイマ機能で使用するデータソースとデータベースのテーブルが必要となります。インストール時にはこれらは設定されていませんので、タイマ機能を使用するためには以下の手順でセットアップを行なう必要があります。

#### ① データソースの登録

タイマを使用するドメインに対して “jdbc/\_TimerPool” という名前のデータソースを追加してください。また対応する JDBC ドライバをドメインのクラスパスに追加してください。

#### ② テーブルの作成

[\$AS\_INSTALL]/lib/install/databases に Oracle 用と PointBase 用のテーブル作成用 SQL ファイルがありますので、これを使用して追加したデータソースで接続するデータベース上にテーブルを作成してください。他のデータベースを使用する場合もこの SQL を参考にして同じテーブル名、カラム名でテーブルを作成してください。

#### ③ ドメイン再起動

ドメインを再起動すると、タイマ機能が使用可能になります。

### 6.4. リソースのプロセス単位のロード設定 (Standard/Enterprise Edition の場合)

Standard/Enterprise Edition では EJB は Web コンテナとは別の Java VM 上で実行されます。また、プロセスグループが異なる EJB も別々の Java VM 上で実行されます。

ドメインに定義されたリソースアダプタ、コネクタリソース、JDBC データソースは既定値ではコンテナの動作している全 Java プロセス上でロードされます。例えば JDBC データソースの初期接続数を 1 に設定していても、Web コンテナ + EJB のプロセスグループのプロセス数分だけ初期接続が張られることとなります。そのリソースを使用していないプロセスでは無駄に接続が張られることになってしまいます。

この場合は、リソースアダプタ、コネクタリソース、JDBC データソース、プロセスグループの以下の属性を設定して使用するプロセスを絞り込んでください。プロセスグループの設定の方はリソースの配備、追加時には自動的に追加されます。

リソースアダプタ、コネクタリソース、JDBC データソースの属性

属性名 (attribute-name)	説明	既定値
use-jmx-agent	リソースを JMX エージェント上でロードするかどうかを指定します	Web コンテナがマル

		チプロセスモードで動作していれば false、そうでなければ true
use-all-ejb-processgroups	全ての EJB プロセスグループで使用するか、または各プロセスグループの属性でロードする、しないの指定を行なうかどうかを選択します。 全ての EJB プロセスグループで使用するを選択した場合、全プロセスグループのロードするリソースアダプタの設定に自動的にこのリソースが追加されます。	true

### プロセスグループの属性

属性名 (attribute-name)	説明	既定値
datasourceList	プロセスグループの起動時にロードするデータソースのリストを表示します。そのプロセスグループ上で動作する EJB が使用するデータソースの JNDI 名を選択してください。	Null
connectorResourceList	プロセスグループの起動時にロードするコネクタリソースのリストを表示します。そのプロセスグループ上で動作する EJB が使用するコネクタリソースの JNDI 名を選択してください。	Null
resourceAdapterList	プロセスグループの起動時にロードするリソースアダプタのリストを表示します。そのプロセスグループ上で動作する EJB が使用するリソースアダプタの名前を選択してください。	null

# 7. JNDIサービスに関する設定

JNDI サービスに関する設定について説明します。

## 7.1. JNDI設定項目・設定方法

JNDIサービスの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

## 7.2. JNDIサービス設定項目一覧

以下は MO で設定可能な JNDI サービス設定項目一覧です。

Dottedname : server.jndi-service

属性名 (attribute-name)	説明	既定値
bind-to-nameservice	JNDI サーバのリファレンスを CosNaming サーバに登録するかどうかを指定します。 EJB を使用する場合や、プロバイダ URL として corbaname 形式を指定する場合は true にする•必要があります。	Web Edition の場合: false それ以外: true
cnsurl	キャッシュ名前サーバを経由して JNDI サーバを使用する場合にキャッシュ名前サーバの URL のリストをカンマで区切って指定します。 例 corbaname://host1:9829, corbaname://host2:9829	null
cns-interval	CNS と連携する場合に、起動した後で CNS との通信を行う間隔 (秒) を指定します。	60
persisttent	名前情報のファイルへ永続化を行うかどうか指定します。	false
persistent-backupinterval	名前情報の永続化を行う場合、何回名前情報が更新されたら永続化ファイルをバックアップかを指定します。1 の場合名前情報が更新されるたびにバックアップを行いません。	1
persistent-maxsize	名前情報の永続化を行う場合に永続化ファイルの最大サイズ (Kbyte) を指定します。	0
persistent-storedir	名前情報をファイルに永続化する場合、永続化ファイルの格納先のディレクトリ名を指定します。	\${INSTANE_ROOT} }/config/jndis p/store
persistent-storedir-backup	名前情報をファイルに永続化する場合、バックアップ先のディレクトリ名を指定します。	\${INSTANE_ROOT} }/config/jndis p/store
remote	他のプロセスから JNDI にアクセスできるかどうかを指定します。 HTTP セッションレプリケーション機能、リモートインタフェースを持つ	true

	EJB、または JMS を使用する場合、リモートアクセス可能である必要があります。リモートアクセス可能の場合、Object Broker のサービス起動が必須となります。	
server-name	JNDI サーバを二重化する場合、それぞれの JNDI サーバを識別するための名前を指定します。指定がない場合は「ローカルホスト名:ドメイン名」が使用されます。 ドメインが正常に停止できなかった後でこの値が変わると JNDI サーバの動作が不正になる場合があります。 JNDI サーバが動作するホストが仮想ホスト名を持ち、その仮想ホスト名が変わる場合、この値を他のドメインと重ならない任意の固定文字列に設定してください。	null
startup-cosnaming-retry-interval	起動時に CosNaming サーバとの通信リトライを行う間隔 (秒) を指定します。	2
startup-cns-retry-interval	CNS と連携する場合に起動時に、時に CNS との通信リトライを行う間隔 (秒) を指定します。	1
url	連携する CosNaming サーバの URL のリストをカンマで区切って指定します。 例 corbaname://host1,corbaname://host2	null

### 7.3.JNDIサーバの通信リスナの設定

JNDI サービスは EJB と同じく、RMI-IIOP を使用して通信を行ないます。このため通信用ポートなどの設定は組み込み IIOP リスナの設定で行ないます。組み込み IIOP リスナについては EJB コンテナの設定方法を参照してください。

### 7.4.JNDIサーバを二重化するための設定

HTTP セッション情報のみをレプリケーションする場合は以下の手順は必要ありません。代わりに Web コンテナの session-replication-jndi-url 属性で設定します。詳しくは Web コンテナの設定方法を参照してください。

JNDI サーバの名前情報全体を複数ドメイン間で二重化するためには、それぞれのドメインで同じ CosNaming サーバと連携するように設定します。あらかじめ JNDI サーバの bind-to-nameservice 属性を true に設定しておく必要があります。

#### ① CosNaming サーバと連携する場合

それぞれのドメインの URL の設定で二重化するドメインの CosNaming サーバの URL を列挙します。

例えば server1 の port1 の CosNaming サーバの動作するドメインと server2 の port2 の CosNaming サーバの動作するドメイン間で二重化する場合、両方のドメインの url の設定値を以下のように設定します。

```
corbaname://server:port1,corbaname://server:port2
```

#### ② キャッシュ名前サーバと連携する場合

キャッシュ名前サーバと連携する場合、url にキャッシュ名前サーバでなく CosNaming サーバのポート番号を設定します。そして cnsurl にキャッシュ名前サーバの URL を指定します。

二重を行なうと、JNDI サーバへのバインドなどの名前を更新する処理は上記の CosNaming サーバの URL に登録されている全ての JNDI サーバに対して行なわれます。あるドメインの JNDI サーバを後から起動した場合でも、他の動作中の JNDI サーバから登録情報を取得して自身の登録情報にマージします。

二重化対象の JNDI サーバのリストは Object Broker の CosNaming サーバに登録されますが、ドメインを停止すると CosNaming

サーバ上のリストがクリアされます。

このため JNDI サーバを二重化している環境で、あるドメインを起動したまま別の二重化対象のドメインを再起動した場合、再起動したドメインの CosNaming サーバに対して停止しなかったドメインの JNDI サーバを再登録する必要があります。

再登録するには統合運用管理ツールで JNDI サービスのみ再起動を行いません。運用管理コマンドを使用する場合は以下のように行いません。この操作を行っても JNDI サーバに登録された情報は消えません。

```
otxadmin>invoke server.jndi-service.stop
```

```
otxadmin>invoke server.jndi-service.start
```

# 8. JMSに関する設定

JMS に関する設定について説明します。

## 8.1. JMS設定項目・設定方法

JMS サーバは独立したプロセスで実装しており、WebOTX の J2EE サーバ本体には、リソースアダプタにより接続連携しています。

JMS サーバや、JMS リソース、および、送信先の基本的な管理は、運用管理コマンド(otxadmin)、あるいは、統合運用管理ツールで行います。さらに詳細な管理を行うためには、JMS サーバが提供している専用の運用管理ツールを利用します。

運用管理コマンド、あるいは、統合運用管理ツールからの各項目の設定方法については、「2各サービスの属性の設定」を参照してください。また、JMSサーバが提供している専用の各運用管理ツールについては、「運用コマンドリファレンスマニュアル」-「JMS」を参照してください。

### 8.1.1. JMSサーバに対する設定

JMS サーバの運用に関するプロパティは、JMS サービス(jms-service)と、JMS ホスト(jms-host)の MO から設定できます。運用管理コマンドや、統合運用管理ツールでは、それぞれ以下のように指定します。

運用管理コマンド	:	server.jms-service server.jms-service.jms-host.default_JMS_host
統合運用管理ツール	:	WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.アプリケーションサーバ.JMS サービス WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.アプリケーションサーバ.JMS サービス ス.JMS ホスト.default_JMS_host

JMS サーバは、①～④のプロパティを取り込みます。優先順位は、後ろのものほど高くなっています。

#### ① default.properties ファイル

プロパティのデフォルト値が定義されたファイルで、次の位置にあります。

`${AS_INSTALL}/wojms/lib/props/broker/default.properties`

**注意**：利用者は直接編集を行わないで下さい。

#### ② install.properties ファイル

インストール時に指定されたプロパティが格納されるファイルで、次の位置にあります。

`${AS_INSTALL}/wojms/lib/props/broker/install.properties`

**注意**：利用者は直接編集を行わないで下さい。

#### ③ config.properties ファイル

ドメイン固有の JMS サーバに対するプロパティ設定を記述するファイルで、次の位置にあります。ファイルはドメイン生成時に作成されます。

`${INSTANCE_ROOT}/wojms/instances/<JMS サーバインスタンス識別子>/props/config.properties`

**備考**：<JMS サーバインスタンス識別子>の既定値は「wojmsbroker」です。

#### ④ JMS サーバの起動引数(server.jms-service.start-args)

起動引数で指定したプロパティは、すべてのプロパティ設定ファイルに対して優先されます。



## 8.1.2.コネクションファクトリに対する設定

### コネクションファクトリリソース

コネクションファクトリリソースは、J2EE アプリケーションから JNDI を通して利用します。項目の設定や、参照を行う場合、運用管理コマンドや、統合運用管理ツールでは、それぞれ以下のように指定します。

「*connection-factory-name*」は、コネクションファクトリリソースの定義名で、JNDI の登録名になります。

運用管理コマンド	:	<code>server.resources.jms-resource.jms-connection-factory.<i>connection-factory-name</i></code>
統合運用管理ツール	:	WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.リソース.JMS リソース.コネクションファクトリリソース. <i>connection-factory-name</i>

## 8.1.3.送信先に対する設定

### 送信先

物理的な送信先の設定や、項目の参照を行う場合、運用管理コマンドや、統合運用管理ツールでは、それぞれ以下のように指定します。「*physical-destination-name*」は、送信先の定義名です。

運用管理コマンド	:	<code>server.jms-service.jms-physical-destination.<i>physical-destination-name</i></code>
統合運用管理ツール	:	WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.アプリケーションサーバ.JMS サービス.送信先名. <i>physical-destination-name</i>

### 送信先リソース

送信先リソースは、J2EE アプリケーションから JNDI を通して利用します。項目の設定や、参照を行う場合、運用管理コマンドや、統合運用管理ツールでは、それぞれ以下のように指定します。

「*logical-destination-name*」は、送信先リソースの定義名で、JNDI の登録名になります。

運用管理コマンド	:	<code>server.resources.jms-resource.jms-logical-destination.<i>logical-destination-name</i></code>
統合運用管理ツール	:	WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.リソース.JMS リソース.送信先リソース. <i>logical-destination-name</i>

## 8.2.JMS設定項目一覧

運用管理コマンド、統合運用管理ツールで制御できる MO の属性と、そのベースとなる JMS のプロパティ/属性一覧を示します。

### 8.2.1.MOの属性一覧

JMS に関連する MO のうち、属性を設定できるものは、以下の 5 種類です。以降、各 MO について設定できる属性を説明します。

JMS サービス(jms-service)

JMS ホスト(jms-host)

コネクションファクトリリソース(jms-connection-factory)

送信先(jms-physical-destination)

送信先リソース(jms-logical-destination)

## JMSサービス

連携する JMS サーバに関する情報を定義します。

この定義は、運用管理コマンドの create-domain によるドメイン生成時に自動的に設定されます。

MO の dottedname は「server.jms-service」です。設定可能な属性や、操作については、「MO 定義リファレンス」-「jms-service」を参照してください。

なお、jms-service の「プロパティ」として設定できるのは、以下の項目です。

プロパティ名	説明	既定値
auto-shutdown	ドメイン停止時に JMS サーバを停止するかどうかを指定します。 true: 停止する false: 停止しない	true
instance-name	JMS サーバインスタンス識別子を指定します。 ※ 同一ホストで JMS サーバクラスタを構成する場合は、この値をホスト内で一意にしておく必要があります。	wojmsbroker

## JMSホスト

JMS サーバの起動、JMS サーバへの接続に必要な情報を定義します。

この定義は、create-domain コマンドによるドメイン生成時に自動的に設定されます。ポート番号については、create-domain コマンドの --domainproperties オプションで、jms.port として指定することができます。

MO の dottedname は「server.jms-service.jms-host.default\_JMS\_host」です。設定可能な属性や、操作については、「運用編」-「MO 定義リファレンス」-「jms-host」を参照してください。

## コネクションファクトリリソース

JNDI を通してアクセスするコネクションファクトリリソースを定義します。

この定義は、運用管理コマンドの create-jms-resource、統合運用管理ツールでは、「WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.リソース.JMS リソース」の「コネクションファクトリリソースの作成」操作で作成されます。コネクションファクトリリソースの作成方法については、「運用編」-「運用と操作」-「JMS の運用操作」-「JMS リソースの操作」を参照してください。

コネクションファクトリリソース(jms-connection-factory)作成時には、コネクタリソース(connector-resource)、コネクタコネクションプール(connector-connection-pool)、および、リソース参照(resource-ref)を自動的に作成し、属性値を設定します。

属性値を変更する場合は、コネクションファクトリリソースの MO を介して行います。

各 MO の dottedname は以下のとおりです。

コネクションファクトリリソース	server.resources.jms-resource.jms-connection-factory.connection-factory-name
コネクタリソース	server.resources.connector-resource.connection-factory-name
コネクタコネクションプール	server.resources.connector-connection-pool.connection-factory-name
リソース参照	server.resource-ref.connection-factory-name

設定可能な属性や、操作については、「運用編」-「MO 定義リファレンス」を参照してください。

### 注意：

コネクションファクトリリソース作成時、「プロパティ」に属性と同じ名前の項目を指定した場合は、「プロパティ」に指定されたものが優先されます。

## 送信先

物理的な送信先を定義します。

この定義は、運用管理コマンドの `create-jmsdest`、統合運用管理ツールでは、「WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.アプリケーションサーバ.JMS サービス」の「送信先作成」操作で作成されます。送信先の作成方法については、「運用編」-「運用と操作」-「JMS の運用操作」-「送信先の操作」を参照してください。

MO の dottedname は「`server.jms-service.jms-physical-destination.destination-name`」です。設定可能な属性や、操作については、「運用編」-「MO 定義リファレンス」-「`jms-physical-destination`」を参照してください。

### 注意：

送信先作成時、「プロパティ」に属性と同じ名前の項目を指定した場合は、「プロパティ」に指定されたものが優先されます。

## 送信先リソース

JNDI を通してアクセスする送信先リソースを定義します。

この定義は、運用管理コマンドの `create-jmsresource`、統合運用管理ツールでは、「WebOTX 管理ドメイン[<ホスト名>].<ドメイン名>.リソース.JMS リソース」の操作である「送信先リソースの作成」で作成されます。送信先リソースの作成方法については、「運用編」-「運用と操作」-「JMS の運用操作」-「JMS リソースの操作」を参照してください。

送信先リソース(`jms-logical-destination`)作成時には、管理オブジェクトリソース(`admin-object-resource`)と、リソース参照(`resource-ref`)を自動的に作成し、属性値を設定します。

属性値を変更する場合は、送信先リソースの MO を介して行います。

各 MO の dottedname は以下のとおりです。

送信先リソース	<code>server.resources.jms-resource.jms-logical-destination.logical-destination-name</code>
管理オブジェクトリソース*	<code>server.resources.admin-object-resource.logical-destination-name</code>
リソース参照	<code>server.resource-ref.logical-destination-name</code>

設定可能な属性や、操作については、「運用編」-「MO 定義リファレンス」を参照してください。

### 注意：

送信先リソース作成時、「プロパティ」に属性と同じ名前の項目を指定した場合は、「プロパティ」に指定されたものが優先されません。

## 8.2.2.JMSのプロパティ／属性一覧

ここでは、以下のプロパティや属性について説明します。

### JMSサーバのプロパティ

「8.1JMS設定項目・設定方法」-「8.1.1JMSサーバに対する設定」で説明した、「③config.propertiesファイル」に設定できるプロパティです。以下のカテゴリに分けて説明しています。

- JMS サーバ全体のプロパティ
- コネクションサービスに関するプロパティ
- セキュリティに関するプロパティ
- 永続ストアに関するプロパティ
- ログ出力に関するプロパティ
- メモリ管理に関するプロパティ
- 自動生成送信先に関するプロパティ
- クライアント監視に関するプロパティ
- JMS サーバクラスタに関するプロパティ

### JMS クライアントのプロパティ

非 J2EE 環境下で動作する Java アプリケーションに有効であり、Java システムプロパティとして設定します。

## JMSサーバのプロパティ

「config.properties」ファイルに設定するプロパティの一覧を示します。このファイルには、JMS サービス(jms-service)や、JMS ホスト(jms-host)の MO から設定できない詳細な項目を指定します。「config.properties」ファイルの変更は、JMS サーバを停止した状態で行ってください。

ファイルの格納場所、設定方法については、「8.1JMS設定項目・設定方法」を参照してください。

ただし、以下のプロパティについては、JMS サービス、あるいは、JMS ホストの MO の属性値が優先されます。これらの値を設定する場合は各 MO の属性値を設定してください。

JMS サーバのプロパティ名	対応する MO の属性
wojms.portmapper.port	server.jms-service.jms-host.default_JMS_host.port
wojms.system.max_count	server.jms-service.systemMaxCount
wojms.system.max_size	server.jms-service.systemMaxSize
wojms.message.max_size	server.jms-service.messageMaxSize
wojms.log.level	server.jms-service.loglevel
wojms.log.file.rolloverbytes	server.jms-service.logfileRolloverBytes
wojms.log.file.rolloversecs	server.jms-service.logfileRolloverSecs
wojms.autocreate.topic	server.jms-service.autocreateTopic
wojms.autocreate.queue	server.jms-service.autocreateQueue
wojms.autocreate.queue. maxNumActiveConsumers	server.jms-service.autocreateMaxNumActiveConsumers
wojms.autocreate.queue. maxNumBackupConsumers	server.jms-service.autocreateMaxNumBackupConsumers
wojms.redelivery.delay	server.jms-service.wojmsRedeliveryDelay
wojms.redelivery.limit	server.jms-service.wojmsRedeliveryLimit
wojms.redelivery.destination	server.jms-service.wojmsRedeliveryDestination
wojms.expiration.destination	server.jms-service.wojmsExpirationDestination
wojms.client_metrics.enabled	server.jms-service.enableClientMetrics
wojms.display.message.count	server.jms-service.displayMessageCount
wojms.log.file.filename	server.jms-service.logfileName
wojms.log.admin.output	server.jms-service.logAdmin
wojms.log.admin.rolloverbytes	server.jms-service.logfileRolloverBytesAdmin
wojms.log.admin.rolloversecs	server.jms-service.logfileRolloverSecsAdmin
wojms.log.admin.filename	server.jms-service.logfileNameAdmin
wojms.log.error.output	server.jms-service.logError
wojms.log.error.rolloverbytes	server.jms-service.logfileRolloverBytesError
wojms.log.error.rolloversecs	server.jms-service.logfileRolloverSecsError
wojms.log.error.filename	server.jms-service.logfileNameError
wojms.log.message.output	server.jms-service.logMessage

wojms.log.message.rolloverbytes	server.jms-service.logfileRolloverBytesMessage
wojms.log.message.rolloversecs	server.jms-service.logfileRolloverSecsMessage
wojms.log.message.filename	server.jms-service.logfileNameMessage
wojms.log.packet.output	server.jms-service.logPacketAdmin server.jms-service.logPacketMessage server.jms-service.logPacketPing server.jms-service.logPacketCluster
wojms.log.packet.rolloverbytes	server.jms-service.logfileRolloverBytesPacket
wojms.log.packet.rolloversecs	server.jms-service.logfileRolloverSecsPacket
wojms.log.packet.filename	server.jms-service.logfileNamePacket
wojms.enable_cluster	server.jms-service.enableCluster
wojms.cluster.brokerlist	server.jms-service.clusterBrokerList
wojms.cluster.masterbroker	server.jms-service.clusterMasterBroker

### JMS サーバ全体のプロパティ

プロパティ名	説明	既定値
wojms.portmapper.port	JMS サーバのポート番号を指定します。同一ホストで複数の JMS サーバのインスタンスを実行する場合、それぞれ固有のポート番号を割り当てる必要があります。	9700
wojms.portmapper.backlog	要求を拒否せずに、同時に処理可能な要求の最大数を指定します。	50
wojms.producer.maxBatch	プロデューサが一度に送信できるメッセージの最大数を指定します。	1000
wojms.message.expiration.interval	有効期限が切れたメッセージを廃棄する間隔を秒単位で指定します。	60
wojms.system.max_count	JMS サーバが保持するメッセージの最大数を指定します。この値を超えるメッセージは拒否されます。-1 は無制限です。	-1
wojms.system.max_size	JMS サーバが保持するメッセージの最大のサイズ (バイト、K バイト、または M バイト単位) を指定します。この値を超えるメッセージは拒否されます。  128,128b は 128 バイトを、128k は 128K バイト (128 × 1024=131072 バイト)、128m は 128M バイト (128 × 1024 × 1024=134217728 バイト) を表します。-1 は無制限です。	-1
wojms.message.max_size	メッセージの本体の最大許容サイズ (バイト、K バイト、または M バイト単位) を指定します。このサイズを超えるメッセージは拒否されます。  128,128b は 128 バイトを、128k は 128K バイト (128 × 1024=131072 バイト)、128m は 128M バイト (128 × 1024 × 1024=134217728 バイト) を表します。-1 は無制限です。	70m

wojms.display.message.count	メッセージ一覧の表示で、出力するメッセージの最大数を指定します。	1000
wojms.redelivery.delay	メッセージ再配信の遅延時間を秒単位で指定します。0 は、待ち合わせ無しです。	0
wojms.redelivery.limit	メッセージ再配信回数の上限を指定します。-1 は無制限、0 は再配信無しです。再配信回数の上限を超過したメッセージは破棄されます。	-1
wojms.redelivery.destination	再配信回数を超過したメッセージを破棄せずに転送する送信先名を指定します。ここに指定する送信先は、あらかじめ作成しておく必要があります(自動的に作成されません)。指定した送信先が存在しない場合、メッセージは破棄されます。	
wojms.expiration.destination	有効期限切れメッセージを破棄せずに転送する送信先名を指定します。ここに指定する送信先は、あらかじめ作成しておく必要があります(自動的に作成されません)。指定した送信先が存在しない場合、メッセージは破棄されます。	

### コネクションサービスに関するプロパティ

プロパティ名	説明	既定値
wojms.service.activelist	JMS 起動時に活性化するサービス群を指定します。 jms: JMS のコネクションサービス ssljms: JMS の SSL コネクションサービス admin: 管理用のコネクションサービス ssladmin: 管理用の SSL コネクションサービス	jms, admin
wojms.shared. connectionMonitor_limit	threadpool_model に shared を選択した場合に、コネクションを監視するスレッドの対象コネクションの最大数を指定します。-1 は無制限です。	Windows: 64 Unix: 512
wojms.<svc>.min_threads	svc で示したコネクションサービスが使用するスレッドプールの最小数を指定します。 既定値はコネクションサービスにより異なります。	jms: 10 ssljms: 10 admin: 4 ssladmin: 4
wojms.<svc>.max_threads	svc で示したコネクションサービスが使用するスレッドプールの最大数を指定します。 既定値はコネクションサービスにより異なります。 0 より大きく、かつ min_threads で指定した値より大きくする必要があります。	jms: 1000 ssljms: 500 admin: 10 ssladmin: 10
wojms.<svc>.threadpool_model	svc で示したコネクションサービスが使用するスレッドプールモデルを指定します。 dedicated: コネクション毎に専用のスレッドを割り当てる。 shared: 複数のコネクション間でスレッドを共有する。共有するコネク	dedicated

	<p>シヨンの数は <code>connectionMonitor_limit</code> で指定する。このモデルは <code>jms</code> と <code>admin</code> の場合のみに指定できる。</p> <p>※ <code>jms</code> コネクションサービスのみは、<code>jms-service</code> の「<code>useSharedThread</code>」属性を <code>true</code> にすることで、<code>shared</code> にすることができます。</p>	
<code>wojms.&lt;svc&gt;.tcp.hostname</code>	<p><code>svc</code> で示したサービスで使用するホスト名、もしくは IP アドレスを指定します。複数のネットワークインタフェース (NIC) を持つような場合に固定することができます。指定がない場合は、任意を意味します。</p>	
<code>wojms.&lt;svc&gt;.tcp.port</code>	<p><code>svc</code> で示したサービスで使用するポート番号を指定します。</p> <p>0 は、動的アサインを意味します。</p>	0

### セキュリティに関するプロパティ

プロパティ名	説明	既定値
<code>wojms.keystore.file.dirpath</code>	<p>SSL ベースのサービスの場合は、キーストアファイルが配置されているディレクトリへのパスを指定します。</p>	<code>\${AS_INSTALL}/wojms/etc</code>
<code>wojms.keystore.file.name</code>	<p>SSL ベースのサービスの場合は、キーストアファイル名を指定します。</p>	<code>keystore</code>
<code>wojms.keystore.password</code>	<p>SSL ベースのサービスの場合に、キーストアのパスワードを指定します。</p> <p>※パスワードはさまざまな方法で指定できます。最も安全な方法は、JMS サーバのプロンプトでパスワードを入力することです。安全性は低くなりますが、パスファイルを使用してパスファイルを読み取り保護することもできます。安全性は最も低くなりますが、次のコマンド行オプションを使用してパスワードを指定することもできます。</p> <p><code>wojmsbrokerd -password</code></p>	
<code>wojms.passfile.enabled</code>	<p>セキュリティ保護される通信用 (SSL、JDBC) のユーザーパスワードをパスファイルで指定するかどうかを指定します。</p> <p><code>true</code>:パスファイルで指定する <code>false</code>:パスファイルで指定しない</p>	<code>false</code>
<code>wojms.passfile.dirpath</code>	<p>パスファイルが配置されているディレクトリへのパスを指定します。</p>	<code>\${AS_INSTALL}/wojms/etc</code>
<code>wojms.passfile.name</code>	<p>パスファイルのファイル名を指定します。</p>	<code>passfile</code>
<code>wojms.authentication.client.response.timeout</code>	<p>JMS サーバからの認証要求に対するクライアントの応答を待機する時間を秒単位で指定します。</p>	180
<code>wojms.accesscontrol.enabled</code>	<p>JMS サーバでサポートされるすべてのコネクションサービスに対して、アクセス制御を行うかどうかを設定します。</p> <p><code>true</code>:アクセス制御を行う</p>	<code>true</code>

	<p>false: アクセス制御を行わない</p> <p>アクセス制御プロパティファイルに指定されているとおり、認証されたユーザがコネクションサービスを使用するためのアクセス権を保持しているかどうか、あるいは特定の送信先に対して特定の JMS 操作を実行するためのアクセス権を保持していることをシステムでチェックするかどうかを指定します。</p>	
<p>wojms.&lt;svc&gt;.accesscontrol.enabled</p>	<p>svc で指定したコネクションサービスに対して、アクセス制御を行うかどうかを設定し、JMS サーバ全体の設定 (wojms.accesscontrol.enabled) をオーバーライドします。</p> <p>true: アクセス制御を行う</p> <p>false: アクセス制御を行わない</p> <p>アクセス制御プロパティファイルに指定されているとおり、認証されたユーザが、指定したコネクションサービスを使用するためのアクセス権を保持しているか、あるいは特定の送信先に対して特定の JMS 操作を実行するためのアクセス権を保持していることをシステムでチェックするかどうかを指定します。</p>	
<p>wojms.accesscontrol.file.dirpath</p>	<p>アクセス制御プロパティファイルが配置されているディレクトリへのパスを指定します。</p>	<p>\${wojms.instance.shome}/\${wojms.instance.name}/\${}</p>
<p>wojms.accesscontrol.file.filename</p>	<p>JMS サーバインスタンスでサポートされるすべてのコネクションサービスに対するアクセス制御プロパティファイルの名前を指定します。ファイル名には、アクセス制御ディレクトリからの相対ファイルパスを指定します。</p>	<p>accesscontrol.properties</p>
<p>wojms.&lt;svc&gt;.accesscontrol.file.filename</p>	<p>JMS サーバインスタンスの、svc で指定したコネクションサービスに対するアクセス制御プロパティファイルの名前を指定します。ファイル名には、アクセス制御ディレクトリからの相対ファイルパスを指定します。</p>	
<p>wojms.authentication.provider</p>	<p>JMS サーバに接続するユーザの管理方式を指定します。</p> <p>APSERVER: WebOTX ドメインのユーザ管理方式を使用します。</p> <p>WOJMS: JMS 固有の管理方式を使用します。</p>	<p>APSERVER</p>
<p>wojms.authentication.type</p>	<p>JMS クライアントからパスワードを送信する際のタイプを指定します。</p> <p>basic: Base64 でエンコーディングしたパスワードを送信します。</p> <p>wojms.authentication.provider を指定しない場合や、APSERVER を指定する場合に使用します。</p> <p>digest: MD5 ダイジェストでハッシュ化したパスワードを Base64 でエンコーディングしたパスワードを送信します。</p> <p>wojms.authentication.provider で WOJMS を指定した場合のみ使用できます。</p>	<p>basic</p>



wojms.authentication.basic.user_repository	Basic 認証に使用されるユーザリポジトリのタイプを指定します。 file:JMS 認証(WOJMS)においてファイルベースでの認証を使用します。 jaas:ドメイン認証(APSERVER)に対して、JAAS(Java Authentication Authorization Service)での認証を使用します。	jaas
wojms.authentication.digest.user_repository	MD5 ダイジェスト認証に使用されるユーザリポジトリのタイプを指定します。 file:JMS 認証(WOJMS)においてファイルベースでの認証を使用します。 jaas:ドメイン認証(APSERVER)では MD5 ダイジェスト認証は使用できません。	jaas
wojms.authentication.jaas.realm	ドメイン認証(APSERVER)で使用する auth-realm 名を指定してください。 省略した場合は、デフォルトレムが使用されます。	

### 永続ストアに関するプロパティ

プロパティ名	説明	既定値
wojms.persist.store	組み込みのファイルストア、またはプラグインの JDBC ストアのどちらを JMS サーバが使用するかを指定します。 file:ファイルストア jdbc:JDBC ストア	file
wojms.persist.file.message_max_record_size	組み込みのファイルストアで、メッセージストレージファイル(個別のファイルではない)に追加されるメッセージの最大サイズ(バイト、K バイト、または M バイト単位)を指定します。 128,128b は 128 バイトを、128k は 128K バイト(128 × 1024=131072 バイト)、128m は 128M バイト(128 × 1024 × 1024=134217728 バイト)を表します。	1m
wojms.persist.file.message_filepool.cleanratio	組み込みのファイルストアの場合に、クリーン状態(サイズを 0 にする)で保持される送信先のファイルプールの空きファイルの割合(%)を指定します。この値が大きいほど、作業中にファイルを削除するのに必要なオーバーヘッドが増えますが、ファイルプールに必要なディスク容量は小さくなります。	0
wojms.persist.file.destination_message_filepool.limit	組み込みのファイルストアの場合に、送信先のファイルプールで再利用できる空きファイルの最大数を指定します。この値が大きいほど、JMS サーバが永続データを処理する速度が速くなります。この値を超える空きファイルは削除されます。	100

wojms.persist.file. message.cleanup	組み込みのファイルストアで、JMS サーバのシャットダウン時に、送信先のファイルプール内の空きファイルを削除するかどうかを指定します。値を false に設定すると、JMS サーバのシャットダウンが速くなりますが、ファイルを格納するためのディスク容量がさらに必要になります。	false
wojms.persist.file. sync.enabled	永続処理でメモリ内の状態を物理的なストレージと同期させるかどうかを指定します。true の場合、システムクラッシュによるデータ損失は回避されるますが、永続処理のパフォーマンスに負荷がかかります。	false
wojms.persist.jdbc.driver	データベースに接続する JDBC ドライバの Java クラス名を指定します。	
wojms.persist.jdbc. opendburl	既存データベースへの接続を開くためのデータベース URL を指定します。	
wojms.persist.jdbc.brokerid (省略可能)	複数の JMS サーバインスタンスが、永続データストアとして、同じデータベースを使用する場合、データベーステーブル名を一意にするために、データベーステーブル名に追加される JMS サーバインスタンス識別子を指定します。通常、1 つの JMS サーバインスタンスだけのデータを格納する組み込みデータベースでは不要です。識別子には、英数字を使用し、データベースで許可されているテーブル名の最大数 12 を超えないようにする必要があります。	
wojms.persist.jdbc. createdburl (省略可能)	データベースを作成する接続を開くためのデータベース URL を指定します。wojmsdbmgr を使用して、データベースを作成する場合にだけ指定します。	
wojms.persist.jdbc. closedburl (省略可能)	JMS サーバをシャットダウンする場合に、現在のデータベース接続をシャットダウンするためのデータベース URL を指定します。	
wojms.persist.jdbc.user (省略可能)	必要に応じて、データベース接続を開くときに使用するユーザ名を指定します。セキュリティ上の理由から、この代わりに、次のコマンド行オプションを使用して値を指定できます。  wojmsbrokerd -dbuser  wojmsdbmgr -u	
wojms.persist.jdbc.password	データベース接続を開くときに使用するパスワードを指定します。  ※パスワードは、さまざまな方法で指定できます。最も安全な方法は、JMS サーバのプロンプトでパスワードを入力することです。安全性は低くなりますが、passfile を使用して passfile を読み取り保護することもできます。安全性はもっとも低くなるが、次のコマンド行オプションを使用してパスワードを指定することもできます。	

	<pre>wojmsbrokerd -dbpassword wojmsdbmgr -p</pre>	
<pre>wojms.persist.jdbc. needpassword (省略可能)</pre>	<p>データベースで JMS サーバのアクセスにパスワードを必要とするかどうかを指定します。値を true にすると、パスワードが必要になります。パスワードは、次のコマンド行オプションを使用して指定できます。</p> <pre>wojmsbrokerd -dbpassword wojmsdbmgr -p</pre> <p>コマンド行オプション、またはパスファイルのどちらかを使用してパスワードを指定しないと、JMS サーバによってパスワードの入力が要求されます。</p>	false
<pre>wojms.persist.jdbc.retrycount</pre>	JDBC ストア利用時のデータベースへの再接続試行回数を指定します。	10
<pre>wojms.persist.jdbc.retryinterval</pre>	JDBC ストア利用時のデータベースへの再接続遅延時間を秒単位で指定します。	10
<pre>wojms.persist.jdbc.table. WOJMSSV35</pre>	バージョンテーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSSCREC35</pre>	設定変更レコードテーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSDEST35</pre>	送信先テーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSINT35</pre>	配信対象テーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMMSG35</pre>	メッセージテーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSPROPS35</pre>	プロパティテーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSILIST35</pre>	配信対象の状態テーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSTXN35</pre>	トランザクションテーブルを作成するための SQL コマンドを指定します。	
<pre>wojms.persist.jdbc.table. WOJMSTACK35</pre>	トランザクション通知テーブルを作成するための SQL コマンドを指定します。	

## ログ出力に関するプロパティ

プロパティ名	説明	既定値
wojms.log.level	ログレベルを指定します。指定できるレベルは高いものから順に、NONE, ERROR, WARNING, INFO, DEBUG, DEBUGMED, DEBUGHIGH です。 あるカテゴリを指定すると、指定したカテゴリとそれより上のレベルのカテゴリが含まれます。	INFO
wojms.log.timezone	ログのタイムスタンプのタイムゾーンを指定します。識別子は、 <code>java.util.TimeZone.getTimeZone()</code> が使用しているものと同じです。	
wojms.log.file.rolloverbytes	ログの出力先ファイルを切り替えるファイルサイズをバイト単位で指定します。-1 は出力先の切り替えを行いません。	1048576
wojms.log.file.rolloversecs	ログの出力先ファイルを切り替える間隔を秒単位で指定します。-1 は出力先の切り替えを行いません。	-1
wojms.log.file.dirpath	ログファイルが格納されているディレクトリへのパスを指定します。デフォルト値は、 <code>install.properties</code> ファイルに定義されています。	<code>\${wojms.instance.shome}/../../logs/wojms</code>
wojms.log.file.filename	ログファイル名を指定します。デフォルト値は、 <code>install.properties</code> ファイルに定義されています。	<code>wojmsserver.log</code>
wojms.log.file.filenum	ログファイルのバックアップ数を指定します。	9
wojms.log.file.output	ログファイルに書き込むロギング情報のカテゴリを指定します。指定できる値は、縦線 ( ) で区切ったロギングのカテゴリ (ERROR, WARNING, INFO, DEBUG, DEBUGMED, DEBUGHIGH) のセットか、ALL または NONE です。 ALL:すべてのカテゴリを出力 NONE:出力しない	ALL
wojms.log.console.stream	コンソールの出力先を指定します。 OUT:標準出力 ERR:標準エラー出力	ERR
wojms.log.console.output	コンソールへ書き込むロギング情報のカテゴリを指定する。指定できる値は、縦線 ( ) で区切ったロギングのカテゴリ (ERROR, WARNING, INFO, DEBUG, DEBUGMED, DEBUGHIGH) のセットか、ALL または NONE です。 ALL:すべてのカテゴリを出力 NONE:出力しない	ERROR WARNING
wojms.log.admin.rolloverbytes	運用管理操作履歴を出力するファイルを切り替えるサイズをバイト単位で指定します。-1 は出力先の切り替えを行いません。	1048576
wojms.log.admin.rolloversecs	運用管理操作履歴を出力するファイルを切り替える時間を秒単位で	-1 (切り替えなし)

	指定します。-1 は出力先の切り替えを行いません。	
wojms.log.admin.dirpath	運用管理操作履歴の出力ディレクトリへのパスを指定します。	\${wojms.log.file.dirpath}
wojms.log.admin.filename	運用管理操作履歴のファイル名を指定します。	wojmsadmin.log
wojms.log.admin.filenum	運用管理操作履歴ファイルのバックアップ数を指定します。	9
wojms.log.admin.output	運用管理操作履歴へ出力するログカテゴリを指定します。指定できる値は、ADMIN または NONE です。	ADMIN
wojms.log.packet.rolloverbytes	パケットログを出力するファイルを切り替えるサイズをバイト単位で指定します。-1 は出力先の切り替えを行いません。	1048576
wojms.log.packet.rolloversecs	パケットログを出力するファイルを切り替える時間を秒単位で指定します。-1 は出力先の切り替えを行いません。	-1 (切り替えなし)
wojms.log.packet.dirpath	パケットログの出力ディレクトリへのパスを指定します。	\${wojms.log.file.dirpath}
wojms.log.packet.filename	パケットログのファイル名を指定します。	wojmspacket.log
wojms.log.packet.filenum	パケットログファイルのバックアップ数を指定します。	9
wojms.log.packet.output	パケットログへ出力するログカテゴリを指定します。指定できる値は、縦線 ( ) で区切ったロギングのカテゴリ (PACKET_MESSAGE、PACKET_PING、PACKET_ADMIN、PACKET_CLUSTER_MESSAGE、NONE) のセットか、ALL または NONE です。 PACKET_MESSAGE: JMS メッセージの送受信でやり取りされるパケット (制御用のパケットを含む) の情報を出力 PACKET_PING: JMS サーバや、JMS クライアントの起動状態確認で送受信される PING パケットの情報を出力 PACKET_ADMIN: 運用管理操作で送受信されるパケットの情報を出力 PACKET_CLUSTER_MESSAGE: JMS サーバクラスタ構成で JMS サーバ間で送受信されるパケットの情報を出力 NONE: 出力しない	NONE
wojms.log.message.rolloverbytes	メッセージライフサイクルログを出力するファイルを切り替えるサイズをバイト単位で指定します。-1 は出力先の切り替えを行いません。	1048576
wojms.log.message.rolloversecs	メッセージライフサイクルログを出力するファイルを切り替える時間を秒単位で指定します。-1 は出力先の切り替えを行いません。	-1 (切り替えなし)
wojms.log.message.dirpath	メッセージライフサイクルログの出力ディレクトリへのパスを指定します。	\${wojms.log.file.dirpath}
wojms.log.message.filename	メッセージライフサイクルログのファイル名を指定します。	wojmsmessage.log
wojms.log.message.filenum	メッセージライフサイクルログファイルのバックアップ数を指定します。	9
wojms.log.message.output	メッセージライフサイクルログへ出力するログカテゴリを指定します。指定できる値は、縦線 ( ) で区切ったロギングのカテゴリ	NONE

	(MESSAGE) か、ALL または NONE です。	
wojms.log.error.rolloverbytes	エラーログを出力するファイルを切り替えるサイズをバイト単位で指定します。-1 は出力先の切り替えを行いません。	1048576
wojms.log.error.rolloversecs	エラーログを出力するファイルを切り替える時間を秒単位で指定します。-1 は出力先の切り替えを行いません。	-1 (切り替えなし)
wojms.log.error.dirpath	エラーログの出力ディレクトリへのパスを指定します。	\${wojms.log.file.dirpath}
wojms.log.error.filename	エラーログのファイル名を指定します。	wojmserror.log
wojms.log.error.fileenum	エラーログファイルのバックアップ数を指定します。	9
wojms.log.error.output	エラーログへ出力するログカテゴリを指定します。指定できる値は、縦線 ( ) で区切ったロギングのカテゴリ (ERROR、WARNING、DUMP_STATE、NONE) のセットか、ALL または NONE です。	ERROR WARNING DUMP_STATE NONE
wojms.metrics.enabled	メトリクス情報を収集するかどうかを指定します。 true: 収集する false: 収集しない	true
wojms.metrics.interval	メトリクス情報が報告される間隔を秒単位で指定します。-1 は報告しません。	-1

### メモリ管理に関するプロパティ

プロパティ名	説明	既定値
wojms.green.threshold	メモリリソースの Green レベル (使用可能なメモリが十分にある状態) のスタート位置をメモリの利用率 (%) で指定します。	0
wojms.yellow.threshold	メモリリソースの Yellow レベル (メモリーが減っている状態) のスタート位置をメモリの利用率 (%) で指定します。	80
wojms.orange.threshold	メモリリソースの Orange レベル (メモリが不十分である状態) のスタート位置をメモリの利用率 (%) で指定します。	90
wojms.red.threshold	メモリリソースの Red レベル (メモリが不足している状態) のスタート位置をメモリの利用率 (%) で指定します。	98
wojms.green.count	Green レベルの時にクライアントが遅延制御せずに送信できるメッセージの最大数を指定します。	5000
wojms.yellow.count	Yellow レベルの時にクライアントが遅延制御せずに送信できるメッセージの最大数を指定します。	500
wojms.orange.count	Orange レベルの時にクライアントが遅延制御せずに送信できるメッセージの最大数を指定します。	50
wojms.red.count	Red レベルの時にクライアントが遅延制御せずに送信できるメッセージの最大数を指定します。	0

## 自動生成送信先に関するプロパティ

プロパティ名	説明	既定値
wojms.autocreate.topic	Topic の自動生成を許可するかを指定します。 true: 許可する false: 許可しない	false
wojms.autocreate.queue	Queue の自動生成を許可するかを指定します。 true: 許可する。 false: 許可しない。	false
wojms.autocreate.topic.consumerFlowLimit	自動生成された Topic に接続された Consumer が一回の送信処理で受信できるメッセージの最大数を指定します。	1000
wojms.autocreate.queue.consumerFlowLimit	自動生成された Queue に接続された Consumer が一回の送信処理で受信できるメッセージの最大数を指定します。	100
wojms.autocreate.queue.maxNumActiveConsumers	自動生成された Queue でメッセージを受信するこの Consumer の最大数を指定します。-1 は無制限です。	-1
wojms.autocreate.queue.maxNumBackupConsumers	自動生成された Queue で待機状態の Consumer の数を指定します。-1 は無制限です。	0
wojms.autocreate.destination.maxNumMsgs	自動生成される送信先の最大メッセージ数を指定します。-1 は無制限です。	100000
wojms.autocreate.destination.maxTotalMsgBytes	自動生成される送信先の最大メッセージサイズ (バイト、K バイト、または M バイト単位) を指定します。-1 は無制限です。 128, 128b は 128 バイトを、128k は 128K バイト (128 × 1024=131072 バイト)、128m は 128M バイト (128 × 1024 × 1024=134217728 バイト) を表します。	10m
wojms.autocreate.destination.maxNumProducers	自動生成される送信先に接続可能なプロデューサ数を指定します。-1 は無制限です。	100
wojms.autocreate.destination.maxBytesPerMsg	自動生成される送信先の 1 メッセージ当たりの最大サイズ (バイト、K バイト、または M バイト単位) を指定します。-1 は無制限です。 128, 128b は 128 バイトを、128k は 128K バイト (128 × 1024=131072 バイト)、128m は 128M バイト (128 × 1024 × 1024=134217728 バイト) を表します。	10k
wojms.autocreate.destination.limitBehavior	自動生成される送信先に対して新しいメッセージが届いたときに、送信先のメモリ制限のしきい値に到達していた場合の JMS サーバの振る舞いを指定します。 FLOW_CONTROL : プロデューサとの間でフロー制御 (低速化) を行います。 REMOVE_OLDEST : 最古のメッセージを破棄します (プロデューサがメッセージ削除の通知を受け取ることはありません)。	REJECT_NEWEST

	<p>REMOVE_LOW_PRIORITY : メッセージの有効期限に従い優先度が最低のメッセージを破棄します(プロデューサがメッセージ削除の通知を受け取ることはありません)。</p> <p>REJECT_NEWEST : 最新のメッセージを拒否します(永続メッセージの場合はプロデューサにメッセージ拒否の例外が発生しますが、非永続メッセージの場合には発生しません)。</p>	
--	---	--

### クライアント監視に関するプロパティ

プロパティ名	説明	既定値
wojms.ping.enabled	<p>クライアントとの接続を監視するか否かを指定します。</p> <p>true: 行う</p> <p>false: 行わない</p>	true
wojms.ping.interval	<p>クライアントとの接続を監視する間隔を秒単位で指定します。</p>	120
wojms.client_metrics.enabled	<p>JMS クライアントのチューニング情報を採取するかどうかを指定します。</p> <p>true: 採取する</p> <p>false: 採取しない</p> <p>true(採取する)を指定すると、コネクション一覧表示(otxadmindlist-jms-connections)で各コネクションに対するメモリ情報を表示します。</p>	false

### JMS サーバクラスタに関するプロパティ

プロパティ名	説明	既定値
wojms.enable_cluster	<p>JMS サーバクラスタを利用するかどうかを指定します。</p>	false
wojms.cluster.brokerlist	<p>クラスタに属するすべての JMS サーバのホスト名とポート番号を指定します。[hostName][:port] の形式で、「,」で区切って指定します。クラスタ内の JMS サーバすべてで同じ値を指定する必要があります。</p>	
wojms.cluster.masterbroker	<p>マスターブローカのアドレスを指定します。[hostName][:port] の形式で指定します(省略可能)。クラスタ内の JMS サーバすべてで同じ値を指定する必要があります。</p>	

### JMSクライアントのプロパティ

JMS クライアントのプロパティの一覧を示します。これらのプロパティは、非 J2EE 環境下で動作する Java アプリケーションに有効であり、Java システムプロパティとして設定します。

ログの利用方法については、「8.3.11 JMSクライアントのログ出力」を、チューニング情報採取に関しては、「運用編」-「チューニング」-「JMSのチューニング」-「JMSクライアントのチューニング情報採取」を参照してください。



プロパティ名	説明	既定値
wojms.client.log.level	JMS クライアントのログを採取する際のログレベル (NONE、ERROR、WARNING、INFO、DEBUG、DEBUGMED、DEBUGHIGH のいずれか) を指定します。 NONE: 出力しない	NONE
wojms.client.log.file	JMS クライアントのログを採取する際のログファイル名を指定します。ログレベルのみが指定され、ログファイル名が指定されていない場合は、標準出力に出力します。	
wojms.client.metrics	メモリや、フロー制御などのチューニング情報を出力するかどうかを指定します。	false
wojms.client.metrics.interval	チューニング情報を出力する間隔(ミリ秒)を指定します。	5000
wojms.client.metrics.log	チューニング情報を出力するファイル名を指定します (JMS クライアントのログファイルとは別です)。ファイル名が指定されていない場合は、標準出力に出力します。	

## 8.3. その他の設定項目・設定方法

JMS サーバのその他の設定項目、および設定方法について説明します。

### 8.3.1. ファイルストアを利用するための設定

ファイルストアを利用するための設定について説明します。

インストール直後は、ファイルストアを利用する設定になっていますが、各プロパティにはデフォルト値が設定されていますので、ご利用になるシステムに合わせてプロパティの設定値を変更してください。ファイルストアをご利用になるためのプロパティの詳細については、「8.2.2JMSのプロパティ/属性一覧」-「JMSサーバのプロパティ」を参照してください。

※ ストアを変更(ファイルストア⇄JDBC ストア)する場合、それまでに永続化された情報(メッセージ、送信先、永続サブスクリプション、トランザクション)は、変更先のストアには引き継がれないことにご注意ください。

### 8.3.2. JDBCストアを利用するための設定

JDBC ストアを利用するための設定について説明します。

インストール直後は、ファイルストアを利用する設定になっています。JDBC ストアをご利用するためには、次に示す手順をおこなってください。

- ① JMSサーバのプロパティファイルに、JDBCストアを利用するためのプロパティを設定します。プロパティの詳細については、「8.2.2JMSのプロパティ/属性一覧」-「JMSサーバのプロパティ」を参照してください。
- ② 次のパスに JDBC ドライバの jar ファイルのコピー、またはシンボリックリンクを配置します。  
`#{ INSTANCE_ROOT }/lib/ext`
- ③ wojmsdbmgr コマンドの create all サブコマンドを実行して、永続データを格納するためのデータベーススキーマを作成します。wojmsdbmgr コマンドの詳細については、「運用コマンドリファレンスマニュアル」-「JMS」-「wojmsdbmgr」を参照してください。

**備考** : JMS サーバインスタンス識別子を指定しているときは、wojmsdbmgr コマンドで -b オプションにより、インスタンス識別子を指定する必要があります。wojmsdbmgr コマンドの詳細については、「運用コマンドリファレンスマニュアル」-「JMS」-「wojmsdbmgr」を参照してください。

データベーススキーマを作成するためのプロパティについて説明します。ここでは、Oracle DBMS を利用する際の例を示します。なお、Oracle DBMS を含め、PostgreSQL や、Microsoft SQL Server の設定は、JMS サーバの設定ファイル(config.properties)にコメントとして記述してありますので、ご利用の DBMS に合わせてコメントをはずし、必要な設定を行ってください。

プロパティ名	設定値
wojms.persist.store	jdbc
wojms.persist.jdbc.brokerid	例)jms1 省略可能
wojms.persist.jdbc.opendburl	jdbc:oracle:thin:@localhost<hostname>:<port>:<sid>
wojms.persist.jdbc.createdburl	jdbc:oracle:thin:@localhost<hostname>:<port>:<sid> 省略可能 ただし、wojmsdbmgr create all コマンドを実行するときは、必ず指定しておく必要があります。
wojms.persist.jdbc.closedburl	jdbc:oracle:thin:@localhost<hostname>:<port>:<sid> 省略可能
wojms.persist.jdbc.user	例)scott
wojms.persist.jdbc.password	例)tigger
wojms.persist.jdbc.driver	oracle.jdbc.driver.OracleDriver
wojms.persist.jdbc.table.WOJMSSV35	CREATE TABLE \${name} (STOREVERSION NUMBER(10) NOT NULL, BROKERID VARCHAR(100))
wojms.persist.jdbc.table.WOJMSSCREC35	CREATE TABLE \${name} (RECORDTIME NUMBER NOT NULL, RECORD LONG RAW)
wojms.persist.jdbc.table.WOJMSDEST35	CREATE TABLE \${name} (DID VARCHAR(100) NOT NULL, DEST LONG RAW, PRIMARY KEY(DID))
wojms.persist.jdbc.table.WOJMSINT35	CREATE TABLE \${name} (CUID NUMBER NOT NULL, INTEREST LONG RAW, PRIMARY KEY(CUID))
wojms.persist.jdbc.table.WOJMMSG35	CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, DID VARCHAR(100), MSGSIZE NUMBER, MSG LONG RAW, PRIMARY KEY(MID))
wojms.persist.jdbc.table.WOJMSPROPS35	CREATE TABLE \${name} (PROPNAME VARCHAR(100) NOT NULL, PROPVALUE LONG RAW, PRIMARY KEY(PROPNAME))
wojms.persist.jdbc.table.WOJMSILIST35	CREATE TABLE \${name} (MID VARCHAR(100) NOT NULL, CUID NUMBER, DID VARCHAR(100), STATE NUMBER(10), PRIMARY KEY(MID, CUID))
wojms.persist.jdbc.table.WOJMSTXN35	CREATE TABLE \${name} (TUID NUMBER NOT NULL, STATE NUMBER(10), TSTATEOBJ LONG RAW, PRIMARY KEY(TUID))
wojms.persist.jdbc.table.WOJMSTACK35	CREATE TABLE \${name} (TUID NUMBER NOT NULL, TXNACK LONG RAW)

※ ストアを変更(ファイルストア⇔JDBC ストア)する場合、それまでに永続化された情報(メッセージ、送信先、永続サブスクリプション、トランザクション)は、変更先のストアには引き継がれないことにご注意ください。

※ Oracle の OCI ドライバを利用する場合、「注意制限事項」-「JDBC データソース」-「Oracle ご利用時の注意・制限事項」を参

考にしてください。JMS も/etc/WebOTX/otx.conf の設定が有効になります。

### 8.3.3.ユーザ管理の設定

ここでは、JMS 認証を利用するためのユーザ管理について説明します。ドメイン認証を利用する場合は、ドメインによって管理されます。ドメインのユーザ管理については、「運用編」-「運用と操作」-「ユーザ管理」を参照願います。

認証に必要なユーザ管理は、wojmsusermgr コマンドで行います。wojmsusermgr コマンドの詳細については、「運用コマンドリファレンスマニュアル」-「JMS」-「wojmsusermgr」を参照してください。

ユーザ情報は、次の場所にあるファイルで管理されています。

`${INSTANCE_ROOT}/wojms/instances/wojmsbroker/etc/passwd`

#### グループ

グループ	説明
admin	JMS に対する管理者権限を有するグループです。
user	管理者権限を持たず、クライアントアプリケーションからのアクセスに必要な権限だけを有するグループです。 コネクション確立時にユーザ認証を行う場合に使用するユーザをこのグループに定義します。
anonymous	user 同等の権限を有するグループです。 コネクション確立時にユーザ認証を行わない場合に暗黙的に使用するユーザを定義します。 このグループに定義できるユーザは唯一です。

#### 状態

管理者は任意ユーザの登録情報の状態(アクティブ/非アクティブ)を変更することができます。

#### ユーザ名とパスワードの形式

次の文字は、ユーザ名として使用できません。

文字	説明
*	アスタリスク
,	カンマ
:	コロソ

改行や CR を文字として使用できません。

空白を含める場合には全体を引用符で囲んでください。

1 文字以上であることが必要ですが、基本的に長さに制限はありません。

#### デフォルトの設定

ユーザ名	パスワード	グループ	状態
admin	adminadmin	admin	アクティブ
guest	guest	anonymous	アクティブ

※ JMS サーバの起動時には admin 権限が必要です。JMS サーバを起動する運用管理エージェント側の定義には、あらかじめこのデフォルト値が設定されています。この設定を変更する場合には、管理エージェント側の設定も同時に変更してくだ

さい。

### 8.3.4. アクセス制御機能を利用するための設定

JMS クライアントからの操作（コネクション、送信先、送信先自動作成）に対するアクセス制御は、次の場所にあるアクセス制御プロパティファイルに定義します。

```
${INSTANCE_ROOT}/wojms/instances/wojmsbroker/etc/accesscontrol.properties
```

#### コネクションのアクセス制御

コネクションサービスへの権限を定義します。

```
connection.resourceVariant.access.principalType = principals
```

要素	説明
resourceVariant	コネクションのタイプを指定します。 NORMAL ADMIN
Access	アクセス制御を指定します。 allow deny
principalType	制御を受ける対象者を指定します。 user group
Principals	権限の制御を受ける対象を指定します。 principalType が user の場合は、ユーザ名（複数の場合はカンマで区切られたリスト）を指定します。 principalType が group の場合は、グループ名（複数の場合はカンマで区切られたリスト）を指定します。 両者とも、全てを対象とする場合には "*" を使用します。

デフォルトでは、通常のコネクションは全てのユーザに権限が、管理用のコネクションは admin グループにのみ権限が与えられています。

```
connection.NORMAL.allow.user=*
```

```
connection.ADMIN.allow.group=admin
```

たとえば、通常のコネクション確立について、ユーザ xxx を拒否するには、次の記述を追加します。

```
connection.NORMAL.deny.user=xxx
```

#### 送信先のアクセス制御

送信先に対する権限を定義します。

```
resourceType.resourceVariant.operation.access.principalType = principals
```

要素	説明
resourceType	送信先のタイプを指定します。 queue

	topic
resourceVariant	送信先の名前、もしくはすべての送信先を意味する"*"を指定します。
operation	クライアントのタイプを指定します。 produce consume browse
access	アクセス制御を指定します。 allow deny
principalType	制御を受ける対象者を指定します。 user group

デフォルトでは、すべてに権限が与えられています。

```
queue.*.produce.allow.user=*
```

```
queue.*.consume.allow.user=*
```

```
queue.*.browse.allow.user=*
```

```
topic.*.produce.allow.user=*
```

```
topic.*.consume.allow.user=*
```

たとえば、トピック xxx からのメッセージ受信について、グループ yyy を拒否するには、次の記述を追加します。

```
topic.xxx.consume.deny.group=yyy
```

## 送信先自動作成のアクセス制御

送信先の自動作成の権限を定義します。

```
resourceType.create.access.principalType = principals
```

要素	説明
resourceType	送信先のタイプを指定します。 queue topic
access	アクセス制御を指定します。 allow deny
principalType	制御を受ける対象者を指定します。 user group

デフォルトでは、すべてに権限が与えられています。

```
queue.create.allow.user=*
```

```
topic.create.allow.user=*
```

たとえば、topic の自動作成について、ユーザ xxx を拒否するには、次の記述を追加します。

```
topic.create.allow.user=*
```

### 8.3.5.SSL通信を利用するための設定

JMS サーバとクライアントアプリケーションとの間、JMS サーバと管理ツールとの間で送受信されるメッセージを SSL で暗号化するために必要な設定について説明します。

前者は ssljms コネクションサービス、後者は ssladmin コネクションサービスが対象です。

#### ① 自己署名型証明書の生成

wojmskeytool コマンドで JMS サーバの自己署名型証明書を生成してください。wojmskeytool コマンドの詳細は、「運用コマンドリファレンスマニュアル」-「JMS」-「wojmskeytool」を参照してください。

```
wojmskeytool -broker
```

証明書が格納されるキースタアの位置は次の場所となります。

```
$WOJMS_HOME/etc/keystore
```

#### ② SSL コネクションサービスの有効化

config.properties ファイルの次のプロパティ設定を変更してください。

```
wojms.service.activelist=jms,admin,ssljms,ssladmin
```

※jms、admin の両方を SSL 化する場合

実際に有効になったかどうかは wojmscmd コマンドで確認することができます。

#### ③ JMS サーバの再起動

次のいずれかの方法でキースタアのパスワードを指定して JMS サーバを再起動してください。

起動引数で指定する方法

パスワードファイルを使用する方法

詳細は、「8.3.6パスワードファイルの使用」を参照してください。

#### ④ JMS クライアントの設定

クライアントの設定は、Java アプリケーションの場合と、J2EE 環境下で動作する J2EE アプリケーションの場合で方法が異なります。

Java アプリケーションの場合

コネクションファクトリに次の属性値を設定してください。

```
wojmsConnectionType=TLS
```

この設定を有効にするには、次の 2 つの方法があります。

##### ① JNDI に登録するオブジェクトに設定する方法

コネクションファクトリリソースの場合は、MO 属性「コネクションタイプ (wojmsConnectionType)」で指定します。

##### ② Java クライアントアプリケーション起動時に Java システムプロパティで指定する方法

通常は JNDI に登録されたオブジェクトにあらかじめ設定された属性値に従って暗黙的に決まりますが、これを Java システムプロパティとして与えることによって、起動時に自由に選択することができます。

J2EE アプリケーションの場合

コネクションファクトリリソースの**プロパティ**として、

```
MessageServiceAddressList=<wojmsBrokerHostName>:<wojmsBrokerHostPort>/ssljms
```

を設定してください。<wojmsBrokerHostName>は「JMS サーバのホスト名」の値を、<wojmsBrokerHostPort>は「JMS サーバのホストのポート番号」の値を指定します。

**補足**：コネクションファクトリリソースに対するプロパティ設定

統合運用管理ツールの場合

「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「リソース」-「JMS リソース」-「コネクションファクトリリソース」-「<JNDI 名

>J-【プロパティの設定】操作で設定します。

統合運用管理ツールの場合

次のように、set コマンドを利用して設定します。

```
set server.resources.jms-resource.jms-connection-factory.<JNDI 名>.property.MessageServiceAddressList=  
<wojmsBrokerHostName>:<wojmsBrokerHostPort>/ssljms
```

例)

```
set server.resources.jms-resource.jms-connection-factory.jms/MyCF.property.MessageServiceAddressList=  
localhost:9700/ssljms
```

### 8.3.6.パスワードファイルの使用

ここでは、JMS サーバが使用する SSL、JDBC のパスワードを起動引数ではなく、パスワードファイルを使用して指定する方法について説明します。

#### ① パスワードファイルの作成

パスワードファイルの作成は、次の位置にある雛形を利用してください。

```
${AS_INSTALL}/wojms/etc/passfile.sample
```

パスワードファイルには次のパスワードを指定することができます。

パスワード	説明
wojms.keystore.password	SSL 運用時に必要なキーストアのパスワードを指定します。
wojms.persist.jdbc.password	データベースとの接続時に必要なパスワードを指定します。

#### ② パスワードファイルの有効化

作成したパスワードファイルを有効にするためには、次のプロパティを指定する必要があります。

```
wojms.passfile.enabled=true
```

#### ③ パスワードファイルの指定

JMS サーバの起動時に使用するパスワードファイルを指定します。

起動引数で指定する方法

次のオプションを JMS サーバの起動引数に指定してください。

```
-passfile <password file>
```

運用管理コマンドで指定する場合は、JMS サービスの「start-args」属性に、統合運用管理ツールで指定する場合は、JMS サービスの MO 属性「起動引数」に上記を設定してください。

config.properties に設定する方法

次のプロパティを設定してください。

```
wojms.passfile.dirpath=<passfile directory>
```

```
wojms.passfile.name=<passfile name>
```

**注意：** 起動引数を指定した場合は、config.properties に設定した内容は無視されます。

### 8.3.7.Standard/Enterprise Editionで動作させる場合の設定

Standard/Enterprise Edition において、TP モニタ上で MDB や JMS の非同期受信を行う CORBA アプリケーションを起動する場合の設定について説明します。

#### MDBでTopicのマルチコンシューマ負荷分散を利用する場合

MDB で Topic のマルチコンシューマ負荷分散を利用するには、クライアント識別子とその共有フラグを以下に示すいずれかの方

法で設定する必要があります。

#### ① 配備記述子での設定

配備ツールにおいて、[メッセージドリブン]タブの[メッセージングサービス]を「非 JMS」にし、[アクティベーション設定プロパティ]に以下のプロパティを指定してください。

プロパティ	設定値
clientId	クライアント識別子
enableSharedClientID	true

#### ② プロセスグループでの設定

プロセスグループのシステムプロパティに以下のプロパティを指定してください。

同じプロセスグループの中で複数の MDB が配備されるような場合、クライアント識別子の共有範囲が複数の MDB 間にまたがることに注意してください。

プロパティ	設定値
wojmsConfiguredClientID	クライアント識別子
wojmsEnableSharedClientID	true

### CORBAアプリケーションでTopicのマルチコンシューマ負荷分散を利用する場合

CORBA アプリケーションで Topic のマルチコンシューマ負荷分散を利用する場合は、自動的にクライアント識別子とその共有フラグを設定します。この場合のクライアント識別子は、[アプリケーショングループ名]\_[プロセスグループ名]です。

以下の方法でクライアント識別子を明示的に指定することも可能です。

#### ① プログラムでの設定

javax.jms.TopicConnection の setClientID メソッドで指定してください。

#### ② プロセスグループでの設定

プロセスグループの Java システムプロパティに以下のプロパティを指定してください。

同じプロセスグループの中で複数のコンシューマを生成するような場合、クライアント識別子の共有範囲が複数のコンシューマ間にまたがることに注意してください。

プロパティ	説明
wojmsConfiguredClientID	クライアント識別子

#### ③ コネクションファクトリリソースでの設定

使用するコネクションファクトリリソースの以下の属性を指定してください。

同じコネクションファクトリリソースを複数のコンシューマで利用すると、クライアント識別子の共有範囲が複数のコンシューマ間にまたがることに注意してください。

server.resources.jms-resource.jms-connection-factory.connection-factory-name :

プロパティ	説明
wojmsConfiguredClientID	クライアント識別子

### CORBAアプリケーションでWebOTXアプリケーション管理機能を利用する場合

JMS の非同期受信を行う CORBA アプリケーションで次のような WebOTX のアプリケーション管理機能を利用するための設定に



ついて説明します。

#### ① プロセスの障害監視・閉塞機能

MessageListener の呼び出しが指定した時間経過しても応答しない場合や、プロセスが異常終了した場合に、該当するプロセスを閉塞し、指定にしたがってプロセスを再起動します。

#### ② 性能情報の採取機能

MessageListener 呼び出しに要した時間や、呼び出し回数といった性能および統計情報を採取します。

アプリケーション管理機能を利用するためには、プロセスグループに次の Java システムプロパティを設定してください。

プロパティ名	説明
wojmsEnableReliableThread	アプリケーション管理機能を利用するかどうかを指定します。 true アプリケーション管理機能を有効にします。 false アプリケーション管理機能を無効にします(既定値)。
wojmsReliableRetryAttempts	アプリケーション管理機能を有効にした場合の、管理機能に対する初期処理が失敗した場合のリトライ回数を指定します。 既定値は、200 回です。
wojmsReliableRetryInterval	アプリケーション管理機能を有効にした場合の、管理機能に対する初期処理が失敗した場合のリトライ間隔を、秒単位で指定します。 既定値は、3(秒)です。

### 8.3.8.ユーザ管理方式の設定

ここでは、JMS サーバに接続するユーザの管理方式を設定する方法について説明します。

- ① JMSサーバのプロパティファイルに、認証方式を指定するためのプロパティを設定します。プロパティの詳細については、「8.2.2JMSのプロパティ/属性一覧」-「JMSサーバのプロパティ」を参照してください。

ユーザ管理方式を指定するためのプロパティについて説明します。ここでは JMS 認証を利用する際の例を示します。

#### JMS 認証を使用する場合

プロパティ名	設定値
wojms.authentication.provider	WOJMS
wojms.authentication.type	digest
wojms.authentication.digest.user_repository	file

### 8.3.9.メッセージ再配信に関する設定

ここでは、ロールバックされたメッセージの再配信に関する設定について説明します。

再配信の遅延時間と回数は、JMS サーバのプロパティか、JMS サービスの MO の属性で指定します。ここで指定した値は、JMS

サーバで動作するコンシューマ全体に有効になります。

server.jms-service :

MOの属性	JMSサーバのプロパティ名	説明
wojmsRedeliveryDelay	wojms.redelivery.delay	再配信遅延時間(秒)
wojmsRedeliveryLimit	wojms.redelivery.limit	再配信回数の上限

各プロパティのデフォルト値などの詳細については、「8.2.2JMSのプロパティ/属性一覧」を参照してください。

なお、再配信の遅延時間は、コネクションファクトリ単位や、コンシューマプログラム向けに用意しているAPI(WebOTX V5よりサポート)で、セッションやメッセージ単位に指定することも可能です。

コネクションファクトリ単位に指定する場合は、以下のプロパティを指定してください。

server.resources.jms-resource.jms-connection-factory.connection-factory-name :

プロパティ	説明
wojmsRedeliveryDelay	再配信遅延時間(秒)

セッションや、メッセージ単位で指定する場合は、「APIリファレンスマニュアル」-「JMS」-「拡張インタフェース」を参照してください。

## MDBで再配信遅延時間を指定する場合

MDBに対して再配信遅延時間を指定するには、以下に示す方法があります。優先順位は、高いものから順に、①、②、③となっており、以下のいずれも指定されていない場合は、前述のJMSサーバの指定(wojms.redelivery.delay)に従って再配信されます。

### ① 配備記述子での設定

配備ツールにおいて、[メッセージドリブン]タブの[メッセージングサービス]を「非JMS」にし、[アクティベーション設定プロパティ]に以下のプロパティを指定してください。

プロパティ	説明
RedeliveryDelay	再配信遅延時間(秒)

### ② プロセスグループでの設定

プロセスグループのシステムプロパティに以下のプロパティを指定してください。

プロパティ	設定値
wojmsRedeliveryDelay	再配信遅延時間(秒)

### ③ コネクションファクトリリソースでの設定

使用するコネクションファクトリリソースの以下の属性を指定してください。

server.resources.jms-resource.jms-connection-factory.connection-factory-name :

属性名	設定値
wojmsRedeliveryDelay	再配信遅延時間(秒)

### 8.3.10.破棄メッセージの転送に関する設定

ここでは、破棄メッセージの転送に関する設定について説明します。「破棄メッセージ」とは、再配信回数を超過したメッセージや、有効期限切れのメッセージのことをさします。

破棄メッセージの転送先は、JMS サーバのプロパティか、JMS サービスの MO の属性で指定します。ここで指定した値は、JMS サーバ全体で有効になります。

server.jms-service :

MO の属性	JMS サーバのプロパティ名	説明
wojmsRedeliveryDestination	wojms.redelivery. destination	再配信回数を超過したメッセージを転送する送信先名
wojmsExpirationDestination	wojms.expiration. destination	有効期限切れメッセージを転送する送信先名

#### 注意 :

- ここで指定する送信先は、あらかじめ作成しておく必要があります。送信先の自動生成許可の設定 (wojms.autocreate.topic, wojms.autocreate.queue) にかかわらず、送信先が自動的に作成されることはありません。
- 指定した送信先が存在しない場合、メッセージは破棄されます。
- 「メッセージの最大数」、「メッセージの最大合計サイズ」を設定している場合、基本的には「制限到達時の振る舞い」の設定に準じた動作となりますが、FLOW\_CONTROL の指定は無視され、そのままメッセージを移動します。
- 「1メッセージ当たりの最大サイズ」の指定を超過したメッセージであっても移動します。
- 移動先の送信先が停止状態 (JMS 固有コマンドによる制御) の場合でもメッセージを移動します。
- 移動中に DB への保存に失敗した場合は、差し戻されて再配信の対象となります。
- 元の送信先と移動先が同じ場合、メッセージを削除します。

移動したメッセージのメッセージヘッダと、メッセージプロパティの扱いについて、以下に記述します。

#### JMS のメッセージヘッダ:

フィールド名	値
JMSCorrelationID	変更しません。
JMSDeliveryMode	PERSISTENT に変更します。
JMSDestination	転送先の送信先に変更します。 元の情報は、WOJMSOriginalDestinationName に送信先名を含めて設定されます。
JMSExpiration	クリアします。
JMSMessageID	新規のメッセージ識別子に変更します。 元の情報は、メッセージプロパティ WOJMSOriginalMessageID に設定されます。
JMSPriority	変更しません。
JMSRedelivered	変更しません。
JMSReplyTo	変更しません。
JMSTimestamp	移動した時刻に変更します。

JMSType	変更しません。
---------	---------

### JMS のメッセージプロパティ:

プロパティ名	値
WOJMSDelayTime	削除します。
WOJMSOriginalMessageID	更新前の JMSMessageID を格納します。
WOJMSOriginalDestinationName	更新前の JMSDestination から取得した送信先名と送信先タイプを連結した文字列を格納します。
WOJMSDeletedReason	メッセージの削除理由を文字列で格納します。 再配信回数を超過したメッセージの設定値: DELETED 有効期限切れメッセージの設定値: EXPIRED

### 8.3.11.JMSクライアントのログ出力

非 J2EE 環境下で動作する Java アプリケーションでは、実行時に Java システムプロパティを設定することにより、ログ出力が可能です。ログは、障害解析などに利用します。

ログを出力するには、以下を Java システムプロパティとして指定します。

```
wojms.client.log.level
wojms.client.log.file
```

各プロパティの詳細については、「8.2.2JMSのプロパティ/属性一覧」-「JMSクライアントのプロパティ」を参照してください。

### 8.3.12.分散トランザクションへの参加

J2EE 環境で、JMS を分散トランザクションに参加させる場合、トランザクションサービスの JCA リソースを登録する必要があります。

JCA リソースの登録方法については、「運用編」-「運用と操作」-「Transaction サービスの運用操作」-「リソースの登録・削除」を参照してください。

#### 注意 :

- 非 J2EE 環境下では JMS を分散トランザクションに参加させることはできません。
- JCA リソースは接続先が異なる JMS サーバごとに 1 つ登録してください。

例えば、以下のように、接続先が同一のコネクションファクトリを複数作成した場合、

```
jms/QueueConnectionFactory { wojmsBrokerHostName=host1, wojmsBrokerHostPort=9700 }
jms/QueueConnectionFactory2 { wojmsBrokerHostName=host1, wojmsBrokerHostPort=9700 }
jms/TopicConnectionFactory { wojmsBrokerHostName=host1, wojmsBrokerHostPort=9700 }
```

JCA リソースとして登録(create-ts-jca-resource)するのは、

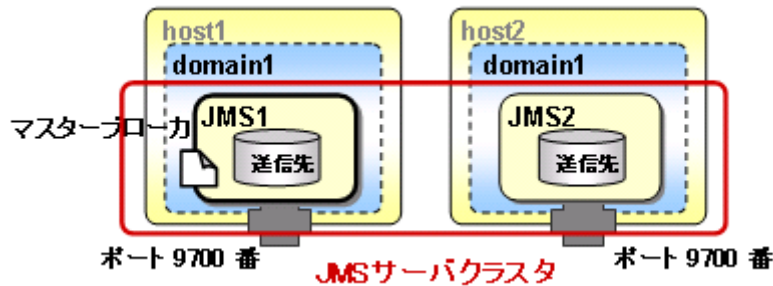
```
jms/QueueConnectionFactory
(あるいは、jms/QueueConnectionFactory2 か、jms/TopicConnectionFactory のいずれでも構いません)
```

の 1 つだけになります。

### 8.3.13.JMSサーバクラスタを利用するための設定

ここでは、2つの JMS サーバで構成される JMS サーバクラスタの設定方法について説明します。

設定は、以下のように、host1 の domain1 で起動する JMS サーバ(JMS1)と、host2 の domain1 で起動する JMS サーバ(JMS2)で JMS サーバクラスタを構成するものとします。また、マスターブローカとして、host1-domain1 上の JMS サーバを指定します。



#### ① host1-domain1、host2-domain1 の JMS サーバの設定

統合運用管理ツールの場合

「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「JMS サービス」の「クラスタ」タブの各属性に対して、以下を設定します。

JMS サーバクラスタの使用	true(チェックする)
クラスタに属する JMS サーバのアドレスリスト	host1:9700,host2:9700
マスターブローカのアドレス	host1:9700

運用管理コマンドの場合

set コマンドで以下を設定します。

```
otxadmin> set server.jms-service.enableCluster=true
otxadmin> set server.jms-service.clusterBrokerList=host1:9700,host2:9700
otxadmin> set server.jms-service.clusterMasterBroker=host1:9700
```

#### 注意：

同一ホストで、JMS サーバのクラスタを構成する場合、上記の設定に加えて、JMS サーバインスタンス識別子をホスト内で一意となるように設定しておく必要があります。

統合運用管理ツールの場合

「WebOTX 管理ドメイン[<ホスト名>]」-「<ドメイン名>」-「アプリケーションサーバ」-「JMS サービス」-【プロパティの設定】で、以下を設定します。

```
instance-name=<JMS サーバインスタンス識別子>
```

運用管理コマンドの場合

set コマンドで以下を設定します。

```
otxadmin> set server.jms-service.property.instance-name=<JMS サーバインスタンス識別子>
```

#### ② JMS サーバの再起動

クラスタの設定は、JMS サーバの再起動後に有効になるため、すべての JMS サーバを再起動します。

#### 注意：

- マスターブローカを設定した場合は、マスターブローカが起動するまで、他の JMS サーバは利用できません。
- マスターブローカを設定した場合は、マスターブローカが停止中、クラスタ内の JMS サーバでは以下の操作ができなくなります。それ以外の操作は正常に行えます。
  - 物理的な送信先の生成/削除/プロパティ更新
  - 永続サブスクリプションの削除

### 8.3.14.接続先分散のための設定

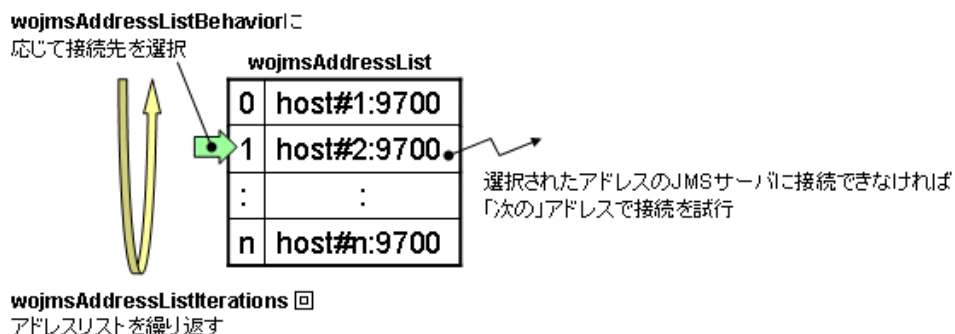
JMS サーバクラスタなどで、接続先の分散を行うためにアドレスリストを使用することができます。アドレスリストは、コネクションファクトリリソースの属性として設定します。

server.resources.jms-resource.jms-connection-factory.connection-factory-name :

属性名	説明	既定値
wojmsAddressList	<p>接続先のリストを指定します。</p> <p>この値を設定した場合は、「JMS サーバのホスト名」(wojmsBrokerHostName)と、「JMS サーバのホストのポート番号」(wojmsBrokerHostPort)の設定は無視されますので、必要に応じてこの接続先をリストに含めてください。</p> <p>アドレスの構文は、[hostName][:port]で、複数のアドレスを指定する場合は、「,」で区切ります。</p>	
wojmsAddressListBehavior	<p>アドレスリストから、接続先を選択する方法を指定します。</p> <p>PRIORITY: アドレスリストに指定された順番</p> <p>RANDOM: ランダムに選択</p> <p>接続障害後の再接続は、いずれの場合も、接続がエラーになった JMS サーバに対するアドレスから始まります。</p>	PRIORITY
wojmsAddressListIterations	<p>接続の試行を中止して例外を通知するまでの、アドレスリストの繰り返し回数を指定します。</p>	1

次に、アドレスリストを設定した場合の動作について説明します。

JMS クライアントランタイムは、wojmsAddressListBehavior の設定に応じて、wojmsAddressList から接続先を選択し接続を行います。最初に選択したアドレスで接続できなかった場合は、wojmsAddressList から「次の」アドレスを選択して接続を試みます。それでも接続できなければ「次の」アドレス、と進めていき、wojmsAddressListIterations 回アドレスリストを繰り返して接続できなければ、例外を発生させるようになっています。



### 8.3.15.再配信メッセージの順序保証のための設定

アプリケーション異常など、トランザクションのロールバックによりメッセージが再配信される場合でもメッセージの配信順序を保証

したい場合は、物理的な送信先に対して次の設定を行います。

server.jms-service.jms-physical-destination.physical-destination-name :

属性名	説明	既定値
supportOrderedRedelivery	メッセージ再配信時の順序保証を行うかどうかを指定します。 true:再配信時の順序保証を行います。接続できるコンシューマ数は1になります。 false:再配信時の順序保証を行いません。	false (再配信時の順序保証はしない)

**注意 :**

順序保証の設定を変更する場合は、送信先のコンシューマ数が 1 以下の状態で行ってください。複数のコンシューマが接続しているときに順序保証の設定を有効にしても、設定は無視されます。

### 再配信メッセージの順序保証の利用条件

再配信メッセージの順序保証が利用できるのは、次の環境に制限されます。

- 送信先タイプはキューであること  
JMS サーバクラス構成では、送信先の「ローカル配信のみ (isLocalOnly)」の設定を有効 (true) にして、ローカルコンシューマ (送信先が作成されたブローカに接続しているコンシューマ) だけにメッセージを配信するようおく必要があります。
- コンシューマのメッセージ受信がトランザクション内で行われていること
- 送信先に接続するコンシューマ数は 1 であること  
送信先に接続できるコンシューマ数は、1 に制限されます。また、MDB の場合は、多重度を 1 に制限しておく必要があります。

### 再配信メッセージの順序保証の利用時の動作

再配信メッセージの順序保証を利用する場合、各設定が行われたメッセージは次のようになります。

- 優先度を設定したメッセージ  
1 件目は優先度の最も高いものがコンシューマに配信されます。コンシューマがそのメッセージをロールバックした場合は、優先度を無視して そのメッセージが再配信されます。2 件目は、その次に優先度の高いものが配信されます。
- 配信時間を設定したメッセージ  
1 件目は、最初に指定時刻に達したメッセージがコンシューマに配信されます。コンシューマがそのメッセージをロールバックすると、配信時間を無視して そのメッセージが再配信されます。コンシューマがメッセージをロールバックしている間は、他のメッセージが 指定時刻になっても配信されません。2 件目は、その次に指定時刻に達したものが配信されます。
- 有効期限を設定したメッセージ  
メッセージの再配信中に有効期限が切れた場合は、そのメッセージが破棄(または、転送)され、コンシューマには、その次のメッセージが配信されます。
- 再配信回数、再配信遅延時間、破棄メッセージ転送先の設定  
いずれの設定も有効になります。コンシューマが 1 件目のメッセージを受信後、ロールバックすると、遅延時間に指定した時間だけ待ってそのメッセージが再配信され、再配信回数の上限を超えると、そのメッセージは破棄(または、転送)されます。その後、キュー内の 2 件目のメッセージが配信されます。

# 9. JDBCデータソースに関する設定

JDBC データソースに関する設定について説明します。

## 9.1. JDBCデータソース設定項目・設定方法

JDBC データソースの MO 名は、「jdbc-datasource.datasource-name」です。datasource-name は、「jdbc/」で始まる JDBC データソースの定義名です。JNDI サーバへの登録名を兼ねます。JDBC データソースの登録を行うと、JDBC データソースの定義情報が domain.xml に登録され、同時に、JNDI サーバへの登録が行われます。また、管理コマンド等で設定変更を行った際には、domain.xml の変更と JNDI サーバへの再登録が行われます。domain.xml の設定をエディタ等で修正しても JNDI サーバへの再登録が行われないため、変更内容を動作に反映させることはできません。

コマンドで各項目を参照・設定する際は、「server.resources.jdbc-datasource.datasource-name」という名前を指定します。また、統合運用管理ツールでは、ドメインツリーの「WebOTXAdmin[ホスト名].ドメイン名.resources.jdbc-datasource.datasource-name」を選択します。項目の設定方法については「2各サービスの属性の設定」を参照してください。

JDBC データソースのプロパティの値を変更した場合、アプリケーションで、JNDI サーバから JDBC データソースを再取得した際に、変更内容が動作に反映されます。ただし、java:comp/env 指定で、アプリケーション固有の名前空間から JDBC データソースを取得する場合に、変更内容を動作に反映するためには、プロセス(JavaVM)の再起動が必要です。コネクションに関連するプロパティ(dataSourceType や jdbcMajorVersion、dataSourceName、serverName、networkProtocol、portNumber、userName、password、loginTimeout、jdbcDriverName、cursorMode、characterEncoding、maxStatements、useJTA、useOneConnectionPerTransaction、useDatabaseCluster、checkGarbageOption)を変更する場合は、プロパティ置換までの待機時間(propertyCycle)が経過した後で、そのプールに存在するコネクション全てが消滅し、新しいコネクションが作成されます。

なお、JDBC データソースでは、旧バージョンで提供していた dsadm コマンドも利用することができます。dsadm コマンドのリファレンスについては「JDBC データソースコマンドリファレンス」を参照ください。

### 定義の追加

> dsadm bind property-file-name

### 項目の参照

> dsadm list

### 項目の設定

> dsadm rebind property-file-name

## 9.2. JDBCデータソース設定項目一覧

JDBC データソースの MO を介して、次の項目を参照・設定することができます。

Dottedname : server.resources.jdbc-datasource.datasource-name

属性名 [attribute-name]	説明	既定値
JNDI サーバへの登録名 [jndiName]	JNDI サーバへの登録名です。 "jdbc/"で始まる名前を設定してください。この名前は JDBC データソースの定義を一意に表す名前として使用されます。	なし
データソースの種別 [dataSourceType]	JDBC データソースで使用する、JDBC ドライバベンダ提供のインタフェースの種別を表わす文字列です。次のいずれかを設定します。 <b>JDBC_EX_Oracle</b>	なし



	<p>バージョン 8.1.7 以降の Oracle の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBCCEX_Sybase</b></p> <p>バージョン 12.5 以降の Sybase Adaptive Server Enterprise の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBCCEX_MSSqlServer</b></p> <p>バージョン 2000 以降の Microsoft SQL Server の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBCCEX_DB2</b></p> <p>バージョン 8.1.4 以降の DB2 Universal Database の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBCCEX_Cloudscape</b></p> <p>バージョン 3.0.3 以降の Cloudscape の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBCCEX_SequeLink</b></p> <p>バージョン 5.0 以降の SequeLink の JDBC Optional Package インタフェースを使用します。Linux 版の Oracle8.1.7 を使用し、JTA と連携して 2 フェーズコミットを行う場合に設定してください。</p> <p><b>JDBCCEX_PostgreSQL</b></p> <p>バージョン 7.3.2 以降の PostgreSQL の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBCCEX_Derby</b></p> <p>バージョン 10.2.2 以降の Apache Derby の JDBC Optional Package インタフェースを使用します。</p> <p><b>JDBC</b></p> <p>java.sql.DriverManager を介して JDBC のインタフェースを使用します。JDBCCEX タイプでサポートされていない JDBC ドライバを使用する場合に使用してください。</p>	
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンです。 2 か 3 か 4 を設定します。	2
JDBC 仕様のマイナーバージョン [jdbcMinorVersion]	JDBC 仕様のマイナーバージョンです。 現在設定する必要はありません。	0
最小プールサイズ [minPoolSize]	JDBC コネクションのプールサイズです。 設定された数までの JDBC コネクションが常時保持されます。0 を設定した場合、プール管理を行いません。この値を越えて払い出されたコネクションは、shrinkDelayTime で指定された時間が経過した後で、コネクションをクローズした時、または、トランザクションが完了した時に解放されます。	4
最大プールサイズ [maxPoolSize]	JDBC コネクションの最大プールサイズです。	0

	maxPoolSize を超える JDBC コネクションの取得要求は失敗します。0 を設定した場合、制限なしとなります。それ以外の場合は、minPoolSize 以上の値を設定してください。	
初期プールサイズ [initialPoolSize]	プール作成時にプールされる JDBC コネクションの数です。  プール作成時とは、通常、ドメインやプロセスグループの起動時です。ただし、スタンドアロンの Java アプリケーションなどから JDBC データソースを利用する場合には、最初に getConnection メソッドを呼び出した時になります。  minPoolSize 以下の値を設定してください。	0
JTA 連携有無 [useJTA]	JTA (EJB のトランザクション) との連携有無を表わすブール値です。  <b>true</b> : JTA と連携します。  <b>false</b> : JTA と連携しません。	true
JDBC データソースについての説明 [description]	JDBC データソースについての説明です。  文字列の内容は自由です。例えば、プロパティの設定内容を判別するための記述を設定してください。	なし
JDBC URL またはデータベース名、 データソース名 [dataSourceName]	データベースサーバと接続するための文字列です。  dataSourceType の値に応じて、データベースとの接続文字列やデータベース名、またはデータソース名を設定します。	なし
ユーザ名 [userName]	データベースとの接続時に使用するユーザ名です。  アプリケーションへの JDBC コネクションの返却や、JDBC データソースのテスト操作、および、データベースサーバの状態監視 (checkServerOption への monitor 指定) が行われる場合に本設定が使用されます。ただし、JDBC コネクションの取得メソッドでユーザ名を指定した場合、本設定は使用されません。	なし
パスワード [password]	データベースとの接続時に使用するパスワードです。  アプリケーションへの JDBC コネクションの返却や、JDBC データソースのテスト操作、および、データベースサーバの状態監視 (checkServerOption への monitor 指定) が行われる場合に本設定が使用されます。ただし、JDBC コネクションの取得メソッドでパスワードを指定した場合、本設定は使用されません。	なし
サーバ名 [serverName]	データベースサーバ側のサーバ名 (TCP/IP ホスト名) です。	なし
ポート番号 [portNumber]	データベースサーバ側のポート番号です。  dataSourceType プロパティの値に応じて初期値が設定されます。統合運用管理ツールで JDBC データソースの登録を行う場合、データソースの種別 [dataSourceType] を変更する度に、デフォルト値が設定されることに注意してください。	dataSourceType  毎の設定を参照してください
ステートメントの最大プール数 [maxStatements]	ステートメントの最大プール数です。  0 を設定した場合、プール管理は行われません。設定可否は dataSourceType 毎に異なります。	0
ログインタイムアウト [loginTimeout]	JDBC コネクション接続時のタイムアウト値 (単位: 秒) です。  0 を設定した場合、タイマ監視は行われません。	0

無通信時間の上限 [maxIdleTime]	JDBC コネクションが未使用状態のままプールされ続ける時間(単位: 秒)です。 設定された時間が経過した未使用状態の JDBC コネクションは破棄されます。0 を設定した場合、アイドル時間監視は行われません。	0
プロパティ置換までの待機時間 [propertyCycle]	動作中の JDBC データソースが使用するプロパティ値を強制的に変更するまでの待ち時間(単位: 秒)です。 JNDI サーバから JDBC データソースを lookup し直した場合、設定された時間が経過した後で新しいプロパティが反映されます。0 を設定した場合、待ち合わせは行われません。	0
接続リトライ回数 [connectRetryMax]	JDBC コネクションの取得に失敗した場合の、接続リトライ回数です。 0 を設定した場合、リトライは行われません。	0
接続リトライ間隔 [connectRetryInterval]	JDBC コネクションの取得に失敗した場合の、接続リトライ間隔(単位: 秒)です。	10
データベースサーバの状態監視オプション [checkServerOption]	データベースサーバの状態監視オプションです。 JDBC データソースでは、データベースサーバの状態監視コマンドを発行することでデータベースサーバの状態を確認し、無効となった JDBC コネクションを取り除きます。 次のいずれかを設定します。 <b>monitor</b> : 定期的にデータベースサーバの状態を確認します。 <b>method</b> : JDBC コネクションを払い出す度に、データベースサーバの状態を確認します。 <b>none</b> : この機能を無効にします。	none
データベースサーバの状態監視間隔 [checkServerInterval]	checkServerOption に <b>monitor</b> を指定した時の監視間隔(単位: 秒)です。	180
データベースサーバの状態監視コマンド [checkServerCommand]	データベースの状態監視のために使用する SQL 命令です。 SELECT 命令など、ロックの競合が発生しない SQL 命令を指定してください。	commit
コネクションのガベージコレクション機能における動作オプション [checkGarbageOption]	JDBC Connection オブジェクトの、ガベージコレクション機能における動作オプションです。次のいずれかを設定します。 <b>close</b> : ガベージコレクションの対象となった JDBC コネクションを切断します。 <b>rollback</b> : ガベージコレクションの対象となった JDBC コネクションを切断します。 また、autoCommit モードに <b>false</b> が指定された状態でローカルトランザクションを実行していた場合は、ロールバックを発行します。 <b>debug</b> : <b>rollback</b> が設定された時と同じ動作を行います。また、ガベージコレクションの対象となった JDBC コネクションがアプリケーションに払い出された際のスタックトレースをログに出力します。 <b>none</b> : この機能を無効にします。  autoCommit モードに <b>true</b> (デフォルト値)が設定された状態でローカルトランザクションを実行する場合や、JTA(EJB)のトランザクションを実行する場合、	none

	<p><code>rollback</code> を指定する必要はありません。<code>close</code> を指定してください。</p> <p><b>その他の場合は、トランザクションが中途半端な状態でコミットされないようにするため、<code>rollback</code> を指定してください。</b></p>	
<p>コネクションのガベージコレクト間隔</p> <p>[<code>checkGarbageInterval</code>]</p>	<p><code>java.lang.System.gc()</code> を呼び出す間隔(単位:秒)です。</p> <p><code>gc()</code> を呼び出すことで、何処からも参照されていない <code>Connection</code> オブジェクトがガベージコレクトの対象となるように促します。0 が設定された場合、または、<code>checkGarbageOption</code> の値が <code>rollback</code> または <code>debug</code> でない場合は、<code>gc()</code> を呼び出しません。</p> <p><code>checkGarbageOption</code> プロパティで設定された動作の実行契機は、ガベージコレクタの実行契機次第ですので、即時性を保障できません。このため、<code>commit</code> や <code>rollback</code> の発行が漏れた際に、そのまま次のトランザクションが実行されてデータベース側のロック解除待ちとなり、長時間応答が返らない可能性があります。そうした障害を回避するためには、このパラメータを指定してください。ただし、<code>System.gc()</code> の呼び出しには、メモリの使用状況に応じてかなりの時間を要するため、少なくとも分単位の値を指定した上で、十分な評価を行ってください。</p>	0
<p>自動コミットのデフォルト値</p> <p>[<code>defaultAutoCommit</code>]</p>	<p>JTA(EJB)のトランザクションを実行しない場合の、<code>autoCommit</code> モードのデフォルト値です。</p> <p>JTA(EJB)のトランザクションを実行する場合、<code>autoCommit</code> モードは必ず自動的に <code>false</code> になります。</p>	true
<p>JTA のトランザクション毎に1つの物理コネクションを使用するかどうか</p> <p>[<code>useOneConnectionPerTransaction</code>]</p>	<p>JTA(EJB)のトランザクション毎に 1 つの物理コネクションを使用するかどうかを表わすブール値です。</p> <p>設定可否は <code>dataSourceType</code> 毎に異なります。JTA のトランザクションで、同一データベースに対する複数の JDBC コネクションを同時に利用する場合、<code>true</code> を設定してください。</p>	false
<p>データベースクラスタの使用有無</p> <p>[<code>useDatabaseCluster</code>]</p>	<p>データベースのクラスタ構成サポート機能の使用有無を表わすブール値です。</p> <p>設定可否は <code>dataSourceType</code> 毎に異なります。</p>	false
<p>コネクションの一括破棄可否</p> <p>[<code>resetAllConnectionsOnFailure</code>]</p>	<p>データベースの状態監視や JDBC ドライバからのコネクション障害通知で障害が検出された際に、プール内の全ての JDBC コネクションを破棄するかどうかを表わすブール値です。</p>	true
<p>コネクション解放までの待機時間</p> <p>[<code>shrinkDelayTime</code>]</p>	<p><code>minPoolSize</code> を超えて払い出された JDBC コネクションを解放するまでの待ち時間(単位:秒)です。</p> <p>JDBC コネクションは、指定された時間が経過した後で、クローズした時、または、トランザクションが完了した時に解放されます。</p>	15
<p>定義ファイルの格納ディレクトリ名</p> <p>[<code>iPXdaConfigDirectoryName</code>]</p>	<p>ACOS Access Toolkit が使用する定義ファイルの格納ディレクトリ名です。</p> <p>JavaVM 内で最初に使用された値が有効になります。このため、複数の JDBC データソースを使用する際は、設定値を統一してください。</p>	<p><code>\$INSTANCE_ROOT</code></p> <p><code>/config/iPXda</code></p>
<p>動作環境定義ファイル名</p>	<p>ACOS Access Toolkit が使用する動作環境定義ファイル名です。</p>	<code>DataAccessEnvC</code>

[iPXdaConfigFileName]	JavaVM 内で最初に使用された値が有効になります。このため、複数の JDBC データソースを使用する際は、設定値を統一してください。	onfig.xml
トランザクション定義ファイル名 [iPXdaInvokeConfigFileName]	ACOS Access Toolkit が使用するトランザクション定義ファイル名です。 JavaVM 内で最初に使用された値が有効になります。このため、複数の JDBC データソースを使用する際は、設定値を統一してください。	DataAccessTrns Config.xml
オプション定義ファイル名 [iPXdaOptionConfigFileName]	ACOS Access Toolkit が使用するオプション定義ファイル名です。 JavaVM 内で最初に使用された値が有効になります。このため、複数の JDBC データソースを使用する際は、設定値を統一してください。	AAT.properties
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを表わすブール値です。	false
初期接続の接続リトライ有無 [reconnectInitialPool]	初期接続の接続リトライを行うかどうかを表わすブール値です。 データベースサーバの状態監視間隔 [checkServerInterval] で指定された値がリトライ間隔として使用されます。	true

そのほかに、アプリケーションプロセスにおけるリソースのロード設定に関する属性があります。詳細は、「6.4リソースのプロセス単位のロード設定(Standard/Enterprise Editionの場合)」を参照してください。

Dottedname : server.resources.jdbc-datasource.*datasource-name*.property

プロパティ名	説明	既定値
jdbcDriverName	データソースタイプが JDBC の場合にロードされる、JDBC ドライバのクラス名です。	自動設定
networkProtocol	データベースとの通信プロトコルを表わす文字列です。 dataSourceType プロパティの値に応じて初期値が設定されます。	dataSourceType 毎の設定を参照してください
useStaticPool	JDBC コネクションの共有範囲を表わすブール値です。 <b>true</b> : 同一プロセス内の JDBC データソース間で JDBC コネクションを共有します。 <b>false</b> : JDBC データソースのインスタンス間で JDBC コネクションを共有しません。	true
cursorMode	データベースカーソルの利用有無を表わす文字列です。 次のいずれかを設定します。設定可否は dataSourceType 毎に異なります。 <b>cursor</b> : データベースカーソルを利用します。 <b>direct</b> : データベースカーソルを利用しません。	cursor
characterEncoding	文字列引数を送信する際の変換方法を表わす文字列です。 次のいずれかを設定します。設定可否は dataSourceType 毎に異なります。 <b>Unicode</b> : Unicode の文字列をそのまま送信します。 <b>Default</b> : データベースのデフォルトの character encoding に従って変換した文字列を送信します。	Default
databaseName	Transaction サービスの DB 情報の定義名 (DB 識別名) です。 Transaction サービスの社間接続を行うために、Transaction サービスの	なし

	RecoveryServerと連携する場合に設定します。そのほかの場合、設定する必要はありません。	
logLevel	JDBC データソースが AP サーバプロセス外で動作する際のログ出力レベルです。 次のいずれかを設定します。  <i>info</i> : 動作に関する重要な情報を標準出力に出力します。 <i>debug</i> : 動作に関する主な情報を標準出力に出力します。 <i>trace</i> : 全ての情報を標準出力に出力します。 <i>none</i> : ログを全く出力しません。	info
autoRecoveryOnOnePhaseCommit	2 フェーズコミットで、dataSourceType の値が"JDBC"であるコネクションに対するコミットが失敗した場合に、トランザクション全体を自動的にロールバックするかどうかを表わすブール値です。  コミットが失敗した場合の結果は、通常ロールバックされた状態になりますが、データベースサーバ側でのコミット処理完了後に通信障害が発生した場合など、稀にコミットされていることがあります。対策としては、この設定に false を設定することで自動的にロールバックされないようにしておき、コミットの結果を JDBC ドライバやデータベースが提供する運用操作や例外メッセージに従って確認した上で、トランザクションを復旧してください。トランザクションの復旧操作については、運用編(Transaction サービスの運用操作)を参照してください。  <i>true</i> : トランザクション全体を自動的にロールバックします。 <i>false</i> : トランザクションの完了処理を保留します。	true

Java のシステムプロパティに、次の項目を設定することができます。JDBC データソースの MO を介して設定する項目に優先する、プロセス内共通の設定となります。通常のログ出力内容の設定方法については、「運用編(ロギング)」を参照してください。

プロパティ名	説明	既定値
wojdbc.Log	設定内容は、logLevel プロパティと同様です。	none
wojdbc.CheckGarbageOption	設定内容は、checkGarbageOption プロパティと同様です。	none
wojdbc.CheckGarbageInterval	設定内容は、checkGarbageInterval プロパティと同様です。	0
wojdbc.CursorMode	設定内容は、cursorMode プロパティと同様です。	cursor
wojdbc.CharacterEncoding	設定内容は、characterEncoding プロパティと同様です。	Default
wojdbc.MaxIdleTime	設定内容は、maxIdleTime プロパティと同様です。	0

### 9.3.Oracleでの設定内容

Oracle の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに **JDBC\_EX\_Oracle** を設定します。また、EJB のトランザクション(JTA)と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名	次のようなデータベースとの接続文字列を設定します。	←(同じです)

[dataSourceName]	"jdbc:oracle:thin:@host:1521:orcl" "jdbc:oracle:oci:@orcl"	
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 か4を設定します。	←(同じです)
サーバ名 [serverName]	---(設定しません)	←(同じです)
ポート番号 [portNumber]	---(設定しません)	←(同じです)
ステートメントの最大プール数 [maxStatements]	プールするステートメントの最大数を設定します。	←(同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを使用するかどうか [useOneConnectionPerTransaction]	JTA のトランザクション毎に 1 つの物理コネクションを使用するかどうかを設定します。	---(設定しません)
データベースクラスタの使用有無 [useDatabaseCluster]	Oracle RAC を利用し、かつ、2 フェーズコミットを行う場合に true を設定します。	---(設定しません)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	---(設定しません)	←(同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	---(設定しません)
その他	必要に応じて設定します。	←(同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	---(設定しません)
その他	jdbcDriverName と cursorMode、characterEncoding、networkProtocol、autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	←(同じです)

## 9.4.DB2 での設定内容

DB2 Universal JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに **JDBC\_EX\_DB2** を設定します。また、EJB のトランザクション(JTA)と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	データベース名を設定します。	←(同じです)
JDBC 仕様のメジャーバージョン	2 か 3 を設定します。	←(同じです)

[jdbcMajorVersion]		
サーバ名 [serverName]	データベースサーバ側のサーバ名 (TCP/IP ホスト名) を設定します。	← (同じです)
ポート番号 [portNumber]	データベースサーバ側のポート番号を設定します。 デフォルト値は 50000 です。	← (同じです)
ステートメントの最大プール数 [maxStatements]	--- (設定しません)	← (同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを使用するかどうか [useOneConnectionPerTransaction]	--- (設定しません。必ず、true として処理が行われます)	← (同じです)
データベースクラスターの使用有無 [useDatabaseCluster]	--- (設定しません)	← (同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	--- (設定しません)	← (同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	--- (設定しません)
その他	必要に応じて設定します。	← (同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	--- (設定しません)
その他	jdbcDriverName と cursorMode、characterEncoding、networkProtocol、autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	← (同じです)

## 9.5. Microsoft SQL Serverでの設定内容

Microsoft SQL Server の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに **JDBC\_EX\_MSSqlServer** を設定します。また、EJB のトランザクション (JTA) と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	データベース名を設定します。	← (同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 を設定します。	← (同じです)



サーバ名 [serverName]	データベースサーバ側のサーバ名 (TCP/IP ホスト名) を設定します。	← (同じです)
ポート番号 [portNumber]	データベースサーバ側のポート番号を設定します。 デフォルト値は 1433 です。	← (同じです)
ステートメントの最大プール数 [maxStatements]	--- (設定しません)	← (同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを使用するかどうか [useOneConnectionPerTransaction]	JTA のトランザクション毎に 1 つの物理コネクションを使用するかどうかを設定します。	--- (設定しません)
データベースクラスタの使用有無 [useDatabaseCluster]	--- (設定しません)	← (同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	--- (設定しません)	← (同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	--- (設定しません)
その他	必要に応じて設定します。	← (同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
cursorMode	データベースカーソルの利用有無を設定します。 設定値は SelectMethod プロパティ値として JDBC ドライバに設定されます。	← (同じです)
characterEncoding	文字列引数を送信する際の変換方法を設定します。 設定値は Unicode かどうかを表すブール値に変換され、SendStringParametersAsUnicode プロパティ値として JDBC ドライバに設定されます。ただし、Microsoft SQL Server 2005 の JDBC ドライバを使用して JTA 連携を行う場合、必ず、"Unicode" として処理が行われます。	← (同じです)
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	--- (設定しません)
その他	jdbcDriverName と networkProtocol、autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	← (同じです)

## 9.6.Sybaseでの設定内容

Sybase の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに **JDBC\_EX\_Sybase** を設定します。また、EJB のトランザクション (JTA) と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合
----------------------	------------------	--------------

		の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	データベース名を設定します。	←(同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	---(設定しません)	←(同じです)
サーバ名 [serverName]	データベースサーバ側のサーバ名(TCP/IP ホスト名)を設定 します。	←(同じです)
ポート番号 [portNumber]	データベースサーバ側のポート番号を設定します。 デフォルト値は 6001 です。	←(同じです)
ステートメントの最大プール数 [maxStatements]	---(設定しません)	←(同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを 使用するかどうか [useOneConnectionPerTransaction]	JTA のトランザクション毎に 1 つの物理コネクションを使用す るかどうかを設定します。	---(設定しません)
データベースクラスタの使用有無 [useDatabaseCluster]	---(設定しません)	←(同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	---(設定しません)	←(同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	---(設定しません)
その他	必要に応じて設定します。	←(同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合 の設定内容
cursorMode	データベースカーソルの利用有無を設定します。 <i>cursor</i> を設定した場合、REPEAT_READ プロパティ値として false が JDBC ドライバに 設定されます。 <i>direct</i> を設定した場合、REPEAT_READ プロパティ値として true が JDBC ドライバに 設定されます。	←(同じです)
characterEncoding	文字列引数を送信する際の変換方法を設定します。 <i>Unicode</i> を設定した場合、JCONNECT_VERSION プロパティ値として 6 が JDBC ドライ バに設定されます。	←(同じです)
networkProtocol	データベースとの通信プロトコルを設定します。デフォルト値は "Tds" です。	←(同じです)
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	---(設定しません)

その他	jdbcDriverName と networkProtocol、autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	←(同じです)
-----	---	---------

## 9.7.SequeLinkでの設定内容

SequeLink の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに *JDBCCEX\_SequeLink* を設定します。また、EJB のトランザクション (JTA) と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合 の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	JDBCCEX_SequeLink を設定します。	←(同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 を設定します。	←(同じです)
サーバ名 [serverName]	データベースサーバ側のサーバ名 (TCP/IP ホスト名) を設定 します。	←(同じです)
ポート番号 [portNumber]	データベースサーバ側のポート番号を設定します。 デフォルト値は 19996 です。	←(同じです)
ステートメントの最大プール数 [maxStatements]	プールするステートメントの最大数を設定します。	←(同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを 使用するかどうか [useOneConnectionPerTransaction]	JTA のトランザクション毎に 1 つの物理コネクションを使用す るかどうかを設定します。	--- (設定しません)
データベースクラスタの使用有無 [useDatabaseCluster]	--- (設定しません)	←(同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	--- (設定しません)	←(同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	--- (設定しません)
その他	必要に応じて設定します。	←(同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合 の設定内容
networkProtocol	データベースとの通信プロトコルを設定します。デフォルト値は、“Tcpip”です。	←(同じです)
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	--- (設定しません)

その他	jdbcDriverName と cursorMode、characterEncoding、 autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	←(同じです)
-----	---	---------

## 9.8.Cloudscapeでの設定内容

Cloudscape の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに *JDBC\_EX\_Cloudscape* を設定します。また、EJB のトランザクション(JTA)と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	データベース名を設定します。	←(同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 を設定します。	←(同じです)
サーバ名 [serverName]	--- (設定しません)	←(同じです)
ポート番号 [portNumber]	--- (設定しません)	←(同じです)
ステートメントの最大プール数 [maxStatements]	--- (設定しません)	←(同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクション を使用するかどうか [useOneConnectionPerTransaction]	JTA のトランザクション毎に 1 つの物理コネクシ ョンを使用するかどうかを設定します。	--- (設定しません)
データベースクラスタの使用有無 [useDatabaseCluster]	--- (設定しません)	←(同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	--- (設定しません)	←(同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユ ーザの JDBC コネクションを使用するかどうかを 設定します。	--- (設定しません)
その他	必要に応じて設定します。	←(同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
networkProtocol	--- (設定しません)	データベースとの通信プロトコルを設定します。 rmi を指定するか指定なしのいずれかです。rmi を指定する とローカルホストの RmiJdbc サーバにデフォルトのポート番 号 1099 で接続します。デフォルト値は指定なしです。 サーバ名とポート番号を指定する場合は

		"rmi://hostname:port 番号/"と記述します。
databaseName	Transaction サービスの DB 識別名を設定します。 通常、設定する必要はありません。	--- (設定しません)
その他	jdbcDriverName と characterEncoding、 cursorMode 以外のプロパティを、必要に応じて設定します。	←(同じです)

## 9.9.PostgreSQLでの設定内容

PostgreSQL の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに *JDBC\_EX\_PostgreSQL* を設定します。また、EJB のトランザクション (JTA) と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	データベース名を設定します。	←(同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 を設定します。	←(同じです)
サーバ名 [serverName]	データベースサーバ側のサーバ名 (TCP/IP ホスト名) を設定します。	←(同じです)
ポート番号 [portNumber]	データベースサーバ側のポート番号を設定します。デフォルト値は 5432 です。	←(同じです)
ステートメントの最大プール数 [maxStatements]	--- (設定しません)	←(同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを使用するかどうか [useOneConnectionPerTransaction]	--- (設定しません。必ず、true として処理が行われます)	←(同じです)
データベースクラスタの使用有無 [useDatabaseCluster]	--- (設定しません)	←(同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	--- (設定しません)	←(同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	--- (設定しません)
その他	必要に応じて設定します。	←(同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	--- (設定しません)
その他	jdbcDriverName と cursorMode、characterEncoding、networkProtocol、autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	← (同じです)

## 9.10. Apache Derby での設定内容

Apache Derby の JDBC ドライバの DataSource インタフェースを使用する場合、dataSourceType プロパティに *JDBC\_EX\_Derby* を設定します。また、EJB のトランザクション (JTA) と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	データベース名を設定します。	← (同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 か 4 を設定します。	← (同じです)
サーバ名 [serverName]	データベースサーバ側のサーバ名 (TCP/IP ホスト名) を設定します。	← (同じです)
ポート番号 [portNumber]	データベースサーバ側のポート番号を設定します。デフォルト値は 1527 です。	← (同じです)
ステートメントの最大プール数 [maxStatements]	--- (設定しません)	← (同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを使用するかどうか [useOneConnectionPerTransaction]	JTA のトランザクション毎に 1 つの物理コネクションを使用するかどうかを設定します。	← (同じです)
データベースクラスタの使用有無 [useDatabaseCluster]	--- (設定しません)	← (同じです)
ACOS Access Toolkit 関連の設定 [iPXdaConfigDirectoryName] [iPXdaConfigFileName] [iPXdaInvokeConfigFileName] [iPXdaOptionConfigFileName]	--- (設定しません)	← (同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	JTA (EJB) の同一トランザクション内で、複数ユーザの JDBC コネクションを使用するかどうかを設定します。	--- (設定しません)
その他	必要に応じて設定します。	← (同じです)

プロパティ名	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
databaseName	Transaction サービスの DB 識別名を設定します。通常、設定する必要はありません。	--- (設定しません)
その他	jdbcDriverName と cursorMode、characterEncoding、networkProtocol、autoRecoveryOnOnePhaseCommit 以外のプロパティを、必要に応じて設定します。	← (同じです)

## 9.11.JDBC での設定内容

そのほかの JDBC ドライバを使用する場合、dataSourceType プロパティに「JDBC」を設定します。また、EJB のトランザクション (JTA) と連携するかどうかによって、次の内容を設定します。

属性名 [attribute-name]	JTA と連携する場合の設定内容	JTA と連携しない場合の設定内容
JDBC URL またはデータベース名、データソース名 [dataSourceName]	例えば次のような、データベースとの接続文字列を設定します。 <pre>"jdbc:oracle:thin:@host:1521:orcl"</pre> <pre>"jdbc:oracle:oci:@orcl"</pre> <pre>"jdbc:microsoft:sqlserver://host:1433"</pre> <pre>"jdbc:sqlserver://host:1433"</pre> <pre>"jdbc:db2://host:50000/SAMPLE"</pre> <pre>"jdbc:sybase:Tds:host:6001"</pre> <pre>"jdbc:sequelink://host:19996"</pre> <pre>"jdbc:cloudscape:/opt/j2ee/cloudscape/tmpdb"</pre> <pre>"jdbc::db2j:toursDB"</pre> <pre>"jdbc:postgresql://host:4321/dbname"</pre> <pre>"jdbc:odbc:dbname"</pre> <pre>"jdbc:nec-acos4-dataaccess:dbname"</pre> <pre>"jdbc:nec-acos4-dataaccess-ex:dbname"</pre> <pre>"jdbc:derby://host:1527/dbname"</pre>	← (同じです)
JDBC 仕様のメジャーバージョン [jdbcMajorVersion]	JDBC 仕様のメジャーバージョンを設定します。 2 か 3 か 4 を設定します。	← (同じです)
サーバ名 [serverName]	--- (設定しません)	← (同じです)
ポート番号 [portNumber]	--- (設定しません)	← (同じです)
ステートメントの最大プール数 [maxStatements]	--- (設定しません)	← (同じです)
JTA 連携有無 [useJTA]	true を設定します。	false を設定します。
JTA のトランザクション毎に1つの物理コネクションを使用するかどうか	--- (設定しません。必ず、true として処理が行われます)	--- (設定しません)

[useOneConnectionPerTransaction]		
データベースクラスタの使用有無 [useDatabaseCluster]	--- (設定しません)	← (同じです)
定義ファイルの格納ディレクトリ名 [iPXdaConfigDirectoryName]	ACOS Access Toolkit の JDBC ドライバを利用する場合に 定義ファイルの格納ディレクトリ名を設定します。	← (同じです)
動作環境定義ファイル名 [iPXdaConfigFileName]	ACOS Access Toolkit の JDBC ドライバを利用する場合に 動作環境定義ファイル名を設定します。	← (同じです)
トランザクション定義ファイル名 [iPXdaInvokeConfigFileName]	ACOS Access Toolkit の JDBC ドライバを利用する場合にト ランザクション定義ファイル名を設定します。	← (同じです)
オプション定義ファイル名 [iPXdaOptionConfigFileName]	ACOS Access Toolkit の JDBC ドライバを利用する場合に オプション定義ファイル名を設定します。	← (同じです)
トランザクション毎の複数ユーザ使用有無 [useMultiUsersPerTransaction]	--- (設定しません。同一トランザクション内で、複数ユーザの JDBC コネクションを同時に使用することはできません。)	--- (設定しません)
その他	必要に応じて設定します。	← (同じです)

プロパティ名	設定内容	
jdbcDriverName	dataSourceName の説明で例示した接続文字列を使用する JDBC ドライバの名前は設 定する必要がありません。それ以外の JDBC ドライバを使用する場合には、JDBC ドライ バのクラス名を設定します。	← (同じです)
autoRecoveryOnOneP haseCommit	2フェーズコミットを実行する場合に、必要に応じて設定します。	--- (設定しません)
その他	cursorMode と characterEncoding、networkProtocol、databaseName 以外 のプロパティを、必要に応じて設定します。	← (同じです)

## 9.12. クラスパス設定

JDBC データソースを使用する場合は、使用するデータベースまたは JDBC ドライバに応じて次のファイルをクラスパスに追加する  
必要があります。

クラスパスの設定方法については「[運用編\(ドメインの運用\)](#)」の「3.7 Java VM環境の設定」を参照してください。設定では、複数バ  
ージョンのクラスが混在しないように、必要なファイルだけをクラスパスに追加してください。

データベースまたは JDBC ドライバ	ファイル名
Oracle	Oracle 11g を使用する場合 Java1.5 以前の場合 ojdbc5.jar orai18n.jar (\$ORACLE_HOME/jlib を参照) Java6 以降の場合 ojdbc6.jar orai18n.jar (\$ORACLE_HOME/jlib を参照) Oracle 10g を使用する場合 ojdbc14.jar orai18n.jar (\$ORACLE_HOME/jdbc/lib か \$ORACLE_HOME/jlib を参照) Oracle 9.2 を使用する場合



	ojdbc14.jar    nls_charset12.jar それ以外の場合 classes12.zip    nls_charset12.zip
DB2	db2jcc.jar db2jcc_license_cu.jar または db2jcc_license_cisuz.jar
Microsoft SQL Server	SQL Server 2005 を使用する場合 sqljdbc.jar SQL Server 2000 を使用する場合 mssqlserver.jar    msbase.jar    msutil.jar
Sybase	jconn2.jar
SequeLink	sljc.jar sljcx.jar
Cloudscape	Cloudscape バージョン 5 以降の場合 db2j.jar    db2jtools.jar それ以外の場合 RMI を使用しない場合 cloudscape.jar RMI を使用する場合 cloudclient.jar    rmiJdbc.jar
PostgreSQL	PostgreSQL バージョン 8.1 を使用する場合 postgresql-8.1-404.jdbc3.jar PostgreSQL バージョン 8.2 を使用する場合 postgresql-8.2-506.jdbc3.jar PostgreSQL バージョン 8.3 を使用する場合 postgresql-8.3dev-601.jdbc3.jar (jar ファイル名は製品バージョン毎に異なります)
Apache Derby	derbyclient.jar

## 9.13. その他の設定

Oracle の OCI JDBC ドライバなど、タイプ2の JDBC ドライバを利用する場合には、必要に応じて環境変数の設定作業を行う必要があります。UNIX 版では、Java からロードする native のライブラリファイルに適切なアクセス権が付与されていることをご確認ください。また、EJB のトランザクション (JTA) と連携する場合には、必要に応じてデータベースサーバ側の設定を行う必要があります。詳細は、オンラインリファレンスのマニュアル一覧にある注意制限事項の「6. JDBC データソース」を参照してください。

そのほか、EJB のトランザクション (JTA) と連携する場合には、「セットアップガイド 2.3.3. データベースを使用するための準備作業 (Java)」をご覧ください。

# 10.Transactionサービスに関する設定

Transaction サービス(JTA)に関する設定について説明します。

Transaction サービスに関するコンフィグレーションの設定、および稼働中のトランザクション情報の取得は、基本的には統合運用管理ツール、および `otxadmin` コマンドを使い、MO を経由して行いますが、いくつか Transaction サービス専用のコマンドも提供しています。前者については次に記述します。コマンドについての詳細は「運用コマンドリファレンスマニュアル」をご参照ください。

また、Transaction サービスでは、実行中のトランザクションに障害が発生した際の復旧処理を行う機能 (Recovery Coordination Service:RCS) を提供しています。RCS によるトランザクションの復旧を行うために必要となる、データベースやコネクタリソースの情報定義もここで実施します。

## 10.1.Transactionサービス全般の設定項目・設定方法

Transaction サービスの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

補足:

設定された属性は以下のディレクトリ配下に格納されます。

```
${INSTANCE_ROOT}/config/persistent-mbean/transactionservice
```

このファイルが存在しない場合にドメインを起動すると、Transaction サービスの設定内容は全て既定値で動作します。このファイルが存在する場合にドメインを再起動した場合は、その値が Transaction サービスの設定内容として使用されます。**このファイルの内容を手で書き換えることはしないでください。**

### Transactionサービスで使用する構成情報設定ファイル

Transaction サービスでは、内部で使用するための構成情報設定ファイル、および実行中のトランザクションの状態などを「`/${INSTANCE_ROOT}/config/TS`」ディレクトリにしています。本ディレクトリ、およびその配下のファイルの削除はしないでください。トランザクションの結果に矛盾が生じる可能性があります。また内容を直接修正することしないでください。

## 10.2.トランザクション情報の表示

各トランザクションに関する情報を管理する MO の種別は「TSTxInfo」、MO の名前は「トランザクション識別子」となります。

統合運用管理ツールでは、ツリーの「<ドメイン名>」-「アプリケーションサーバ」-「Transaction サービス」-「Transaction 情報」-「トランザクション識別子」をクリックすると、トランザクションの状態や識別子名などが右側に表示されます。また、コマンドで各項目を参照するには「`server.transactionservice.tstxlist.<トランザクション識別子>`」を使用します。例えばトランザクション識別子が「0209210086910000」のトランザクションの場合、「`server.transactionservice.tstxlist.0209210086910000`」になります。

### コマンドによる項目の参照

項目の参照には、以下の `get` コマンドを使用してください。

```
get server.transactionservice.tstxlist.<トランザクション識別子>.attribute-name
```

### 項目の設定

設定可能な項目はありません。

## 10.3.Transactionサービスに関する設定

以下は MO で設定可能トランザクションサービス設定項目の一覧です。

Dottedname : server.transactionservice

項目 [属性名 : attribute-name]	説明	既定値
RCS (C++) ポート番号 [rcs-cpp-port]	C++アプリケーションで動作するトランザクションを管理するプロセス (C++版 RCS) が使用するポート番号です。設定可能範囲は 1~65,535 です。また一度決定した番号を変えると未完了トランザクションの復旧が失敗することがありますので注意してください。ポート番号を変更する場合は、復旧中のトランザクションが存在しない状態で実施してください。「rcs-cpp-startup」を true にした場合は必ず指定する必要があります。なお、Web Edition、および Standard-J Edition では使われない項目です。 この項目の変更は Transaction サービスの再起動後から有効となります。	5965
C++ AP 用の RCS を自動起動する [rcs-cpp-startup]	C++版 RCS プロセスを Transaction サービス起動時に自動的に起動するか指定します。C++アプリケーションで Transaction サービス (OTS) を利用したトランザクション管理を実施する場合はチェックしてください。 なお、Web Edition、および Standard-J Edition ではチェックできません。 この項目の変更は Transaction サービスの再起動後から有効となります。	true(自動起動する)
クライアントトランザクションを動作 [rcs-proxy-startup]	クライアントアプリケーション上で分散トランザクションを開始し、データベースの更新を行う場合はチェックしてください。これによってクライアントアプリケーション障害時に実行中だったトランザクションの復旧を Transaction サービスで自動的に実施します。なお、Web Edition、ではチェックできません。 この項目の変更は Transaction サービスの再起動後から有効となります。	false (自動起動しない)
トレース情報出力先ディレクトリ [trace-file-path]	Transaction サービスに関連するプロセスによって出力されるトレースファイルの出力先ディレクトリ名です。ファイルは 3 種類 (Web Edition では 1 種類) 存在しますが、ファイルごとに出力先を変えることはできません。またファイル名自体を変更することもできません。 この項目の変更は RCD/RCS プロセスについては Transaction サービスの再起動後から、AP 上で動作する Transaction サービス機能については設定直後に起動したもから有効となります。 <b>ディレクトリは存在するものを指定してください。</b>	\${INSTANCE_ROOT} /logs/TS
トレースファイルサイズ [trace-size]	Transaction サービスで出力するトレースファイルのサイズです。設定可能範囲は 1~512,000 (KB) です。ファイルは 3 種類 (Web Edition では 1 種類) 存在しますが、ファイルごとにサイズを変えることはできず、変更は 3 ファイル全てに適用されます。 この項目の変更は RCD/RCS プロセスについては Transaction サービスの再起動後から、AP 上で動作する Transaction サービス機能については設定直後に起動したもから有効となります。	1024 (KB)
トレースレベル	Transaction サービスで出力するトレースのレベルです。設定可能範囲は	2

[trace-level]	1 (エラーレベル) ~ 5 (デバッグレベル) です。この項目の変更は設定直後に有効となります。なおファイルは 3 種類 (Web Edition では 1 種類) 存在しますが、ファイルごとにレベルを変えることはできず、変更は 3 ファイル全てに適用されます。	
トレースファイル初期化を行う [trace-init]	Transaction サービス開始時にトレースファイルの内容を初期化するかどうかです。"true" あるいは "false" を指定します。ファイルは 3 種類 (Web Edition では 1 種類) 存在しますが、ファイルごとに属性を変えることはできず、変更は 3 ファイル全てに適用されます。  この項目の変更は RCD/RCS プロセスについては Transaction サービスの再起動後から、AP 上で動作する Transaction サービス機能については設定直後に起動したものから有効となります。	false (初期化しない)
コミットリトライ回数 [retry-num]	トランザクションのコミットが失敗した際のリトライ回数です。この回数分リトライを実施しても状況が変わらない場合、トランザクションサービスが自動的にトランザクションの状態をヒューリスティックに設定します。設定可能範囲は 0 ~ 32767 回です。  この項目の変更は設定直後に有効となります。	5(回)
コミットリトライ間隔 [retry-interval]	コミットリトライの時間間隔です。設定可能範囲は 2 ~ 32767 秒です。  この項目の変更は設定直後に有効となります。	600(秒)
トランザクションタイムアウト時間 [tx-timeout]	トランザクションのタイムアウト時間です。  トランザクションを開始してからこの時間が経過しても完了していない場合はトランザクションサービスが自動的にロールバック処理を実施します。設定可能範囲は 0 ~ 32767 秒です。また 0 秒を指定した場合、タイムアウトは発生しません。  この項目の変更は設定直後に有効となります。	600(秒)
トランザクション復旧開始時間 [recover-start-timer]	復旧したトランザクションについて、復旧が完了してから自動的に完了処理を開始するまでの時間を設定します。設定可能範囲は 0 ~ 32767 秒です。  この項目の変更は設定直後に有効となります。	60(秒)
リソース復旧開始時間 [recovery-time]	障害などによって、トランザクションマネージャからリソースへの指示 (commit, rollback, forget 要求) が出されない場合に、リソースが自動的に復旧を開始するまでの時間を指定します。  設定可能範囲は 0 ~ 32767 秒です。0 はタイムアウトなしを意味します。  この項目の変更は設定直後に有効となります。	60(秒)
リソース復旧間隔 [recovery-interval]	リソースが復旧に失敗した場合に、次に復旧処理をリトライするまでの間隔を指定します。  設定可能範囲は 0 ~ 32767 秒、既定値は 2 秒です。  この項目の変更は設定直後に有効となります。	2(秒)

Dottedname : server.transactionservice.tstxlist

項目 [属性名 : attribute-name]	説明	既定値
トランザクション情報採取レベル [monitor-level]	Transaction サービスに関する統計情報採取レベルです。ここでは参照のみ可能です。  設定については、統合運用管理ツールでは「モニタリングサービス」-「モジュールモニタリングレベル」ツリーをクリックした際に右側に表示される、「transactionsservice モニタリングレベル」で行うことができます。 otxadmin コマンドでも設定可能ですが、方法については運用管理(モニタリング)編を参照してください。  この項目の変更は設定直後から有効となります。	0 (モニタ情報を採取しない)

Dottedname : server.transactionsservice.tstxlist.<トランザクション識別子>

これらの値は全て参照のみ可能です。

項目 [属性名 : attribute-name]	説明	既定値
トランザクション識別子 [stxid]	Transaction サービスの内部でそれぞれのトランザクションを一意に識別するために払い出した数字です。	無し
状態 [status]	トランザクションの状態を表します。それぞれの詳細については運用管理(モニタリング)編をご参照ください。	無し
開始時間 [starttime]	トランザクションを開始した日時です。	無し
Otid [otid]	DBMS 側でトランザクションを管理するのに使われる一意の識別子です。	無し
所有者 [owner]	トランザクションの管理を行っているのがどこかをあらわします。「Active」あるいは「Recover」のいずれかが表示されます。通常トランザクションは AP の内部で管理されます。その場合は「Active」と表示されます。ただしトランザクションが完了しないうちに AP が停止した場合、その管理が RCS プロセスに移管され、完了処理を自動的に継続します。この場合「Recover」と表示されます	無し

## 10.4. リソースを管理するためのコマンド

2フェーズコミットの途中で障害が発生した場合、Transaction サービスは、トランザクションのリカバリ処理を実行します。リカバリ処理で使用するデータベースやコネクタリソースの情報を、Transaction サービスのリソースとして定義します。

### Javaアプリケーション用のリソース

JDBC データソースやコネクタリソース、JMS のコネクションファクトリリソースへの参照情報 (JNDI サーバへの登録名と、上書き設定するユーザ名およびパスワード) を、Transaction サービスの JDBC リソースまたは JCA リソースに設定します。

JDBC リソースは、「JDBC データソースのテスト」操作を行った時や、トランザクション実行時に自動的に生成されます。自動生成された JDBC リソースをそのまま使用する場合は、「11.5 データベースを使用するための準備」に記載されているトランザクションのリカバリを行うための権限を、JDBC データソースの定義時に指定したユーザに直接付与してください。リカバリを行う時にだけ特権を持ったユーザを使用する場合は、自動生成された JDBC リソースのユーザ名とパスワードを書き換えてください。

### C++アプリケーション用のリソース

Transaction サービスの C++ XA リソースでは、トランザクションのリカバリを行うためだけでなく、アプリケーションからデータベースに接続するための接続情報を設定します。また、アプリケーションから、Oracle のオープン文字列中のユーザ名を使い分ける場合には、オプションとして、AP 専用のリソース定義を行うことができます。

Transaction サービスの C++用のリソース定義を行った後は、そのリソースの名前を、「tpssystem.applicationGroups.アプリケーショングループ名.processGroups.プロセスグループ名」で、Transaction Service のデータベース識別名として指定します。

これらのリソースについては、以降の MO で設定、または、参照することができます。

## JDBCリソース

MO 名は、「jdbc-resource.resource-name」です。resource-name は、JDBC データソースの名前と同じです。

コマンドで各項目を参照・設定する際は、「server.transaction.service.resources.jdbc-resource.resource-name」という名前を指定します。また、統合運用管理ツールでは、ドメインツリーの「WebOTXAdmin[ホスト名].ドメイン名.server.transactionservice.jdbc-resource.resource-name」を選択します。項目の設定方法については「2各サービスの属性の設定」を参照してください。

属性名 [attribute-name]	説明	既定値
リソース名 [name]	JDBC データソースの参照定義の名前 (JDBC データソースの JNDI 名) です。 NULL 文字を含め、最大 256 文字の名前を指定してください。 データソースの種類 [dataSourceType] が "JDBC" の場合、その JDBC データソースを JDBC リソースとして登録しないでください。	なし
接続状態 [status]	データベースとの接続状態です。 値は、Opened、Closing、Closed のいずれかになります。トランザクションのリカバリ処理を実行中の場合に stop オペレーションを実行すると、リカバリ処理が完了するまで Closing になります。	なし
ユーザ名 [user-name]	トランザクションのリカバリを行うための、特別な権限を持ったユーザ名です。	なし
パスワード [password]	データベースと接続するためのパスワードです。	なし

## JCAリソース

MO 名は、「jca-resource.resource-name」です。resource-name は、コネクタリソースまたは JMS のコネクションファクトリリソースの名前と同じです。

コマンドで各項目を参照・設定する際は、「server.transaction.service.resources.jca-resource.resource-name」という名前を指定します。また、統合運用管理ツールでは、ドメインツリーの「WebOTXAdmin[ホスト名].ドメイン名.server.transactionservice.jca-resource.resource-name」を選択します。項目の設定方法については「2各サービスの属性の設定」を参照してください。

属性名 [attribute-name]	説明	既定値
リソース名 [name]	コネクタリソースまたは JMS リソースの参照定義の名前 (該当リソースの JNDI 名) です。 NULL 文字を含め、最大 256 文字の名前を指定してください。	なし
接続状態 [status]	コネクタリソースとの接続状態です。 値は、Opened、Closing、Closed のいずれかになります。トランザクションのリカバリ処理を実行中の場合に stop オペレーションを実行すると、リカバリ処理が完了するまで Closing になります。	なし
ユーザ名 [user-name]	コネクタリソースと接続するためのユーザ名です。	なし
パスワード [password]	コネクタリソースと接続するためのパスワードです。	なし
ConnectionRequestInfo の使用有無	コネクタリソースとの接続で、javax.resource.spi.ConnectionRequestInfo クラスを使用するかどうかを表わすブール値です。デフォルトでは、	false

[use-con-req-info]	javax.resource.spi.security.PasswordCredential クラスを使用します。 JMS リソースを使用する場合は、ConnectionRequestInfo クラスを使用しません。	
ConnectionRequestInfo クラス名 [con-req-info-name]	コネクタリソースとの接続情報を格納するための ConnectionRequestInfo クラスの名前です。コネクタリソースの実装クラスを使用する必要がある場合に、そのクラス名を指定してください。	なし

## C++ XAリソース

MO 名は、「cppxa-resource.resource-name」です。resource-name は、C++ XA リソースの名前です。

コマンドで各項目を参照・設定する際は、「server.transaction.service.resources.cppxa-resource.resource-name」という名前を指定します。また、統合運用管理ツールでは、ドメインツリーの「WebOTXAdmin[ホスト名].ドメイン名.server.transactionservice.cppxa-resource.resource-name」を選択します。項目の設定方法については「2各サービスの属性の設定」を参照してください。

属性名 [attribute-name]	説明	既定値
リソース名 [name]	C++ XA リソースの名前です。  NULL 文字を含め、最大16文字の名前を指定してください。	なし
接続状態 [status]	データベースとの接続状態です。  値は、Opened、Closing、Closed のいずれかになります。トランザクションのリカバリ処理を実行中の場合に stop オペレーションを実行すると、リカバリ処理が完了するまで Closing になります。	なし
XA リソースタイプ [xa-resource-type]	データベースまたはメッセージキューベンダが提供するインタフェースの種別を表わす文字列です。値は、次のいずれかになります。  Oracle_XA10G : Oracle 10g の XA インタフェースを使用します。 Oracle_XA91 : Oracle 9i の XA インタフェースを使用します。 Oracle_XA81 : Oracle 8i の XA インタフェースを使用します。 MsSql_XA : Microsoft SQL Server の XA インタフェースを使用します。 IBMMq_XA : IBM WebSphere MQ の XA インタフェースを使用します。 Oracle_AX10G : Oracle 10g の AX インタフェースを使用します。 Oracle_AX91 : Oracle 9i の AX インタフェースを使用します。 Oracle_AX81 : Oracle 8i の AX インタフェースを使用します。  XA インタフェースでは、アプリケーションのメソッドが呼び出される際に、事前に、データベース側のトランザクションが開始されます。これに対して、AX インタフェースでは、アプリケーションから SQL 命令を実行した際にトランザクションが開始されます。  アプリケーションに SQL 命令を実行しないメソッドがある場合や、メソッド毎に使用するデータベースが異なる場合には、AX インタフェースを利用した方が効率的です。	なし
オープン文字列 [open-string]	データベースに接続するために使用する文字列です。  Oracle を使用する場合は、オープン文字列を指定します。オープン文字列の詳細は、Oracle のリファレンスマニュアルを参照してください。IBM WebSphere MQ を使用する場合は、キュー管理プログラム名を指定します。Microsoft SQL Server を使用する	なし

	場合は、指定しません。	
クローズ文字列 [close-string]	クローズ文字列を指定する必要はありません。	なし

## AP用C++ XAリソース

MO名は、「`cppxa-ap-resource.parent-name.resource-name`」です。`parent-name`は、上位のC++ XAリソースの名前です。また、`resource-name`は、AP用C++ XAリソースの名前です。

コマンドで各項目を参照・設定する際は、「`server.transaction.service.resources.cppxa-resource.parent-name.resource-name`」という名前を指定します。また、統合運用管理ツールでは、ドメインツリーの「WebOTXAdmin[ホスト名].ドメイン名.`server.transactionservice.cppxa-resource.parent-name.resource-name`」を選択します。項目の設定方法については「2各サービスの属性の設定」を参照してください。

属性名 [attribute-name]	説明	既定値
リソース名 [name]	AP用C++ XAリソースの名前です。 NULL文字を含め、最大16文字の名前を指定してください。	なし
XAリソースタイプ [xa-resource-type]	データベースまたはメッセージキューベンダが提供するインタフェースの種別を表わす文字列です。値は、次のいずれかになります。  <code>Oracle_XA10G</code> : Oracle 10gのXAインタフェースを使用します。 <code>Oracle_XA9I</code> : Oracle 9iのXAインタフェースを使用します。 <code>Oracle_XA8I</code> : Oracle 8iのXAインタフェースを使用します。 <code>Oracle_AX10G</code> : Oracle 10gのAXインタフェースを使用します。 <code>Oracle_AX9I</code> : Oracle 9iのAXインタフェースを使用します。 <code>Oracle_AX8I</code> : Oracle 8iのAXインタフェースを使用します。	なし
オープン文字列 [open-string]	データベースに接続するために使用する文字列です。 Oracleを使用する場合は、オープン文字列を指定します。オープン文字列の詳細は、Oracleのリファレンスマニュアルを参照してください。IBM WebSphere MQを使用する場合は、キュー管理プログラム名を指定します。Microsoft SQL Serverを使用する場合は、指定しません。	なし
クローズ文字列 [close-string]	クローズ文字列を指定する必要はありません。	なし



# 11.Object Brokerに関する設定

Object Broker に関する設定について説明します。

## 11.1.Object Broker設定項目・設定方法

Object Brokerの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

### 11.1.1.Object Broker設定項目一覧

以下はMOで設定可能なObject Broker設定項目一覧です。

Dottedname : server.objectbrokerconfig

objectbrokerconfigでの設定項目は、[共通] [Cpp] [Java] の3つのタブに分かれています。

共通タブ : Object Broker C++およびObject Broker Javaの共通の設定を行うことができます。

属性名 (attribute-name)	説明	既定値
ORB インストール ディレクトリ (OrbRoot)	ORB がインストールされているディレクトリをフルパスで表示します。このディレクトリには conf および log サブディレクトリが必要です。	参照のみ可能な項目です。既定値はありません。
oad のポート番号 (OadPort)	oad が使用するポート番号を指定します。 ORB 通信を行うクライアントとサーバ間で同一のポート番号を使う必要があります。  運用開始後に oad のポート番号を変更すると、それ以前に作ったオブジェクトを呼び出すことができなくなります。変更した場合は、システムを再起動してください。	9825
名前サービスが動作しているホスト名 (NameServiceHostName)	名前サーバと oad が動作するホスト名を指定します。	指定なし
リクエスト呼び出しのタイムアウト時間 (RequestTimeout)	メソッド呼び出しの最大待ち時間(0 以上の整数)を設定します。0 を設定すると処理が戻ってくるまで待ち続けます。	30(秒)
クライアント無通信監視タイマ (ClientAutoTimeout)	クライアント側で一定時間送受信要求の無いコネクションを監視するタイマ値(秒)を指定します。指定されたタイマ値の間メッセージの送受信を行わなかったクライアントコネクションは切断されます。既定値は 0 で、この場合、無通信監視は行いません。	0
コネクションの最大接続数 (MaxConnection)	コネクションの最大接続数(0 以上の整数)を設定します。この設定がないとき、および、0 を設定したときは上限を設定しません。	0
corbaloc サーバ呼び出し時の動作 (CorbalocAskWithMT)	初期サービスのアドレスを複数指定した場合の corbaloc サーバ呼び出し時の動作を選択します。 true - マルチスレッドによる同時処理 false - 順次処理(既定値) キャッシュ名前サービスを利用する場合は、true を指定して同時処理を選択してください。	false
インタオペラブル名前サービスで使用するデフォルトのポート番号 (CorbalocDefaultPort)	インタオペラブル名前サービス利用時の URL 記述でポート番号を指定しなかったときに使用するポート番号を指定します。	2809
リファレンス生成時に格納するホスト名 (AlternativeHostName)	リファレンス生成時に格納するホスト名を指定します。 ドメインのホスト名(domain.hostname)に localhost(デフォルト)以外のホスト名が指定された場合、この設定が上書きされます。このため、この設定を変更した後でドメインのホスト名を設定すると値が書き換わってしまいますので注意してください。	指定なし

AlternativeHostNames	IOR に設定されているホスト名(または IP アドレス)がこのプロパティで指定された値に含まれていて、かつ、呼び出し先の CORBA オブジェクトが同一プロセス内で動作していれば、ローカルスタブで呼び出しを実行します。 複数の指定をする場合は、コロン (:) で区切って記述します。	指定なし
MultiConnection	オブジェクトリファレンスが多重化されている場合に、クライアント側でコネクションの多重化を行うかどうかを指定します。 多重化を行う場合は true を指定します。多重化を行わない場合は false を指定します。	true
ConnectionRoundRobin	オブジェクトリファレンスが多重化されている場合に、ラウンドロビンにより呼び出しを各サーバに振り分けるかどうかを指定します。	true
インタフェースリポジトリが動作するホスト名 (InterfaceRepository Host)	インタフェースリポジトリが動作するホスト名を指定します。	指定なし
プールのスレッド数 (PoolThreadNumber)	クライアントからの要求を並行処理(同時処理)するプールのスレッド数(1以上)を指定します。	10
GIOP のマイナー番号 (GIOPMinorVersion)	クライアントがサーバに送信する要求メッセージの GIOP バージョンのマイナー番号には、サーバが IOR の IIOP プロファイルに格納した IIOP バージョンのマイナー番号を使います。GIOPMinorVersion を指定すると、この要求メッセージの GIOP バージョンのマイナー番号を変更することができます。ただし、GIOPMinorVersion で指定した値よりもサーバが IOR の IIOP プロファイルに格納した IIOP バージョンのマイナー番号のほうが小さい場合、サーバが格納した番号が優先されます。  GIOPMinorNumber に設定可能な値を以下に示します。 ・ GIOP1.0 を使用 ・ GIOP1.1 を使用 ・ GIOP1.2 を使用(既定値)	GIOP1.2 を使用
IIOP のマイナー番号 (IIOPMinorVersion)	IIOPMinorVersion には、サーバが IOR の IIOP プロファイルに格納する IIOP バージョンのマイナー番号を指定します。  IIOPMinorNumber に設定可能な値を以下に示します。 ・ IIOP1.0 を使用 ・ IIOP1.1 を使用 ・ IIOP1.2 を使用(既定値)	IIOP1.2 を使用
TCP バックログの指定 (ListenBackLog)	TCP バックログ(接続要求の待ち行列の最大長)を指定します。	0
TCP_NODELAY ソケットオプション(TCPNODELAY)	TCP_NODELAY ソケットオプションの指定を選択します。 true(既定値) - 有効 false - 無効 TCP_NODELAY ソケットオプションが無効な場合、TCP プロトコルのアルゴリズムにより、送信データがある一定量溜まるまで送出手が遅延することがあります。 既定値 true のままであれば、送出手が遅延することはありません。	true
クライアント端末に公開するサーバのホスト名 (ExternalHostName)	負荷分散装置や、クラスタ構成での仮想化のための、クライアント端末に公開する IP アドレスまたはホスト名を指定します。サーバ側で指定すると、全てのクライアント端末で有効になります。この設定は、ローカルマシン内での EJB や CORBA の通信では使用されません。このため、設定されたホスト名が有効になっていなくても、サーバマシン内で EJB や CORBA の通信を行うことができます。 リファレンスを作る際、指定されたアドレスが、AlternativeHostName より優先して格納されます。また、指定されたアドレスと自ホストのアドレスを、内部で自動的に AlternativeHostNames プロパティに設定しています。	指定なし

C++タブ : Object Broker C++の設定を行うことができます。

属性名 (attribute-name)	説明	既定値
CppCodeSetNegotiationStrictCheck	GIOP バージョン 1.0 で通信を行う場合、および、コードセットのネゴシエーション機能を持たない ORB と通信を行う場合に、通信するデータに wchar 型もしくは wstring 型が含まれていたとき、CORBA 仕様に準拠してアプリケーションに例外を返却するかどうかを指定します。 true を指定した場合は、通信するデータに wchar 型もしくは wstring 型が含まれていた場合、アプリケーションに例外を返却します。 false を指定した場合は、wchar 型もしくは wstring 型のコードセットに UCS2L1 を使用して通信します。 既定値は false です。	false
ConnectionRoundRobinTimeout	多重化使用時に個々のサーバへのメソッド呼び出しにかかる最大待ち時間(0以上の整数)を設定します。また0を設定した時は処理が戻ってくるまで待ち続けます。	30
ConnectionWatchInterval	多重化されたオブジェクトのコネクションの状態を監視する間隔時間(0以上の整数)を秒単位で設定します。ただし MultiConnection が有効でない場合は、意味を持ちません。	300
CppConversionCodeSets	string 型のコンバージョンコードセットの指定	JIS_eucJP.ISO8859-1,ISO646 (Windows)  OSF_SJIS1.ISO8859-1,ISO646 (HP-UX, Solaris, Linux)
CppConversionCodeSetSW	wchar/wstring 型のコンバージョンコードセットの指定	UTF16
GetHostNameCache	false を指定すると IP アドレスの解決に毎回 gethostbyname を使います。true を指定したときは同一ホスト名に対して 1 プログラムからは 1 回しか gethostbyname を呼び出しません。	true
LoggingLevel	例外として返せないエラーなどの情報をロギングファイルに出力するときの出力レベルです。ERROR を指定するとエラー情報のみを出力します。WARNING を指定するとエラーとワーニングの情報を出力します。INFORMATION を指定すると全ての情報を出力します。	ERROR
LoggingOutput	ロギングファイル名を指定します。	指定なし
LogLimit	各ログファイルの上限値(キロバイト)を指定します。ログファイルのサイズがこの値を超えると、新しいログファイルが作成されます。古いログファイルは、< ログファイル名_old > として保存されます。	0
LogicalHostName	MC/SG によって多重化されたオブジェクトとして動作させるときの論理ホスト名を指定します。	指定なし
LogicalIPAddress	MC/SG によって多重化されたオブジェクトとして動作させるときの論理 IP アドレス(ドット区切りの文字列)を指定します。	指定なし
NameServiceDuplicationAware	true を指定すると CORBA::ORB::__duplication_aware(1)で名前サーバを動作させます。つまり、MC/SG により多重化された論理ホスト上の名前サーバとして動作します。この指定を行ったときは LogicalHostName および LogicalIPAddress の指定も必要となります。false の場合は物理ホスト上の名前サーバとして動作します。	false
NativeCodeSet	string 型のネイティブコードセットの指定	OSF_SJIS1 (Windows)  JIS_eucJP (HP-UX, Solaris, Linux)
NativeCodeSetW	wchar/wstring 型のネイティブコードセットの指定	UCS2_LEVEL1
MesBufNum	初期メッセージバッファの分割されたメモリ領域個数を指定します。	1
MesBufTryLen	MesBufSize に 32767 以上を指定した場合で、send システムコールで一度に送信要求する大きさを 32767 以上にしたいとき、MesBufTryLen で指定します。MesBufSize を越えて設定することはできません。	2147483647

MesBufSize	メッセージバッファ分割の大きさを指定します。単位はバイトです。	8192
MessagingLog	true を指定すると通信相手、通信量のログをとることができます。	false
MessageBodyLog	true を指定すると通信内容のログをとることができます。	false
MessageLogFile	通信ログ出力先ファイル名を指定します。	指定なし
SendCodeSetContextEverytime	この設定が"true"のときは、CodeSetContext を毎回送ります。設定が"false"のときは、1 コネクションにつき最初の一回だけ送ります。ObjectSpinner R4.x のサーバとコードセットを使用して通信を行うには設定を"true"にする必要があります。	false
ServerKeepAlive	true を指定すると、TCP/IP レベルでの KEEPALIVE をそのマシン内の全サーバプロセスで使用します。	true
ServerRequestTimeout	クライアントからのリクエスト処理にかかる最大待ち時間(0 以上の整数)を秒単位で設定します。0 を設定すると処理が戻ってくるまで待ち続けます。	30
SockBufSize	OS 内部のソケットごとのバッファサイズを指定します。単位はバイトです。未設定時には WebOTX Object Broker C++ は明示的な値の設定を行いません。	0
SysExcWithOMGORG	旧バージョンや他社製品で IDL:omg.org/ で始まるリポジトリ ID を正しく処理できない ORB と通信を行う場合に false を指定します。false を指定すると例外のリポジトリ IDL:omg.org/) を従来形式(IDL:) でマーシャルします。	true
Syslog	true を指定するとシステムログ情報を syslog に出力します。	false
SyslogDetail	true を指定すると詳細なシステムログ情報を syslog に出力します。	false
SyslogForOperator	true を指定するとシステム運用者向けのログ情報を出力します。	false
SyslogForDeveloper	true を指定するとアプリケーション開発者向けのログ情報を出力します。	false
UseCodeSet	この設定が"false"のときは、コードセットを使用せずに通信を行います。"true"のときは、コードセットを使用して通信を行います。	true
UseProfileBodyComponents	IOR Components を ProfileBody 内に格納するか、MultipleComponent 内に格納するかを設定します。この設定が true のときは、ProfileBody 内に格納します。false のときは、MultipleComponent 内に格納します。ただし、IOPMinorVersion が 0 に設定されているときは、この設定にかかわらず、MultipleComponent 内に格納します。	false
CppAddWcharWidthOctet	GIOP バージョン 1.2 において、wchar 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。	true
CppAddWstringWidthOctet	GIOP バージョン 1.2 において、wstring 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。	false
CppClientCertKey	クライアント証明書のファイル名を設定します。 クライアント認証を要求するサーバと通信する場合は必ず設定しなければなりません。 (例) D:\temp\userCA.p12	指定なし
CppSSLProvider	SSL 通信で使用する製品を指定します。  以下の値を設定することができます。 ・SSL-C ・OpenSSL(既定値)	OpenSSL
CppTrustedCACerts	信頼する CA 証明書のファイル名を設定します。 (例) D:\temp\CA.DER	指定なし

Javaタブ : Object Broker Javaの設定を行うことができます。

(注) 以下の項目に無い Object Broker Java のプロパティを、オペレーションから設定可能です。

オペレーションからの設定方法、および設定可能プロパティ一覧は、「運用編(運用操作)」-

「8. Object Broker の運用操作」-「8.1 Object Broker オペレーション一覧」-

「server.objectbrokerconfig」の【3】Object BrokerJava プロパティ設定 を参照してください。

属性名 (attribute-name)	説明	既定値
-------------------------	----	-----

Java エンコーディング (JavaCodeSetEncoding)	<p>通信に使用するコードセットに対応する Java™エンコーディングを変更します。指定は、コードセット名と新たに設定するエンコーディング名をイコール (=) で区切って記述します。複数の指定をする場合は、カンマ (,) で区切って記述してください。</p> <p>-CodeSetEncoding OSF_SJIS1=MS932,0x05020002=MS932</p> <p>コードセット名のかわりにコードセット番号で指定することもできます。10 進数もしくは 16 進数での指定が可能です。16 進数の場合は先頭に 0x を付け加えてください。</p> <p>Object Broker Java でサポートするコードセットと、このプロパティを指定しない場合のエンコーディングの既定値はアプリケーション開発ガイド(CORBA 編)の文字コードセットを参照してください。</p> <p>Object Broker Java で対応していないコードセットに対しては指定できません。指定しても設定は更新されません。また、Java™ がサポートしていないエンコーディングは指定できません。Java™ がサポートするエンコーディングは Java™ のバージョンによって異なります。Java™ がサポートしていないエンコーディングを指定した場合は、文字コード変換が発生した時点 (文字および文字列データの送受信) でアプリケーションに INV_OBJREF 例外を返却します。</p>	この項目を指定しない場合の既定値は、アプリケーション開発ガイド(CORBA 編)の文字コードセットを参照してください。
Char、string 型のコンバージョンコードセット (JavaConversionCodeSets)	<p>char 型、string 型のコンバージョンコードセットを変更します。指定可能なコードセットはアプリケーション開発ガイド(CORBA 編)の文字コードセットを参照してください。</p> <p>ConversionCodeSets プロパティの設定は、サーバに対してのみ有効です。複数のコードセットを指定する場合は、カンマ(,)で区切って記述してください。</p>	この項目を指定しない場合の既定値は、OSF_SJIS1,JIS_eucJP,ISO8859-1,ISO646 です。
wchar、wstring 型のコンバージョンコードセット (JavaConversionCodeSetsW)	<p>wchar 型、wstring 型のコンバージョンコードセットを変更します。指定可能なコードセットはアプリケーション開発ガイド(CORBA 編)の文字コードセットを参照してください。</p> <p>ConversionCodeSetsW プロパティの設定は、サーバに対してのみ有効です。複数のコードセットを指定する場合は、カンマ(,)で区切って記述してください。</p>	この項目を指定しない場合の既定値は、UTF16 です。
コードセットの指定がないときに例外を返却するかどうかの指定 (JavaCodeSetNegotiationStrictCheck)	<p>GIOP バージョン 1.0 で通信を行う場合、および、コードセットのネゴシエーション機能を持たない ORB と通信を行う場合に、通信するデータに wchar 型もしくは wstring 型が含まれていたとき、CORBA 仕様に準拠してアプリケーションに例外を返却するかどうかを指定します。</p> <p>true を指定した場合は、通信するデータに wchar 型もしくは wstring 型が含まれていた場合、アプリケーションに例外を返却します。</p> <p>false を指定した場合は、wchar 型もしくは wstring 型のコードセットに UCS2L1 を使用して通信します。</p>	false
バッファの初期プール数 (JavaInitialBufferSize)	内部バッファが使用するバッファの初期プール数を設定します。内部バッファが使用するバッファは再利用されます。InitialBufferSize で指定された数だけ、ORB 起動時にバッファが確保されます。	4
バッファの初期サイズ (JavaInitialBufferSize)	内部バッファが使用するバッファの初期サイズを設定します。バッファサイズの指定は性能の改善に有効です。既定値(8192 バイト)より大きなデータの送受信を頻繁に行う場合、そのデータサイズより大きいバッファを設定することで、性能を改善することができます。	8192
バッファプールの最大数 (JavaMaxBufferSize)	内部バッファが使用するバッファの最大プール数を設定します。内部バッファが使用するバッファは再利用されます。MaxBufferSize で指定された数以上は、バッファはプールされません。	20
AddWcharWidthOctet (JavaAddWcharWidthOctet)	GIOP バージョン 1.2 において、wchar 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。	true
AddWstringWidthOctet (JavaAddWstringWidthOctet)	GIOP バージョン 1.2 において、wstring 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。	false
クライアント認証の要否 (JavaCertRequest)	クライアントに対する、クライアント証明書の要求の指定を選択します。サーバ側で有効です。 true - クライアント証明書を要求します false - クライアント証明書を要求しません	false
SSL 通信で使用する製品 (JavaSSLProvider)	SSL 通信で使用する製品を指定します。以下の値を設定することができます。	JSSE

	<ul style="list-style-type: none"> <li>・SSL-J</li> <li>・JSSE</li> </ul>	
クライアント証明書のファイル名 (JavaClientCertKey)	クライアント証明書のファイル名を設定します。 クライアント認証を要求するサーバと通信する場合は必ず設定しなければなりません。 (例) D:\temp\userCA.p12	指定なし
クライアント証明書のファイルのパスワード (JavaClientCertKeyPassPhrase)	クライアント証明書のファイルのパスワードを設定します。	指定なし
クライアント証明書の鍵の別名 (JavaClientCertAlias)	クライアント証明書の鍵の別名を設定します。 証明書ファイルが JKS 形式の場合のみ必要です。	指定なし
クライアント証明書の鍵の別名のパスワード (JavaClientCertAliasPassPhrase)	クライアント証明書の鍵の別名のパスワードを設定します。 証明書ファイルが JKS 形式の場合のみ必要です。	指定なし
クライアント証明書の鍵の形式 (JavaClientCertKeyType)	クライアント証明書の鍵の形式を設定します。	この項目を指定しない場合の既定値は DSA です。
クライアント証明書のファイル形式 (JavaClientCertKeyFileType)	クライアント証明書のファイルの形式を設定します。	PKCS12
CA 証明書のファイル名 (JavaTrustedCACerts)	信頼する CA 証明書のファイル名を設定します。 (例) D:\temp\CA.DER	指定なし
CA 証明書のファイルのパスワード (JavaTrustedCACertsPassPhrase)	信頼する CA 証明書のファイルのパスワードを設定します 証明書ファイルが JKS 形式の場合のみ必要です。	指定なし
CA 証明書のファイル形式 (JavaTrustedCACertsFileType)	信頼する CA 証明書のファイルの形式を設定します。DER もしくは JKS を指定することができます。PEM 形式の場合は DER と指定してください。	DER
SSLCipherSuites (JavaSSLCipherSuites)	SSLで利用する CipherSuites の種類を指定します。使用する各 CipherSuites を""で区切って指定します。指定可能な CipherSuites については Sun の JSSE リファレンスガイドを参照ください。	JSSE を使用し、このプロパティを指定しない場合は JSSE の既定値に従います。
スレッドポリシー (JavaThreadPolicy)	サーバ処理のスレッドポリシーを選択します。以下に ThreadPolicy に設定可能な値を示します。 <ul style="list-style-type: none"> <li>・PerClientThread - クライアント単位にスレッドを生成し、そのスレッド上でクライアントからの要求をシリアルに処理します</li> <li>・PooledThread - クライアントからの要求をあらかじめプールされた複数のスレッドに配分してパラレルに処理します</li> </ul> PerClientThread の場合、クライアントが異なれば、スレッドも異なるため、それらの要求はパラレルに処理します。 PooledThread の場合、プールスレッドは、クライアント単位ではなく、クライアント全体で共通のものです。その数は、共通タブの PoolThreadNumber で変更できます。	PerClientThread
Dynamic RMI-IIOP 機能を有効にするかどうかの指定 (UseDynamicStub)	Dynamic RMI-IIOP 機能(RMI-IIOP のダイナミック・プロキシによる通信モード)を有効にするかどうかを指定します。 true - RMI-IIOP のダイナミック・プロキシによる通信モードを有効にします false - RMI-IIOP のダイナミック・プロキシによる通信モードを無効にします	true

Dottedname : server.objectbrokerservice.oad

属性名 (attribute-name)	説明	既定値
OadPort	oad のポート番号(任意の整数)を指定します。 指定可能な値の範囲はOSに依存します。 ORB 通信を行うクライアントとサーバ間で同一のポート番号を使う必要があります。  同一ホス内の OadPort には、(他のサーバも含めて)重複しないポート番号を指定する必要があります。 運用開始後に oad のポート番号を変更すると、それ以前に作ったオブジェクトを呼び出すことができなくなります。変更した場合は、システムを再起動してください。	9825
OadActivationMode	disabled を指定すると一切サーバの自動起動を行いません。 enabled を指定するとサーバの自動起動を行います。	enabled
OadNoCheckActivation	enabled を指定するとあらゆるホストからの活性化を受け付けます。またサーバの実行ファイルについても一切の制限なく受け付けます。このモードの使用はセキュリティ上の問題を引き起こす恐れがあります。 disabled を指定すると、活性化を受け付けるホストのチェックを行います。	disabled
OadAllowRelativePath	true を指定すると、相対パスのサーバ実行ファイルでも自動起動を行います。 false が指定されたときは"相対パスのサーバ実行ファイルは自動起動を行いません。	false
OadActivationTimeout	oad によるサーバプロセス自動起動時に起動完了通知を待つ時間を秒単位で設定します。	10
OadStartup	ドメイン起動時、ObjectBroker MO からのサービススタート時に oad を起動するかどうかを指定します。 OadMO からの oad 起動時は、この値に関係なく、oad を起動します。	true
OadLoggingLevel	oad の LoggingLevel を設定します。設定と引数の両方が指定されている場合は、引数が優先されます。指定可能な値は、LoggingLevel と同じです。未指定時は、LoggingLevel の値が使用されます。	指定なし

Dottedname : server.objectbrokerservice.namesv

属性名 (attribute-name)	説明	既定値
NameServicePort	名前サーバのポート番号。指定可能な値の範囲はOSに依存します。 名前サービス運用開始後にポート番号を変えるとそれ以前に付けられた名前は使えなくなります。 運用開始後にポート番号を変更する場合は、いったん名前サービスを停止後、 \${INSTANCE_ROOT}/config/ObjectBroker/namesv.ndf ファイルを削除してから行ってください。	2809
NameServiceHostName	名前サーバが動作しているホスト名を指定します。	自ホスト
NameServiceExportHostName	名前サーバが動作するホスト上で指定すると、名前サーバの IOR 中に含まれるホスト情報として使われます。NameServiceHostName と同時に指定した場合、この設定が優先されます。	指定なし
NameServiceFileSaveRetryInterval	名前サーバ終了時にパーシステントデータを保存できないとき、データ書き込みの再試行を行う時間を設定します。設定できる上限の値は 2,147,483,647 です。この値を越えたり、負の値を設定してはいけません。	30
NameServiceIdWithOMGORG	旧バージョンや他社製品で IDL:omg.org/ で始まるリポジトリ ID を正しく処理できない ORB と通信を行う場合に false を指定します。false を指定すると名前サービスのリポジトリ ID (IDL:omg.org/) を従来形式 (IDL:) でマーシャルします。	true
NameServiceAccessCheck	true を指定すると \${INSTANCE_ROOT}/config//config/ObjectBroker/namesvwt ファイルに記述されたホストおよび名前サーバが動作しているホスト以外からは resolve 以外の名前サービスへのアクセスができなくなります。namesvwt ファイルには書き込みアクセスを許可するホストのホスト名または IP アドレスを記述します。	false
NameServiceThreadPool	スレッド処理方針を設定します。 指定なし…シングルスレッドで実行されます。 "permethod"…メソッドごとにスレッドを作成して実行されます。 "pool"…あらかじめ作成されたスレッドで実行されます。	指定なし

NameServiceRoundRobin	trueを設定するとラウンドロビン機能が有効になります。falseの場合にラウンドロビンのインタフェースが呼ばれたときの動作は保証しません。	true
NameServiceConvertIOR	変換データファイルのパス または auto を指定します。	指定なし
NameServiceUseCNS	この設定は、名前サーバがキャッシュ名前サーバと連携するかどうかを指定します。この設定を"true"にすると、キャッシュ名前サーバとの連携を受け付けます。キャッシュ名前サーバと連携させる名前サーバが動作しているホストは"true"に指定してください。	false
NameServiceStartup	ドメイン起動時、ObjectBroker MO からのサービススタート時に名前サーバを起動するかどうかを指定します。 NameServiceMO からの名前サーバ起動時は、この値に関係なく、名前サーバを起動します。	true
NameServiceFileSaveRetryCount	名前サーバ終了時にパーシステントデータを保存できないとき、データ書き込みの再試行を行う回数を設定します。設定できる上限の値は2,147,483,647です。この値を越えた値や負の値を設定してはいけません。	2
NameServiceDuplicationAware	trueを指定するとCORBA::ORB::__duplication_aware(1)で名前サーバを動作させます。つまり、MC/SGにより多重化された論理ホスト上の名前サーバとして動作します。 この指定を行ったときはLogicalHostName およびLogicalIPAddressの指定も必要となります。 設定がないとき、および、true以外を以外を指定したときには物理ホスト上の名前サーバとして動作します。	false
NameServiceBackupNDF	NDFファイルのバックアップ機能を利用するかどうかをtrue/falseで指定します。デフォルトはtrueです。	true
NameServiceManagementNDF	壊れているNDFファイルは、調査の為に回避されますが、何世代まで回避しておくかを指定します。デフォルトは2世代です。回避されたNDFファイルは、{OrbRoot}/conf配下に格納されます。	2
NameServiceConfDirectory	namesv.ndfファイルが置かれるディレクトリを指定します。設定がされていないときは\${OrbRoot}/confとなります。	指定なし
NameServiceLoggingLevel	名前サーバのLoggingLevelを設定します。設定と引数の両方が指定されている場合は、引数が優先されます。指定可能な値は、LoggingLevelと同じです。未指定時は、LoggingLevelの値が使用されます。	指定なし

複数の WebOTX Object Broker 利用アプリケーションをインストールしている場合、まれに、NameServiceHostName に他ホストの設定を指定し、かつ、そのホスト上で名前サーバを起動する必要があります。この場合は NameServiceExportHostName に自ホスト名を設定してください。

Dottedname : server.objectbrokerservice.irsv

属性名 (attribute-name)	説明	既定値
InterfaceRepositoryPort	インタフェースリポジトリのポート番号。指定可能な値の範囲はOSに依存します。 インタフェースリポジトリ運用開始後にポート番号を変えるとそれ以前に登録されたインタフェース定義情報は使えなくなります。運用開始後にポート番号を変更する場合は、いったんインタフェースリポジトリを停止後、 \${INSTANCE_ROOT}/config/ObjectBroker/InterfaceRepository.irf ファイルを削除してから行ってください。	9830
InterfaceRepositoryHostName	インタフェースリポジトリが動作しているホスト名を指定します。	自ホスト
InterfaceRepositoryFileSaveRetryInterval	インタフェースリポジトリ終了時にパーシステントデータを保存できないとき、データ書き込みの再試行を行う時間を設定します。設定できる上限の値は2,147,483,647です。この値を越えたり、負の値を設定してはいけません。	30
InterfaceRepositoryIdWithOMGORG	旧バージョンや他社製品でIDL:omg.org/で始まるリポジトリIDを正しく処理できないORBと通信を行う場合にfalseを指定します。指定するとインタフェースリポジトリのリポジトリID(IDL:omg.org/)を従来形式(IDL:)でマーシャルします。	true
InterfaceRepositoryStartup	ドメイン起動時、ObjectBroker MO からのサービススタート時にインタフェースリポジトリを起動するかどうかを指定します。 InterfaceRepositoryMO からのインタフェースリポジトリ起動時は、この値に	false



	関係なく、インタフェースリポジトリを起動します。	
InterfaceRepositoryLoggingLevel	インタフェースリポジトリの LoggingLevel を設定します。設定と引数の両方が指定されている場合は、引数が優先されます。指定可能な値は、LoggingLevel と同じです。未指定時は、LoggingLevel の値が使用されます。	指定なし

Dottedname : server.objectbrokerservice.corbaloc

属性名 (attribute-name)	説明	既定値
CorbalocServerPort	corbaloc サーバのポート番号。指定可能な値の範囲はOSに依存する。	9827
CorbalocDefaultPort	corbaloc URL にポート番号を指定しなかった場合の既定値になります。	2809
CorbalocAskWithMT	この設定は、corbaloc URL に初期サービスのアドレスを複数指定したときの動作を指定します。 この設定を "true" にすると、マルチスレッドによる同時処理をします。 "false" の場合は、左から順次処理が行われます。	false
ToUrlForIiopname	この設定に false を指定したとき、 CosNaming::NamingContextExt::to_url メソッドは、corbaname URL を返します。未設定時は iiopname URL を返します。	true
CorbalocStartup	ドメイン起動時、ObjectBroker MO からのサービススタート時に corbaloc サーバを起動するかどうかを指定します。 CorbalocMO からの corbaloc サーバ起動時は、この値に関係なく、corbaloc サーバを起動します。	false
CorbalocLoggingLevel	corbaloc サーバの LoggingLevel を設定します。設定と引数の両方が指定されている場合は、引数が優先されます。指定可能な値は、LoggingLevel と同じです。未指定時は、LoggingLevel の値が使用されます。	指定なし

Dottedname : server.objectbrokerservice.cnamesv

属性名 (attribute-name)	説明	既定値
CacheNameServicePort	キャッシュ名前サーバが使用するポート番号を 10 進数で指定します。指定可能な値の範囲はOSに依存します。 未設定もしくは 0 を指定したときは自動的に割り当てられます。	9829
CacheNameServiceThreadPolicy	スレッド処理方針を設定します。 指定無し…シングルスレッドで実行されます。 permethod…メソッドごとにスレッドを作成して実行されます。 pool…あらかじめ作成されたスレッドで実行されます。  キャッシュ名前サーバが動作するホストではこの設定を pool にしてください。また、これにともない PoolThreadNumber も設定します。	指定なし
CacheSynchroInterval	キャッシュ名前サーバと連携している名前サーバが動作しているか確認する間隔 (秒) を指定します。 名前サーバのダウンを検出した場合、その名前サーバから収集したキャッシュは削除されます。 また、ダウンしていた名前サーバが再起動していた場合は、キャッシュ名前サーバ上にあるキャッシュ (これは別の名前サーバから収集したものです) に相当するオブジェクトを名前サーバから収集し、キャッシュに加えます。名前サーバの生死状態に変化がない場合は何も行いません。	300
nmsvlist	nmsvlist はキャッシュ元となる名前サーバのリストファイルです。 キャッシュ名前サーバを動作させるマシンの \${INSTANCE_ROOT}/config/ObjectBroker ディレクトリに格納されます。 nmsvlist には名前サーバ 1 つにつき 1 行ずつ、ホスト名、IOR 文字列、corbaloc: もしくは corbaname: の形で記述します。 複数の名前サーバを設定する場合は、「,」(カンマ) で区切り、指定して下さい。	指定なし
CacheNameServiceStartup	ドメイン起動時、ObjectBroker MO からのサービススタート時にキャッシュ名前サーバを起動するか指定します。 CacheNameServiceMO からのキャッシュ名前サーバ起動時は、この値に関係なく起動します。	false

FullSynchroInterval	FullSynchroInterval を指定すると、名前サーバの生死に関らず、指定した間隔で、キャッシュの更新を行います。 値は CacheSynchroInterval の n 倍というように指定します。CacheSynchroInterval が 300(秒)のとき、FullSynchroInterval に 4 を指定すると、1200 秒(20 分)毎に、キャッシュの更新を行います。  (注意) キャッシュの更新は、キャッシュ名前サーバおよび名前サーバに大きな負荷が掛かります。名前サーバに沢山のオブジェクトが登録されている場合は、CacheSynchroInterval × FullSynchroInterval で計算される時間が小さくならないようにして下さい。 未設定時は 0 で、FullSynchroInterval 間隔でのキャッシュ更新は行いません。	0
CNSUpdatePolicy	名前サーバが更新されたとき、関連する CNS に、どのように情報を通知するかを決定します。 "all"を指定すると、オブジェクト登録時および削除時に情報を通知します。未設定時は削除時のみの通知となります。 この設定を有効にする場合、名前サーバ側、キャッシュ名前サーバ側の双方で "all"を指定して下さい。それ以外の組み合わせでは意味を持ちません。 例えば、名前サーバ側が "all" で、キャッシュ名前サーバ側が "all" でない場合、名前サーバは全ての情報を通知しますが、キャッシュ名前サーバは削除情報のみを受け入れます。これとは逆の設定の場合、キャッシュ名前サーバは全ての通知を受け入れますが、名前サーバが削除情報しか通知しないため、結果として、削除情報のみしか受け取れません。	指定なし
CacheNameServiceLoggingLevel	CNS の LoggingLevel を設定します。設定と引数の両方が指定されている場合は、引数が優先されます。 引数で設定する場合は、「-ORBLoggingLevel」を指定します。指定可能な値は、LoggingLevel と同じです。未指定時は、設定の LoggingLevel の値になります。	指定なし

Dottedname : server.objectbrokerservice.oadj

oadj の設定項目は、[一般] のタブにあります。

(注) 属性重要度をデフォルトの 1 から 3 に上げると、[拡張]タブが表示され、設定可能な項目が増えますが、[拡張タブ]の項目は、通常、設定の必要のないものです。

一般タブ : OadJに関する設定を行うことができます。

属性名 (attribute-name)	説明	既定値
OadJ を起動するかどうかの指定(Startup)	ドメイン起動時および ObjectBroker MO からのサービススタート時に OadJ を起動するかどうかを指定します。OadJ MO からの OadJ 起動時は、この値に関係なく、OadJ を起動します。	true
ライセンス情報を取得できるかどうかを表す値(Licensed)	インストール時にライセンスが登録されたかどうかを表します。ただし、OadJ が起動していない場合は必ず false が返ります。	参照のみ可能な項目です。既定値はありません。
OadJ のポート番号(Port)	OadJ の使用するポート番号です。	9826
Oad のポート番号(OadPort)	Oad のポート番号です。 OadJ は起動時に Oad を使用します。Oad のポート番号の変更は、oad MO から行うことができます。	参照のみ可能な項目です。既定値は 9825 です。
OadJ 起動時のログ採取有無(InitLog)	OadJ の起動に関するログを採取するかどうかを指定します。ログは \${INSTANCE_ROOT}/logs/webotx_ospi.log に出力されます。oadj プロセスの起動に失敗するような問題の解決に有効です。	true
OadJ ログファイルサイズ(LogFileSize)	oadj のログファイルの上限値(キロバイト)を指定します。ログファイルのサイズがこの値を超えると、新しいログファイルが作成されます。古いログファイルは、<ログファイル名.bak>として保存されます。 (注)この値は目安であり、ログファイルサイズが設定値を上回る場合があります。	100
OadJ のログレベル	ログレベルを切り替えることにより、OadJ の運用時のログをとることができます。ロ	ERROR

(LogLevel)	<p>ログは<code>{INSTANCE_ROOT}/logs/ObjectBroker/oadj.log</code> に出力されます。ログレベルには以下のものがあります。</p> <p>NO - 診断メッセージを出力しません。</p> <p>ERROR - 通信エラーなど、エラーが発生した時に診断メッセージを出力します。</p> <p>WARN - 警告的な意味を持つ状況が発生した時に診断メッセージを出力します。</p> <p>INFO - アプリケーションプログラムの動作情報を出力します。</p> <p>DEBUG - 詳細な診断情報を出力します。自動起動サーバプロセスが起動されないような問題の解決に有効です。</p>	
JavaCommand	<p>OadJ が使用する Java コマンドをフルパスで設定します。</p> <p>例)</p> <p><code>D:¥j2sdk1.4.2_12¥bin¥java</code></p>	<p>未指定時</p> <p><code>System.getProperty("java.home")</code> で取得した JDK の Java コマンドを使用します。</p>

拡張タブ: OadJに関する拡張設定を行うことができます。通常はこのタブの項目の設定を行う必要はありません。

項目名(属性名 (attribute-name))	説明	既定値
OadJ 起動時 Java オプション (JavaOptions)	<p>OadJ 起動時に呼び出される <code>java/jre</code> 起動コマンドに渡すコマンドライン オプションを指定します。複数のオプションが指定できます。通常は指定する必要はありませんが、何らかの <code>java/jre</code> 実行時の問題を回避するような場合に有効となります。オプションについては、JDK/JRE のドキュメントを参考にしてください。未指定時は何も渡されません。</p> <p>例)メモリ割り当てプールの初期サイズと最大サイズを指定する場合。</p> <p><code>OadJJavaOptions=-Xms64m -Xmx128m</code></p>	<p>指定なし(未指定時は何も渡されません。)</p>
OadJ のクラスパス (Classpath)	<p>OadJ 起動時のクラスパスを指定します。</p> <p>例)</p> <p><code>D:¥libtemp¥aaa.jar;D:¥libtemp¥bbb.jar;D:¥libtemp¥cccc.jar</code></p>	<p>この表の下にある &lt;“OadJ のクラスパス”の既定値&gt;を参照してください。</p>
AliveCheckSleepTime	<p>oadj により起動されたサーバプロセスの生存を確認する間隔をミリ秒単位で設定します。</p>	<p>30000(ミリ秒)</p>
ServerActivationTimeout	<p>OadJによるサーバプロセス自動起動時に起動完了通知を待つ時間をミリ秒単位で設定します。</p>	<p>30000(ミリ秒)</p>
OadJAvoidAliveCheckProblem	<p>サーバプロセス生存確認時に発生する <code>IOException</code> の種類が JDK のバージョンによって異なることを吸収します。</p>	<p>false</p>

<“OadJ のクラスパス(Classpath)”の既定値>

Windows の場合 :

`<webotx_home>¥ObjectBroker¥lib¥ospiorb50.jar;<webotx_home>¥ObjectBroker¥lib¥ospiname50.jar;<webotx_home>¥ObjectBroker¥lib¥ospioadj50.jar;<webotx_home>¥ObjectBroker¥lib¥jssocks.jar`

UNIX の場合 :

`/usr/lib/ObjectSpinner/lib/ospiorb50.jar;/usr/lib/ObjectSpinner/lib/ospiname50.jar;/usr/lib/ObjectSpinner/lib/ospioadj50.jar;/usr/lib/ObjectSpinner/lib/jssocks.jar`

Dottedname : `server.objectbrokerservice.ospprxy`

ospprxy の設定項目は、[一般] のタブにあります。

一般タブ: ospprxyに関する設定を行うことができます。

属性名 (attribute-name)	説明	既定値
ospprxy を起動するかどうかの指定(Startup)	<p>ドメイン起動時および ObjectBroker MO からのサービススタート時に ospprxy を起動するかどうかを指定します。ospprxy MO からの ospprxy 起動時は、この値に関係なく、ospprxy を起動します。</p>	<p>false</p>

## 11.2.WebOTX Object Broker Java™ におけるORBのプロパティ定義

この章では ORB に設定できるプロパティについて説明します。

### 項目の参照と設定

項目の参照と設定は、Java のシステムプロパティと同様に行ってください。

詳細は、4.2.エージェント設定項目一覧 の Dottedname : server.java-config、属性名「jvm-options」の説明を参照してください。

各プロパティで、プレフィックス「jp.co.nec.orb.」が付く名称(例:jp.co.nec.orb.AddWStringWidthOctet)のものは、プレフィックスをつけない名称でも設定可能です。

WebOTX Ver6 で既定値を変更したプロパティの一覧を、以下に示します。

プロパティ名(括弧()内の名前でも指定可能 )

jp.co.nec.orb.AddWStringWidthOctet

説明	Ver5 までの既定値	Ver6 での既定値
GIOP バージョン 1.2 において、wchar 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。	true	false

jp.co.nec.orb.ClientRequestTimeout (jp.co.nec.orb.RequestTimeout)

説明	Ver5 までの既定値	Ver6 での既定値
クライアントからサーバにオペレーション呼び出しを行ってサーバからその返事が返ってくるまでのタイムアウト値を設定します。このプロパティを指定したプロセス全体にこのタイムアウト値が適用されます。単位は秒です。0 を設定するとタイムアウトせず、サーバからリプライが返るまで待ちつづけます。 設定した時間を過ぎてもサーバからリプライがない場合には org.omg.CORBA.NO_RESPONSE 例外 (マイナーコード 5130) が発生します。	タイムアウトしない	30 秒

jp.co.nec.orb.UseConnectionRoundRobin (jp.co.nec.orb.ConnectionRoundRobin)

説明	Ver5 までの既定値	Ver6 での既定値
オブジェクトリファレンスが多重化されている場合に、コネクションのラウンドロビンを行うかどうかを指定します。 ラウンドロビンを行う場合は true を指定します。ラウンドロビンを行わない場合は false を指定します。	false	true

jp.co.nec.orb.ConnectionRoundRobinTimeout

説明	Ver5 までの既定値	Ver6 での既定値
オブジェクトリファレンスが多重化されている場合に、各オブジェクトに対する呼び出しのタイムアウト時間を秒単位で指定します。0 を指定した場合はタイムアウトしません。	タイムアウトしない	ClientRequestTimeout の設定に従う

jp.co.nec.orb.LogLevel (jp.co.nec.orb.LoggingLevel)

説明	Ver5 までの既定値	Ver6 での既定値
<p>WebOTX Object Broker Java™ は、動作中に診断メッセージを表示することができます。診断メッセージを表示すると、アプリケーションのデバッグなどに便利なこともあります。診断メッセージは設定されている LogLevel に基づいて出力する情報が変わります。以下に LogLevel に設定可能なレベルを示します。</p> <ul style="list-style-type: none"> <li>•NO - 診断メッセージを表示しません</li> <li>•ERROR - 通信エラーなど、エラーが発生した時に診断メッセージを表示します</li> <li>•WARN - 警告的な意味を持つ状況が発生した時に診断メッセージを表示します</li> <li>•INFO - アプリケーションプログラムの動作情報を表示します</li> <li>•DEBUG - 詳細な診断情報を表示します</li> </ul> <p>LogLevel は上記のリストの下の方を指定するほど表示される情報は多くなります。</p>	ERROR	NO(出力しない)

jp.co.nec.orb.SSLProvider

説明	Ver5 までの既定値	Ver6 での既定値
<p>SSL 通信で使用する製品を指定します。以下の値を設定することができます。</p> <ul style="list-style-type: none"> <li>•SSL-J</li> <li>•JSSE</li> </ul>	SSL-J	JSSE

WebOTX Object Broker Java™ における ORB のプロパティ一覧を、以下に示します。

- プロパティ名(括弧()内の名前でも指定可能)

jp.co.nec.orb.AddWcharWidthOctet

説明	既定値
<p>GIOP バージョン 1.2 において、wchar 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。指定しない場合の既定値は true です。</p>	true

jp.co.nec.orb.AddWstringWidthOctet

説明	既定値
<p>GIOP バージョン 1.2 において、wchar 型のデータをマーシャルするときに、1 文字あたりのバイト数を示すオクテットを各文字に付加するかどうかを指定します。付加する場合は true を指定します。付加しない場合は false を指定します。</p>	false

jp.co.nec.orb.AllowNullString

説明	既定値
<p>オペレーション呼び出しにおいて、string 型および wstring 型の引数、戻り値に null を指定することを許すかどうかを指定します。許す場合は true を指定します。許さない場合は false を指定します。</p>	false

jp.co.nec.orb.AlternativeHostName

説明	既定値
<p>リファレンス生成時に格納するホスト名を指定するときに使用します。 サーバソケットの bind では全てのローカルアドレスへの接続を受け入れます。</p>	java.net.InetAddress.getLocalHost() の戻り値

jp.co.nec.orb.AlternativeHostNames

説明	既定値
<p>IOR に設定されているホスト名(または IP アドレス)がこのプロパティで指定された値に含まれていて、かつ、呼び出し先の CORBA オブジェクトが同一プロセス内で動作していれば、ローカルスタブで呼び出します。複数の指定をする場合は、コロン(:)で区切って記述します。</p>	指定しない

jp.co.nec.orb.AppletProxyHost

説明	既定値
AppletProxy のホスト名	指定なし

jp.co.nec.orb.AppletProxyPort

説明	既定値
AppletProxy のポート番号	-1

jp.co.nec.orb.CertRequest

説明	既定値
クライアントに対して、クライアント証明書の要求を行うかどうかを指定します。サーバ側で有効です。 ・true - クライアント証明書を要求します ・false - クライアント証明書を要求しません	false

jp.co.nec.orb.ClientAutoTimeout

説明	既定値
クライアント側で一定時間送受信要求の無いコネクションを監視するタイム値(秒)を指定します。指定されたタイム値の間メッセージの送受信を行わなかったコネクションは切断されます。指定されない場合、監視は行いません。	監視しない

jp.co.nec.orb.ClientCertAlias

説明	既定値
JSSE で使用するクライアント証明書の鍵の別名を設定します。証明書ファイルが JKS 形式の場合のみ必要です。	指定しない

jp.co.nec.orb.ClientCertAliasPassPhrase

説明	既定値
JSSE で使用するクライアント証明書の鍵の別名のパスワードを設定します。証明書ファイルが JKS 形式の場合のみ必要です。	指定しない

jp.co.nec.orb.ClientCertKey

説明	既定値
JSSE で使用するクライアント証明書のファイル名を設定します。 クライアント認証を要求するサーバと通信する場合は必ず設定しなければなりません。	指定しない

jp.co.nec.orb.ClientCertKeyFileType

説明	既定値
JSSE で使用するクライアント証明書のファイルの形式を設定します。 PKCS12 もしくは JKS を指定することができます。	PKCS12

jp.co.nec.orb.ClientCertKeyPassPhrase

説明	既定値
JSSE で使用するクライアント証明書のファイルのパスワードを設定します。	指定しない

jp.co.nec.orb.ClientCertKeyType

説明	既定値
JSSE で使用するクライアント証明書の鍵の形式を設定します。このプロパティを指定しない場合は DSA が既定値として選択されます。	DSA

jp.co.nec.orb.ClientRequestTimeout (jp.co.nec.orb.RequestTimeout)

説明	既定値
クライアントからサーバにオペレーション呼び出しを行ってサーバからその返事が返ってくるまでのタイムアウト値(単位:秒)を設定します。このプロパティを指定したプロセス全体にこのタイムアウト値が適用されます。 0 を設定するとタイムアウトせず、サーバからリプライが返るまで待ちつづけます。設定した時間を過ぎてもサーバからリプライがない場合には org.omg.CORBA.NO_RESPONSE 例外 (マイナーコード 5130) が発生します。	30 秒

jp.co.nec.orb.CodeBasePropertyFile

説明	既定値
プロセスで共通のコードベースプロパティファイルのファイル名を指定します。このプロパティを指定しない場合の既定値は カレントディレクトリ/codebase.conf です。	./codebase.conf

jp.co.nec.orb.CodeBasePropertyFileDir

説明	既定値
コードベースプロパティ名に対応するコードベースプロパティファイルの格納ディレクトリを指定します。	カレントディレクトリ

jp.co.nec.orb.CodeBasePropertyFileSuffix

説明	既定値
コードベースプロパティ名に対応するコードベースプロパティファイルの拡張子を指定します。	cbp

jp.co.nec.orb.CodeSetEncoding

説明	既定値
<p>通信に使用するコードセットに対応する Java™ エンコーディングを変更します。</p> <p>指定は、コードセット名と新たに設定するエンコーディング名をイコール (=) で区切って記述します。複数の指定をする場合は、カンマ (,) で区切って記述してください。</p> <p>-CodeSetEncoding OSF_SJIS1=MS932,0x05020002=MS932</p> <p>コードセット名のかわりにコードセット番号で指定することもできます。10 進数もしくは 16 進数での指定が可能です。16 進数の場合は先頭に 0x を付け加えてください。</p> <p>WebOTX Object Broker Java™ で対応していないコードセットに対しては指定できません。指定しても設定は更新されません。Java™ がサポートしていないエンコーディングは指定できません。Java™ がサポートするエンコーディングは Java™ のバージョンによって異なります。Java™ がサポートしていないエンコーディングを指定した場合は、文字コード変換が発生した時点 (文字および文字列データの送受信) でアプリケーションに INV_OBJREF 例外を返却します。</p>	<p>「アプリケーション開発ガイド 4.4 章」-</p> <p>「 4.15.6.Object Broker Java の機能」</p> <p>-「文字コードセット」を参照してください。</p>

jp.co.nec.orb.CodeSetNegotiationStrictCheck

説明	既定値
<p>GIOP バージョン 1.0 で通信を行う場合、および、コードセットのネゴシエーション機能を持たない ORB と通信を行う場合に、通信データに wchar 型もしくは wstring 型が含まれていたとき、CORBA 仕様準拠してアプリケーションに例外を返却するかどうかを指定します。指定できる値は true もしくは false です。true を指定した場合は、通信するデータに wchar 型もしくは wstring 型が含まれていた場合、アプリケーションに例外を返却します。false を指定した場合、wchar 型もしくは wstring 型のコードセットとして UCS2L1 を使用します。</p>	false

jp.co.nec.orb.CompatibleMinorCode

説明	既定値
R7.10 より Object Broker Java™独自のマイナーコードの番号の一部を CORBA 標準マイナーコードに変更しました。そのため、R6.x もしくはそれ以前で作成したアプリケーションでマイナーコードで処理を切り分けている場合は、正しく動作しなくなる可能性があります。その場合、このプロパティに 6 を指定してください。	8

jp.co.nec.orb.UseConnectionRoundRobin(jp.co.nec.orb.ConnectionRoundRobin)

説明	既定値
オブジェクトリファレンスが多重化されている場合にコネクションのラウンドロビンを行うかどうかを指定します。ラウンドロビンを行う場合は true を指定します。ラウンドロビンを行わない場合は false を指定します。	true

jp.co.nec.orb.ConnectionRoundRobinTimeout

説明	既定値
オブジェクトリファレンスが多重化されている場合に、それぞれのオブジェクトに対する呼び出しのタイムアウト時間を秒単位で指定します。0 を指定した場合はタイムアウトしません。	ClientRequestTimeout の設定に従う

jp.co.nec.orb.ConnectionSharingUnit

説明	既定値
<p>クライアント側でコネクションを生成、管理するときの共有単位を設定します。</p> <p>WebOTX Object Broker Java™ では設定された共有単位毎にコネクションを生成、管理します。</p> <p>プロセス (JavaVM) 単位、ORB インスタンス単位、Applet インスタンス単位のいずれかの設定が可能です。</p> <p>ConnectionSharingUnit は ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます</p> <ul style="list-style-type: none"><li>•Process - プロセス (JavaVM) 単位</li><li>•ORB - ORB.init() で生成された ORB インスタンス単位</li><li>•Applet - ORB.init() の引数で指定された Applet インスタンス単位</li></ul> <p>クライアントコネクションの切断、解放は ORB インスタンスの shutdown()、もしくは destroy() のいずれかが呼び出されたときに行います。切断、解放の対象となるコネクションは ConnectionSharingUnit に設定された共有単位になります。</p> <p>共有単位が同じである全ての ORB インスタンスの shutdown()、destroy() が呼び出されない限り切断、解放は行いませんので注意が必要です。例えば、Process を設定して生成した ORB インスタンスが一つのプロセス内に複数存在する場合、それら全ての ORB インスタンスの shutdown()、destroy() が呼び出されない限り切断、解放は行いません。</p>	Process

jp.co.nec.orb.ConversionCodeSets

説明	既定値
<p>char 型、string 型のコンバージョンコードセットを変更します。指定可能なコードセットは「アプリケーション開発ガイド」-「5.2.7.Object Broker Java の機能」-「文字コードセット」を参照してください。</p> <p>ConversionCodeSets プロパティの設定は、サーバに対してのみ有効です。複数のコードセットを指定する場合は、カンマ (,) で区切って記述してください。</p>	OSF_SJIS1, JIS_eucJP, ISO8859-1, ISO646

jp.co.nec.orb.ConversionCodeSetsW

説明	既定値
<p>wchar 型、wstring 型のコンバージョンコードセットを変更します。指定可能なコードセットは「アプリケーション開発ガイド」-「5.2.7.Object Broker Java の機能」-「文字コードセット」を参照してください。</p> <p>ConversionCodeSetsW プロパティの設定は、サーバに対してのみ有効です。複数のコードセットを指定する場合は、カンマ (,) で区切って記述してください。</p>	UTF16

jp.co.nec.orb.CorballocAskWithMT

説明	既定値
<p>初期サービスのアドレスを複数指定した場合の corballoc サーバ呼び出し時の動作を選択します。</p> <ul style="list-style-type: none"><li>•true - マルチスレッドによる同時処理</li><li>•false - 順次処理</li></ul> <p>キャッシュ名前サービスを利用する場合は、true を指定して同時処理を選択してください。</p> <p>CorballocAskWithMT は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。</p>	false

jp.co.nec.orb.CorballocDefaultPort

説明	既定値
<p>名前サービス利用時の URL 記述 (corballoc URL) でポート番号を指定しなかったときに使用するポート番号を指定します。</p> <p>CorballocDefaultPort は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。</p>	2809



jp.co.nec.orb.DomainName

説明	既定値
<p>サーバホストのドメイン名を指定します。DomainNameは、ドメイン名の付いたホスト(例:host1.domain1.nec.co.jp)と付かないホスト(例:host2)が混在している場合に、ホスト名の整合性を取るために使用します。</p> <p>WebOTX Object Broker Java™ がクライアントとして動作する場合、ホスト名解決に失敗すると、DomainNameで指定されたドメイン名を補完または削除してリトライします。サーバとして動作する場合、サーバのホスト名にドメイン名が付いていなければ、DomainNameで指定されたドメイン名を付加した名前オブジェクトリファレンスを作成します。</p>	指定しない

jp.co.nec.orb.ExternalHostName

説明	既定値
<p>負荷分散装置や、クラスタ構成での仮想化のための、クライアント端末に公開するIPアドレスまたはホスト名を指定します。サーバ側で指定すると、全てのクライアント端末で有効になります。この設定は、ローカルマシン内でのEJBやCORBAの通信では使用されません。このため、設定されたホスト名が有効になっていなくても、サーバマシン内でEJBやCORBAの通信を行うことができます。</p> <p>リファレンスを作る際、指定されたアドレスが、AlternativeHostNameより優先して格納されます。また、指定されたアドレスと自ホストのアドレスを、内部で自動的にAlternativeHostNamesプロパティに設定しています。</p>	指定しない

jp.co.nec.orb.GIOPMinorNumber(jp.co.nec.orb.GIOPMinorVersion)

説明	既定値
<p>クライアントがサーバに送信する要求メッセージのGIOPバージョンのマイナー番号には、サーバがIORのIIOPプロファイルに格納したIIOPバージョンのマイナー番号を使います。GIOPMinorNumberを指定すると、この要求メッセージのGIOPバージョンのマイナー番号を変更することができます。ただし、GIOPMinorNumberで指定した値よりもサーバがIORのIIOPプロファイルに格納したIIOPバージョンのマイナー番号のほうが小さい場合、サーバが格納した番号が優先されます。</p> <p>GIOPMinorNumberに設定可能な値を以下に示します。</p> <ul style="list-style-type: none"> <li>・0 - GIOP1.0を使用</li> <li>・1 - GIOP1.1を使用</li> <li>・2 - GIOP1.2を使用</li> </ul>	2 - GIOP1.2を使用

jp.co.nec.orb.IIOPCompatible

説明	既定値
<p>IIOPに準拠した通信を行うかどうかを指定します。</p> <ul style="list-style-type: none"> <li>・true - IIOPに準拠した通信を行う</li> <li>・false - IIOPに準拠した通信を行わず、拡張プロトコルによる通信を行う</li> </ul>	false

jp.co.nec.orb.IIOPMinorNumber(jp.co.nec.orb.IIOPMinorVersion)

説明	既定値
<p>サーバがIORのIIOPプロファイルに格納するIIOPバージョンのマイナー番号を指定します。IIOPMinorNumberに設定可能な値を以下に示します。</p> <ul style="list-style-type: none"> <li>・0 - IIOP1.0を使用</li> <li>・1 - IIOP1.1を使用</li> <li>・2 - IIOP1.2を使用</li> </ul>	2 - IIOP1.2を使用

jp.co.nec.orb.ImplementationName

説明	既定値
<p>自動起動サーバのインプリメンテーション名を設定します。自動起動サーバのオブジェクトリファレンスを生成する場合に使用します。「インプリメンテーションの登録」オペレーション(instimpl)による自動起動サーバの登録時に指定するインプリメンテーション名と同じ名前を指定してください。</p>	指定しない

jp.co.nec.orb.InitialBufferPoolSize

説明	既定値
<p>ストリームが使用するバッファの初期プール数を設定します。ストリームが使用するバッファは再利用されます。InitialBufferPoolSizeで指定された数だけ、ORB起動時にバッファが確保されます。</p>	4

jp.co.nec.orb.InitialBufferSize

説明	既定値
ストリームが使用するバッファの初期サイズを設定します。バッファサイズの指定は性能の改善に有効です。既定値 (8192 バイト) より大きなデータの送受信が頻繁におこるような場合、そのデータサイズより大きいバッファを設定しておけば、バッファの拡張/破棄等の無駄な処理を避けることができます。	8192

jp.co.nec.orb.InterfaceRepositoryHost

説明	既定値
インタフェースリポジトリが動作するホスト名を指定することができます。このプロパティが設定されているとき、この設定は ORBInitialHost より優先されます。 インタフェースリポジトリについて複数の設定が同時に行われた場合、優先されるプロパティの順番を下記に示します。 1. jp.co.nec.orb.InterfaceRepositoryIOR 2. jp.co.nec.orb.InterfaceRepositoryHost 3. org.omg.CORBA.ORBInitialHost  InterfaceRepositoryHost は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。	指定しない

jp.co.nec.orb.InterfaceRepositoryIOR

説明	既定値
インタフェースリポジトリを表す文字列形式のオブジェクトリファレンスを指定することができます。指定するオブジェクトリファレンスは下記のような文字列です。 IOR:010000000d0... InterfaceRepositoryIOR は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。	指定しない

jp.co.nec.orb.ListenBackLog

説明	既定値
TCP のバックログ (接続要求の待ち行列の最大長) を指定するときに使用します。	Java VM の実装依存

jp.co.nec.orb.LogFileName (jp.co.nec.orb.LoggingOutput)

説明	既定値
診断メッセージや通信データの出力先ファイル名を、フルパス指定、または相対パス指定で設定することができます。既に指定ファイルが存在している場合は、append 出力します。指定ファイルに書き込めない場合は、BAD_PARAM 例外が返ります。	指定しない

jp.co.nec.orb.LogFileSize (jp.co.nec.orb.LogLimit)

説明	既定値
LogFileName で指定したファイルのサイズ上限値を単位: キロバイトで設定することができます。-1 (既定値) 指定時は無制限となります。ファイルのサイズが指定値を越えたら、カレントファイルを "ファイル名.bak" として保存します。LogFileName を指定した場合にのみ有効です。指定サイズが不正な場合は、BAD_PARAM 例外が返ります。	無制限

jp.co.nec.orb.LogLevel (jp.co.nec.orb.LoggingLevel)

説明	既定値
WebOTX Object Broker Java™ は、動作中に診断メッセージを表示することができます。診断メッセージを表示すると、アプリケーションのデバッグなどに便利なこともあります。診断メッセージは設定されている LogLevel に基づいて出力する情報が変わります。以下に LogLevel に設定可能なレベルを示します。 ・NO - 診断メッセージを表示しません ・ERROR - 通信エラーなど、エラーが起ったときに診断メッセージを表示します ・WARN - 警告的な意味を持つ状況が起ったときに診断メッセージを表示します ・INFO - アプリケーションプログラムの動作情報を表示します ・DEBUG - 詳細な診断情報を表示します LogLevel は上記のリストの下の方を指定するほど、表示される情報は多くなります。	NO (出力しない)

jp.co.nec.orb.MaxAcceptNumber

説明	既定値
サーバにおける同時に接続可能なクライアントの数を指定します。MaxAcceptNumber で指定された数以上の接続要求はコネクション切断となります。 システムの設定値によっては、プロセスあたりで利用できるファイル数の制限のために accept に失敗し以降の要求を受け付けられない現象が発生します。このような場合にその数に到達しない範囲で指定すると有効です。未指定時は無制限となります。	無制限

jp.co.nec.orb.MaxBufferPoolSize

説明	既定値
ストリームが使用するバッファの最大プール数を設定します。ストリームが使用するバッファは再利用されません。MaxBufferPoolSize で指定された数以上はバッファはプールされません	20

jp.co.nec.orb.MaxConnectionNumber (jp.co.nec.orb.MaxConnection)

説明	既定値
コネクションの最大接続数(0以上の整数)を設定します。この設定がないとき、および、0を設定したときは上限を設定しません。	0(上限なし)

jp.co.nec.orb.MessageDumpLevel (jp.co.nec.orb.MessageBodyLog)

説明	既定値
WebOTX Object Broker Java™ は、動作中に通信データを採取することができます。通信データを採取すると、障害解析などに便利なこともあります。採取データは設定されている MessageDumpLevel に基づいて出力する情報が変わります。以下に MessageDumpLevel に設定可能なレベルを示します。 ・NO - 通信データを採取しません ・HEAD - 通信データのヘッダのみ採取します ・ALL - 通信データ全てを採取します	NO

jp.co.nec.orb.MessageDumpSize

説明	既定値
通信データ採取サイズの上限値を単位:バイトで設定することができます。-1(既定値)指定時は無制限となります。採取する通信データに対して、指定上限値を越えるデータは出力しません。指定サイズが不正な場合は、BAD_PARAM 例外が返ります。	無制限

jp.co.nec.orb.NameServiceHost (jp.co.nec.orb.NameServiceHostName)

説明	既定値
名前サーバが動作するホスト名を指定することができます。このプロパティが設定されているとき、この設定は ORBInitialHost より優先されます。 名前サーバについて複数の設定が同時に行われた場合、優先されるプロパティの順番を下記に示します。 1. jp.co.nec.orb.NameServiceIOR 2. jp.co.nec.orb.NameServiceHost 3. org.omg.CORBA.ORBInitialHost  NameServiceHost は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。	指定しない

jp.co.nec.orb.NameServiceIOR

説明	既定値
名前サービスのルートコンテキストを表す文字列形式のオブジェクトリファレンスを指定することができます。指定するオブジェクトリファレンスは下記のような文字列です。 IOR:01000000d0... NameServiceIOR は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。	指定しない

jp.co.nec.orb.ORBDefaultInitRef

説明	既定値
初期サービスのオブジェクトを以下の形式で指定します。 -ORBInitRef <ObjectID>=<ObjectURL> プログラムが初期サービスのオブジェクトリファレンスを取得するメソッド org.omg.CORBA.ORB.resolve_initial_references を使用していたときに有効です。この指定がされると、org.omg.CORBA.ORB.resolve_initial_references メソッドは、ObjectID が渡されたときに ObjectURL が表わすオブジェクトリファレンスを返します。 ORBInitRef は、ORB.init () を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。	指定しない

jp.co.nec.orb.ORBSubNetMask

説明	既定値
org.omg.CORBA.ORB.resolve_initial_references が呼び出されたとき、WebOTX Object Broker Java はサービスを提供しているホストを調べます。このとき、サービスを提供しているホストが指定されていない場合は、ブロードキャストを行ってホストを探し出そうとします。ブロードキャストを行うと、要求されているサービスが存在しているホストの OAD が返事を返します。このときのブロードキャストするためのサブネットマスクを ORBSubNetMask に指定します。ORBSubNetMask の設定値は、必ず自分のネットワークのサブネットマスクに合わせる必要があります。	255.255.255.255

jp.co.nec.orb.PooledThreadNumber

説明	既定値
サーバ処理のスレッドポリシーとして PooledThread を選択した場合 (jp.co.nec.orb.ThreadPolicy=PooledThread 時)、クライアントからの要求をパラレル処理するプールスレッド数(1以上)を指定します。0以下の値、もしくは指定しない場合は既定値(10)で動作します。	10

jp.co.nec.orb.PropertyFile

説明	既定値
プロパティ定義を記述したプロパティ定義ファイル名を指定します。自動起動などの環境によっては絶対パス名での指定が必要になる場合があります。プロパティ定義ファイルへの記述は下記ようになります。 -プロパティ名 プロパティ値 プロパティ定義ファイルではコメントの記述も可能です。'#' を行の先頭または途中で記述することでコメントとみなされます。ただし行の途中でコメントの記述では、'#' の前に必ず空白またはタブを挿入してください。 PropertyFile は、ORB.init () を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。	指定しない

jp.co.nec.orb.ServerAddr

説明	既定値
複数のネットワークと接続しているサーバホストについて、クライアントからの要求を受け付けるアドレスを特定するために指定します。サーバホストのアドレスは、「host.nec.co.jp」のようなホスト名か、「10.8.45.84」のような IP アドレスの文字列表現で指定します。後述の UseHostName プロパティで yes が指定されていない場合は、ServerAddr をホスト名で指定していても IP アドレス形式に変換されます。	指定しない

jp.co.nec.orb.ServerCertAlias

説明	既定値
JSSSE で使用するサーバ証明書の鍵の別名を設定します。証明書ファイルが JKS 形式の場合のみ必要です。	指定しない

jp.co.nec.orb.ServerCertAliasPassPhrase

説明	既定値
JSSSE で使用するサーバ証明書の鍵の別名のパスワードを設定します。証明書ファイルが JKS 形式の場合のみ必要です。	指定しない

jp.co.nec.orb.ServerCertKey

説明	既定値
JSSSE で使用するサーバ証明書のファイル名を設定します。SSL 通信を使用する場合、必ず設定してください。	指定しない

jp.co.nec.orb.ServerCertKeyFileType

説明	既定値
JSSE で使用するサーバ証明書のファイルの形式を設定します。PKCS12 もしくは JKS を指定することができます。	PKCS12

jp.co.nec.orb.ServerCertKeyPassPhrase

説明	既定値
JSSE で使用するサーバ証明書のファイルのパスワードを設定します。	指定しない

jp.co.nec.orb.ServerCertKeyType

説明	既定値
JSSE で使用するサーバ証明書の鍵の形式を設定します。 このプロパティを指定しない場合は DSA が既定値として選択されます。	DSA

jp.co.nec.orb.ServerPort

説明	既定値
クライアントからの接続を固定のポート番号で待ち合わせる場合に指定します。 (注)WebOTX システムでは、誤動作を招く恐れがあるため、このプロパティは設定しないでください。	システムによって自動で設定される

jp.co.nec.orb.SoKeepAlive (KEEPALIVE)

説明	既定値
SO_KEEPALIVE ソケットオプションを指定します。 ・true - 有効 ・false - 無効 このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	true

jp.co.nec.orb.SoLinger

説明	既定値
SO_LINGER ソケットオプション (linger-on-close タイムアウト値 (秒)) を指定します。 このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	-1

jp.co.nec.orb.SoRcvBuf

説明	既定値
SO_RCVBUF ソケットオプション (着信ネットワーク入出力用としてプラットフォームで使われるバッファのサイズ (バイト)) を指定します。 このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	8192

jp.co.nec.orb.SoReuseAddr

説明	既定値
SO_REUSEADDR ソケットオプションを指定します。 このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	true

jp.co.nec.orb.SoSndBuf

説明	既定値
SO_SNDBUF ソケットオプション (送信ネットワーク入出力用としてプラットフォームで使われるバッファのサイズ (バイト)) を指定します。 このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	8192

jp.co.nec.orb.SoTimeout

説明	既定値
SO_TIMEOUT ソケットオプション (単位: ミリ秒) を指定します。このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	0

jp.co.nec.orb.SSLCipherSuites

説明	既定値
SSL で利用する CipherSuites の種類を指定します。使用する各 CipherSuites を ":" で区切って指定します。指定可能な CipherSuites については Sun の JSSE リファレンスガイドを参照ください。	JSSE を使用し、このプロパティを指定しない場合は JSSE の既定値にしたがいます。

jp.co.nec.orb.SSLPort

説明	既定値
SSL を用いた通信で使用するポート番号を指定します。サーバ側で有効です。 0 以上の値を指定した場合は SSL を使用して通信します。0 を指定した場合はポート番号をシステムが自動で割り当てます。 ospprxy を利用して applet 上でサーバを作成する場合は 0 を指定してください。1 以上を指定した場合は、ポート番号に指定された値が使用されます。指定しない場合は SSL による通信を行いません。	指定しない

jp.co.nec.orb.SSLProvider

説明	既定値
SSL 通信で使用する製品を指定します。以下の値を設定することができます。 •SSL-J •JSSE	JSSE

jp.co.nec.orb.TcpNoDelay(jp.co.nec.orb.TCPNODELAY)

説明	既定値
TCP_NODELAY ソケットオプションを指定します。 •true - 有効 •false - 無効 TCP_NODELAY ソケットオプションが無効な場合、TCP プロトコルのアルゴリズムにより、送信データがある一定量溜まるまで送出手が遅延することがあります。 既定値 true のままであれば、送出手が遅延することはありません。 このプロパティは、プロセス (VM) 内で、また、クライアント側、サーバ側で共通の設定となります。	true

jp.co.nec.orb.ThreadPolicy

説明	既定値
サーバ処理のスレッドポリシーを選択します。以下に ThreadPolicy に設定可能な値を示します。 •PerClientThread - クライアント単位にスレッドを生成し、そのスレッド上でクライアントからの要求をシリアルに処理します •PooledThread - クライアントからの要求をあらかじめプールされた複数のスレッドに配分して平行に処理します PerClientThread の場合、クライアントが異なれば、スレッドも異なるため、それらの要求は平行に処理します。 PooledThread の場合、プールスレッドは、クライアント単位ではなく、クライアント全体で共通のものです。また、その数は既定値 (10) ですが、プロパティ PooledThreadNumber によって変更することができます。 ThreadPolicy を指定しない場合、既定値 (PerClientThread) で動作します。	PerClientThread

jp.co.nec.orb.TrustedCACerts

説明	既定値
JSSE で使用する信頼する CA 証明書のファイル名を設定します。	指定しない

jp.co.nec.orb.TrustedCACertsFileType

説明	既定値
JSSE で使用する信頼する CA 証明書のファイルの形式を設定します。DER もしくは JKS を指定することができます。PEM 形式の場合は DER と指定してください。	DER

jp.co.nec.orb.TrustedCACertsPassPhrase

説明	既定値
JSSE で使用する信頼する CA 証明書のファイルのパスワードを設定します。証明書ファイルが JKS 形式の場合のみ必要です。	指定しない

jp.co.nec.orb.UseCodeBase

説明	既定値
コードベースダウンロード機能を使用するかどうかを指定します。使用する場合は true、使用しない場合は false を指定してください。	false

jp.co.nec.orb.UseCSI

説明	既定値
CSIv2 機能を使用するかどうかを指定します。使用する場合は true を指定します。使用しない場合は false を指定します。	false

jp.co.nec.orb.UseDynamicStub

説明	既定値
RMI-IIOP のダイナミック・プロキシによる通信モードを有効にするかどうかを指定します。false の場合は、Stub/Tie を事前に生成する必要があります。 true (既定値) - RMI-IIOP のダイナミック・プロキシによる通信モードを有効にします false - RMI-IIOP のダイナミック・プロキシによる通信モードを無効にします	true

jp.co.nec.orb.UseFragment

説明	既定値
メッセージをフラグメントに分割して送信するかどうかを設定します。 ・true - フラグメントに分割して送信する ・false - フラグメントに分割しないで一括して送信する true を指定した場合、ストリームが持つバッファのサイズ以上のデータを送信する時には一つのメッセージをバッファサイズ分に分割して送信します。 false を指定した場合、ストリームが持つバッファのサイズ以上のデータを送信する時にはバッファを拡張してデータを書き込みます。	false

jp.co.nec.orb.UseHostName

説明	既定値
オブジェクトリファレンスのホスト名に、デフォルトホスト名を無変換で設定するか否かを指定します。 ServerAddr プロパティで IP アドレス形式のホスト名が指定されている場合は、本プロパティの指定内容に関わらず、常に IP アドレス形式のホスト名がオブジェクトリファレンスに設定されます。 UseHostName に設定できる値を下記に示します。 yes - デフォルトホスト名を無変換で設定する no - デフォルトホスト名を IP アドレス形式に変換して設定する	no

jp.co.nec.orb.VersionCompatible

説明	既定値
ライブラリの互換バージョンを指定します。R6.x およびそれ以前のバージョンの Object Broker (ObjectSpinner) Java™ と相互に接続する場合は、6 を指定してください。R7.10 の Object Broker Java™ と相互に接続する場合は、710 を指定してください。R7.20 の Object Broker Java™ と相互に接続する場合は、7 を指定してください。	8

jp.co.nec.orb.UseMultiConnection(jp.co.nec.orb.MultiConnection)

説明	既定値
オブジェクトリファレンスが多重化されている場合に、クライアント側で接続の多重化を行うかどうかを指定します。 多重化を行う場合は true を指定します。多重化を行わない場合は false を指定します。 指定しない場合の既定値は true です。	true

jp.co.nec.orb.UseProfileBodyComponents

説明	既定値
<p>通常、WebOTX Object Broker Java™ では、IIOP1.1/1.2 時の IOR コンポーネント情報（コードセット等）を TAG_MULTIPLE_COMPONENT に格納します。</p> <p>本プロパティでは、コンポーネント情報を TAG_INTERNET_IOP に含まれる ProfileBody の中に格納するか否かを指定します。この指定はサーバ側でのみ有効となります。</p> <p>通常は指定する必要はありませんが、他社 ORB 製品との接続において、TAG_MULTIPLE_COMPONENTS の情報を認識できないような場合に有効です。</p> <ul style="list-style-type: none"> <li>•true - TAG_INTERNET_IOP に格納する</li> <li>•false - TAG_MULTIPLE_COMPONENT に格納する</li> </ul>	false

jp.co.nec.orb.UseSSL

説明	既定値
<p>SSLと非SSLの通信を受け付けるサーバにリクエストを発行する際、SSL通信を使用するかどうかを指定します。</p> <p>他社のORBを使用して構築されたサーバと接続する場合は、この設定を使用しないでください。</p> <ul style="list-style-type: none"> <li>•true - SSLを使用した通信を行います</li> <li>•false - SSLを使用しない通信を行います</li> </ul>	JSSE がインストールされていれば SSL で通信する

org.omg.CORBA.ORBClass

説明	既定値
<p>Object Broker Java を使用するために必要な設定です。jp.co.nec.orb.OSPORB を指定してください。</p> <p>(注)WebOTX システムでは、誤動作を招く恐れがあるため、このプロパティは設定しないでください。</p>	指定なし

org.omg.CORBA.ORBInitialHost

説明	既定値
<p>WebOTX Object Broker ランタイムが動作しているホスト名を設定します。もし WebOTX Object Broker ランタイムを初期値設定のままお使いになるのであれば、このプロパティを設定する必要はありません。</p> <p>ORBInitialHost は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。</p>	localhost

org.omg.CORBA.ORBInitialPort (jp.co.nec.orb.OadPort)

説明	既定値
<p>OAD のポート番号を設定します。もし WebOTX Object Broker ランタイムを初期値設定のままお使いになるのであれば、このプロパティを設定する必要はありません。</p> <p>ORBInitialPort は、ORB.init() を呼び出す時のプロパティ値を変更することで、ORB インスタンス毎に異なる設定をすることができます。</p>	9825

org.omg.CORBA.ORBSingletonClass

説明	既定値
<p>Object Broker Java を使用するために必要な設定です。</p> <p>jp.co.nec.orb.OSPORBSingleton と指定してください。</p> <p>(注)WebOTX システムでは、誤動作を招く恐れがあるため、このプロパティは設定しないでください。</p>	指定なし

com.nec.webotx.instanceRoot

説明	既定値
<p>クライアントアプリケーションからドメインごとの設定を参照する場合に、動作するドメインのルートディレクトリ (例:C:\WebOTX\domains\domain1) を設定してください。</p> <p>(注)WebOTX システムでは、誤動作を招く恐れがあるため、このプロパティは設定しないでください。</p>	指定なし



## 11.3.WebOTX Object Broker C++ における環境設定

この章では WebOTX Object Broker C++の環境設定について説明します。

### 項目の設定方法

次の設定ファイルを編集し、設定値を指定します。

設定ファイル:

`#{INSTANCE_ROOT}/config/ObjectBroker/orbconf`

設定ファイルでは以下の形式で指定します。

<設定名>=<設定値>

例)

`ListenBackLog=20`

WebOTX Object Broker C++における環境設定項目の一覧を、以下に示します。

名前サーバに関する設定項目:

設定名	説明	既定値
NameServiceFileSaveRetryCount	名前サーバ終了時にパーシステントデータを保存できないとき、データ書き込みの再試行を行う回数を設定します。設定できる上限の値は 2,147,483,647 です。この値を越えた値や負の値を設定してはいけません。	2
NameServiceCheckAlive	oadによる名前サーバの定期的な生死確認を行なうかどうかを on / off で指定します。デフォルトは off です。静止確認時に名前サーバからの応答がない場合は、oad から、名前サーバの情報を削除します。ネットワーク負荷の大きなシステムでは、誤削除を避けるため off を指定して下さい。	off
NameServiceManagementNDF	壊れている NDF ファイルは、別名のファイルに退避されますが、何世代まで退避しておくかを指定します。退避された NDF ファイルは、{OrbRoot}/conf 配下に格納されます。	2

複数の WebOTX Object Broker 利用アプリケーションをインストールしている場合、まれに、NameServiceHostName に他ホストの設定を指定し、かつ、そのホスト上で名前サーバを起動する必要があります。この場合は NameServiceExportHostName に自ホスト名を設定してください。

キャッシュ名前サーバに関する設定項目:

設定名	説明	既定値
CacheNameServiceCheckAlive	oadによるキャッシュ名前サーバの定期的な生死確認	off

	<p>を行なうかどうかを on / off で指定します。デフォルトは on です。静止確認時にキャッシュ名前サーバからの応答がない場合は、oad からキャッシュ名前サーバの情報を削除します。</p> <p>ネットワーク負荷の大きなシステムでは、誤削除を避けるため off を指定して下さい。</p>	
--	--	--

イベントサービスに関する設定項目：

設定名	説明	既定値
CosEventBufSize	<p>イベントサービスの内部バッファの大きさを指定します。内部バッファとしては、設定値+1 を確保しますが、常時使えるのは設定値までとなります。設定できる上限の値は 2,147,483,647 です(ただし、メモリの許す範囲)。この値を越えた値や負の値を設定してはいけません。</p>	128
CosEventDoHealthCheck	<p>true を指定すると、イベントチャンネルと接続している生産者または消費者との通信が CosEventPollInterval 以上途切れると生産者または消費者との接続を切ると同時に接続が切れたことを表わすイベントを各消費者に最優先で送ります。</p> <p>false を指定した場合には、この機能は働きません。</p>	false
CosEventPollInterval	<p>上記接続断判定に用いる時間を設定します。</p> <p>設定できる上限の値は 42,929,672,951 です。この値を越えた値や負の値を設定してはいけません。</p>	120

共通の設定項目：

設定名	説明	既定値
LogIdsInHex	<p>true を設定すると通信ログでのプロセス ID、スレッド ID を 16 進表示します。</p>	false
MaxPathMark	<p>LocationForward プロトコル処理で使われる IOR のキャッシュエントリ数を指定します。</p>	5
UseConnectionRetry	<p>true を指定すると、サーバとの接続が無効になった場合にリトライを行います。既定値は false で、その際はリトライを行いません。</p>	false
RetryForConnectionRoundRobin	<p>true を指定すると、接続ラウンドロビン利用時に、サーバとの接続が無効になった場合にリトライを行います。既定値は false で、その際はリトライを</p>	false

	行いません。	
ConnectTimeout	コネクション確立のタイマ値(秒)を指定します。既定値は3です。	3
MaxConnectRetry	コネクション確立失敗時のリトライ回数を指定します。既定値は10です。	10

※全体のタイムアウト時間は、(ConnectTimeout × MaxConnectRetry) 秒になります。

ただし、RequestTimeout の設定がこれより短い場合には RequestTimeout の値が使用されます。全体のタイムアウト時間を過ぎるか、コネクトのリトライを MaxConnectRetry 回行う間にコネクトができなければ、例外がスローされます。

インプリメンテーション単位のオプション設定:

1 サーバプロセスで扱うことのできるインプリメンテーションは通常1つなので、関数での設定ではインプリメンテーション名を CORBA::ORB::\_implementation\_name 関数で1度だけ指定します。各設定関数では指定しません。

これに対し、設定名を使った設定では、どのインプリメンテーションに対しての設定であるかを区別させるためにインプリメンテーション名の後ろに設定する内容を示すキーワードをつなげた設定名を使います。たとえば、サーバパス名の設定時には、

“インプリメンテーション名”PathName

とします。

例)

インプリメンテーション名 : SampleImpl

サーバパス名 : /XXX/YYY/ZZZ

のとき、サーバパス名の設定は

SampleImplPathName=/XXX/YYY/ZZZ

となります。

次の表にある設定はインプリメンテーション単位の設定です。

設定内容	設定名/関数		値の種類または範囲	既定値
インプリメンテーション名	設定	なし	任意の文字列	"noname"
	関数	_implementation_name (*1)		
サーバパス名	設定	"ImplName"PathName	サーバパスを表す文字列	""
	関数	_server_path_name (*1)		

サーバポート番号	設定	"ImplName"Port	0以上の整数(unsigned short 値)	0 (*2)
	関数	_server_port (*1)		
サーバの活性化方針	設定	"ImplName"ActPolicy	"corba_sharedserver"(共有サーバ) "corba_unsharedserver" (非共有サーバ) "corba_persistentserver" (永続サーバ) "corba_serverpermethod" (メソッドごとのサーバ)	共有サーバ
	関数	_server_activation_policy (*1)	Obi::SharedServer (共有サーバ) Obi::UnsharedServer (非共有サーバ) Obi::PersistentServer (永続サーバ) Obi::ServerPerMethod (メソッドごとのサーバ)	
サーバホスト名	設定	"ImplName"HostName	サーバが動作するホスト名	hostname で返される名前
	関数	_server_hostname (*1)		
サーバプロセス自動起動引数	設定	なし	ORB 用引数のみ	ORB 用引数のみ
	関数	_server_argument (*1)		
自動起動引数でORBが利用する位置指定	設定	なし	0以上の整数(unsigned long 値)	0
	関数	_server_argument_index (*1)		
サーバプロセスでのスレッド処理方針	設定	"ImplName"ThreadPolicy	指定なし (サーバはシングルスレッドで実行される) "permethod" (サーバはメソッドごとにスレッドを作成して実行される) "pool" (サーバはあらかじめ作成されたスレッドで実行される)	サーバはシングルスレッドで実行される
	関数	_server_thread_policy (*1)	Ob_thr_permethod (サーバはメソッドごとにスレッドを作成して実行される)	

			Ob_thr_pool (サーバはあらかじめ作成されたスレッドで実行される)  Ob_thr_default (サーバはシングルスレッドで実行される)	
--	--	--	---	--

※ "ImplName" はインプリメンテーション名を表します。

(\*1) クラス CORBA::ORB のメンバ関数

(\*2) 0 を指定すると実行時に OS が空いているポート番号を割り当てます。0 を指定したときだけ自動起動が可能です。0 以外を指定するとそのポートが使われず。また、0 以外を指定したときは自動起動されません。

(\*3) SSL を使用したサーバの自動起動はサポートしていません。

oad でのサーバプロセス自動起動に関するオプション設定:

Object Broker の既定値の設定ではサーバプロセスが立ち上がっていないときにクライアントがオブジェクトを呼び出すと、oad がサーバプロセスを自動的に立ち上げます。この自動起動に関して、設定ファイルとは別のファイルを使った設定項目として、以下の 2 点があります。

クライアントホスト名または IP アドレスにより当該ホストからの自動起動を許す(許さない)ことを指定できます。

`${INSTANCE_ROOT}/config/ObjectBroker` 配下の次のファイルで記述します。

actallow: 自動起動を許すホストのリスト

actdeny: 自動起動を許さないホストのリスト

これらのファイルはどちらか一方だけでも両方を置いてかまいません。また、両方置かないことも可能です。ファイルの組み合わせの意味は次のとおりとなります。

組み合わせ	意味
actallow、actdeny とも存在しない	すべてのホストからの自動起動を許す。
actdeny だけ存在する	actdeny に記述されていないホストからの自動起動を許す。
actallow だけ存在する	actallow に記述されていないホストからの自動起動は許さない。
両方存在する	actdeny に記述されていなくてかつ actallow に記述されているホストからの自動起動だけを許す。actallow、actdeny 双方に記述されているホストからの自動起動は許さない。どちらにも記述されていないホストからの自動起動は許さない。

2. 自動起動を許す(許さない)ファイルのパス名を指定できます。

`${INSTANCE_ROOT}/config/ObjectBroker` 配下の次のファイルで記述します。

exeallow: 自動起動を許すファイルパス名のリスト

exedeny: 自動起動を許さないファイルパス名のリスト

ホストでの制限同様これらのファイルはどちらか一方だけでも両方を置いてかまいません。また、両方置かないことも可能です。ファイルの組み合わせの意味は次のとおりとなります。

組み合わせ	意味
-------	----

exeallow、exedeny とも存在しない	すべてのファイルの自動起動を許す。
exedeny だけ存在する	exedeny に記述されていないファイルの自動起動を許す。
exeallow だけ存在する	exeallow に記述されていないファイルの自動起動は許さない。
両方存在する	exedeny に記述されていなくてかつ exeallow に記述されているファイルの自動起動だけを許す。exeallow、exedeny 双方に記述されているファイルの自動起動は許さない。どちらにも記述されていないファイルの自動起動は許さない。

形式)

1 行につき 1 ファイルパス名を記述します。このとき余分な空白を書いてはいけません。

例: /usr/local/bin/adminserver の自動起動を許す場合

exeallow ファイル:

/usr/local/bin/adminserver

Object Broker ではオブジェクトリファレンス中にサーバプログラムのファイル名および引数が含まれていますので、安全のためサーバ側での自動起動可能ホスト管理、自動起動可能ファイルパスの管理をしっかりと行うことをおすすめします。

なお、ファイル名の同一性判定は単純な文字列としての同一性で判定しています。それゆえ、//bin/rm、/bin/rm、/bin/./rm はそれぞれ別ファイルとして扱われますのでご注意ください。

Object Broker C++では下記のコードセットを利用することができます。

設定内容	設定名/引数名/関数		値の種類または範囲	既定値
string 型のネイティブコードセットの指定	設定名	NativeCodeSet	ISO8859-1	OSF_SJIS1 (Windows)  JIS_eucJP (HP-UX, Solaris, Linux)
	引数名	-ORBOSPINativeCodeSet	(ISO 8859-1:1987; Latin Alphabet No. 1)	
	関数	CORBA::ORB::_native_code_set	JIS_eucJP (JIS eucJP:1993; Japanese EUC) OSF_SJIS1 (OSF Japanese SJIS-1) ISO646 (1991 IRV (International Reference Version)) UTF8 (X/Open UTF-8; UCS Transformation Format 8 (UTF-8))	
wchar/wstring 型のネイティブコードセットの指定	設定名	NativeCodeSetW	UCS2_LEVEL1 (ISO/IEC 10646-1:1993; UCS2, Level 1)	UCS2_LEVEL1
	引数名	-ORBOSPINativeCodeSetW		
	関数	CORBA::ORB::_native_code_set_w		

string 型のコンバージョンコードセットの指定	設定名	ConversionCodeSets	ISO8859-1 (ISO 8859-1:1987; Latin Alphabet No. 1) JIS_eucJP (JIS eucJP:1993; Japanese EUC)	JIS_eucJP,ISO8859-1,ISO646 (Windows) OSF_SJIS1,ISO8859-1,ISO646 (HP-UX, Solaris, Linux)
	引数名	-ORBOSPIConversionCodeSets	OSF_SJIS1 (OSF Japanese SJIS-1) ISO646 (1991 IRV (International Reference Version)) UTF8	
	関数	CORBA::ORB::__conversion_code_sets	(X/Open UTF-8; UCS Transformation Format 8 (UTF-8)) から重複しない任意の組み合わせ	
wchar/wstring 型のコンバージョンコードセットの指定	設定名	ConversionCodeSetsW	UTF16 (ISO/IEC 10646-1:1993; UTF-16, UCS Transformation Format 16-bit form)	UTF16
	引数名	-ORBOSPIConversionCodeSetsW		
	関数	CORBA::ORB::__conversion_code_sets_w		

#### 設定方法

##### [設定]

初期設定はインストーラによって行われます。

##### [引数]

プログラム起動時の引数で設定する例を以下に示します。キーワードが複数記述されていたときは、最後のものを有効とします。

```
-ORBOSPINativeCodeSet OSF_SJIS1
-ORBOSPINativeCodeSetW UCS2_LEVEL1
-ORBOSPIConversionCodeSets JIS_eucJP,ISO8859-1,ISO646
-ORBOSPIConversionCodeSetsW UTF16
```

##### [関数]

関数で設定する例を以下に示します。設定は CORBA::ORB\_init 関数呼び出し直後に行います。途中で変更した場合の動作は不定です。コードセットの設定はプロセス単位で行われます。スレッド毎にコードセットを変更することはできません。

```
CORBA::ORB_ptr orbobj = CORBA::ORB_init(...);
```

```

orbobj->_native_code_set("OSF_SJIS1");
orbobj->_native_code_set_w("UCS2_LEVEL1");
Ob_StringSeq conv_cdsets;
conv_cdsets.length(3);
conv_cdsets[0] = CORBA_string_dup("JIS_eucJP");
conv_cdsets[1] = CORBA_string_dup("ISO8859-1");
conv_cdsets[2] = CORBA_string_dup("ISO646");
orbobj->_conversion_code_sets(conv_cdsets);
Ob_StringSeq conv_cdsets_w;
conv_cdsets_w.length(1);
conv_cdsets_w[0] = CORBA_string_dup("UTF16");
orbobj->_conversion_code_sets_w(conv_cdsets_w);

```

**注意:**

コードセットを使用するためには、GIOP および IIOP のバージョンを 1.1 以上に設定する必要があります。

**GIOP/IIOP に関するオプション設定:**

GIOP/IIOP に関する以下の設定項目は、設定ファイルまたは起動時引数(この引数は CORBA::ORB\_init 関数にそのまま渡します)で指定できます。

引数での指定例)

MyApp -ORBGIOPMinorVersion 1

引数で指定した場合、設定ファイルでの指定より優先されます。

設定名/引数	意味	既定値
GIOPMinorVersion	GIOP のマイナーバージョンを指定します。 GIOP1.0 を使用して通信を行うためには、0 を指定します。指定可能な値は、0、1、2 および 3 です。	3
-ORBGIOPMinorVersion		
IIOPMinorVersion	IIOP のマイナーバージョンを指定します。 IIOP1.0 を使用して通信を行うためには、0 を指定します。  ただし、GIOPMinorVersion の設定よりも大きい数字を指定したときは、GIOPMinorVersion と同じ設定になります。指定可能な値は、0、1、2 および 3 です。	3
-ORBIIOPMinorVersion		
UseFragment	true を指定した場合、メッセージ送信時にフラグメント分割します。	false
-ORBUseFragment		
AddWcharWidthOctet	true を指定した場合、wchar の直前に長さを示す文字列を付与します。	true



-ORBAddWcharWidthOctet	すオクテットを追加します。	
AddWstringWidthOctet	true を指定した場合、wstring の各文字の直前に長さを示すオクテットを追加します。	false
-ORBAddWstringWidthOctet		
PrepareWcharBOM	true を指定した場合、UTF-16 の wchar、wstring のデータをアンマーシャルするときに BOM が付加されたデータとして処理します。	false
-ORBPrepareWcharBOM		
PriorToBOMForMultiOctets	UTF-16 の wchar、wstring のデータをアンマーシャルするときに BOM が付加されていない場合に従うバイトオーダーを BOM 仕様のバイトオーダーとするかメッセージヘッダに指定されているバイトオーダーに従うかどうかを指定します。  true の場合は BOM 仕様に従います。  false の場合はメッセージヘッダに従います。	false
-ORBPriorToBOMForMultiOctets		
AllowNullString	true を指定した場合、wstring のデータをマーシャルするときに NULL 文字を許します。	false
-ORBAllowNullString		
TargetAddress	リクエスト送信時に使用するアドレスモードを指定します。  KeyAddr: ObjectKey を使用します。	KeyAddr
-ORBTargetAddress	ProfileAddr: IOR を使用します。  ReferenceAddr: 参照アドレス使用します。	
LocateReplyAlign8	true を指定した場合、LocateReply のとき、ヘッダの後ろで 8byte アラインを行います。	true
-ORBLocateReplyAlign8		
PrepareWcharBOMAtMarshaling	UTF16 の wchar および wstring のエンコードで BOM を付加するかどうかを指定します。  true を指定した場合、BOM を付加します。	false
WcharMarshalWithMessageByteorder	UTF16 の wchar および wstring をエンコードするときのバイトオーダーを指定します。  true を指定すると、メッセージのバイトオーダーでエンコードします。  false を指定すると、BigEndian(CORBA 仕様)でエンコードします。	true

OpenSSL 関連のオプション設定:

C++では、OpenSSL 関連の設定は、以下の設定(レジストリ)、または CORBA::ORB\_init()の引数のどちらかで行ないます。

C++のサーバの場合、ServerCertFile と SSLPort の設定が最低限必要です。証明書ファイルに private key が含まれていない場合、private key も指定する必要があります。Private key の指定は、以下の設定(レジストリ)または引数で行います。

設定名	意味	既定値
ServerCertFile	サーバ証明書を格納したファイルを指定する。	なし
ClientCertFile	クライアント証明書を格納したファイルを指定する。	なし
ServerCertKeyPassPhrase	サーバ証明書のパスフレーズを指定する。	なし
ClientCertKeyPassPhrase	クライアント証明書のパスフレーズを指定する。	なし
TrustedCACertsPassPhrase	信頼できる CA の証明書のパスフレーズを指定する。	なし
ServerPrivateKeyFile	サーバの private key を格納したファイルを指定する。	なし
ClientPrivateKeyFile	クライアントの private key を格納したファイルを指定する。	なし

[引数]

上記設定名に”-ORB”を付加した引数を以下の形式で指定します。

-ORB<設定名> <設定値>

例)

-ORBServerCertFile <証明書を格納したファイル名>

SSLPort と CertRequest については以下の設定を参照してください。

設定内容	設定名/関数	値の種類または範囲	既定値
サーバポート番号(SSL)	設定	”ImplName”SSLPort	0 (SSL を使用しない) (*3)
	関数	_use_ssl (*1)	
クライアントの証明書を要求	設定	”ImplName”CertRequest	false

サポートしている証明書のファイルの形式は、PEM 形式のみです。SSLPort と CertRequest については、レジストリで指定する場合は、設定名は、<インプリメンテーション名>を先頭に付加した名前になります。

例)

MyImplSSLPort

IPv6 関連のオプション設定:

C++では、IPv6 関連の設定は、以下の設定(レジストリ)、または CORBA::ORB\_init()の引数のどちらかで行いません。

設定名	説明	既定値
PreferIPv6Addresses	IPv4 と IPv6 の両方が利用可能な環境で、IPv6 のアドレスを優先するかどうかを指定します。true を指定すると、IPv6 アドレスを優先して利用します。false を指定すると IPv6 アドレスを優先します。  注意:ドメインの domain.ipv6-enable 属性の値を変更した場合、本設定値は、domain.ipv6-enable 属性の値で上書きされます。Object Broker のプロセスのみをドメインと異なる動作をさせたい場合には、ドメインの domain.ipv6-enable 属性の値変更後に本設定の値を変更する必要があります。	false
PreferIPv4Stack	IPv4 のみを利用可能にするかどうかを指定します。true を指定すると、IPv4 のみ利用可能となります(IPv6 が利用できる環境でも、IPv6 が利用されません)。false を指定した場合は、IPv4 と IPv6 の両方が利用可能になります。	false

#### [引数]

上記設定名に“-ORB”を付加した引数を以下の形式で指定します。

-ORB<設定名> <設定値>

例)

-ORBPreferIPv4Stack true

このほか、WebOTX Object Broker C++を利用するアプリケーションからドメインごとの設定を参照する場合には、起動時引数(この起動時引数は CORBA::ORB\_init() にそのまま渡します)に次の引数を追加してください。

-WODomain <動作するドメインのルートディレクトリ>

例)

myapp -WODomain “C:¥WebOTX¥domains¥domain1”

## 12.TPモニタに関する設定

### 12.1.エージェント設定項目・設定方法

TPモニタの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

### 12.2.TPモニタ設定項目一覧

TP モニタに関する設定項目について説明します。

TP モニタでは以下の MO を提供しています。それぞれの属性やオペレーションについては、「MO 定義リファレンス」を参照ください。なお以下の dotted-name 一覧イタリックで記述しているものは該当 MO の名前を示しています。

WebOTX システム(TP システム)

j2eeType : [WebOTXSystem](#)

dottedname : tpsystem

WebOTX アプリケーショングループリスト

j2eeType : [WebOTXApplicationGroupList](#)

dottedname : tpsystem.applicationGroups  
WebOTX アプリケーショングループ  
j2eeType : [WebOTXApplicationGroup](#)  
dottedname : tpsystem.applicationGroups.apg-name  
WebOTX プロセスグループリスト  
j2eeType : [WebOTXProcessGroupList](#)  
dottedname : tpsystem.applicationGroups.apg-name. processGroups  
WebOTX プロセスグループ  
j2eeType : [WebOTXCORBACPPPProcessGroup](#)  
j2eeType : [WebOTXCORBAJ2SEProcessGroup](#)  
j2eeType : [WebOTXJ2EEProcessGroup](#)  
dottedname : tpsystem.applicationGroups.apg-name. processGroups.pgname  
WebOTX 常駐オブジェクトリスト  
j2eeType : [WebOTXCORBACPPPPermanentObjectList](#)  
j2eeType : [WebOTXCORBAJ2SEPermanentObjectList](#)  
dottedname : tpsystem.applicationGroups.apg-name.processGroups.pgname.PermanentObjects  
WebOTX 常駐オブジェクト  
j2eeType : [WebOTXCORBACPPPPermanentObject](#)  
j2eeType : [WebOTXCORBAJ2SEPermanentObject](#)  
dottedname : tpsystem.applicationGroups.apg-name.processGroups.pgname.PermanentObjects.permobjname  
WebOTX Java 非同期メッセージ  
j2eeType : [WebOTXCORBAJ2SEJavaAsyncMsg](#)  
j2eeType : [WebOTXJ2EEJavaAsyncMsg](#)  
dottedname : tpsystem.applicationGroups.apg-name.processGroups.pgname.javaAsyncMsg  
WebOTX 接続クライアント情報  
j2eeType : [WebOTXClientSession](#)  
dottedname : tpsystem.ClientSession  
WebOTX IIOP リスナ  
j2eeType : [WebOTXIIOPListener](#)  
dottedname : tpsystem.IIOPListener  
WebOTX OLF リスナ  
j2eeType : [WebOTXOLFTPLListener](#)  
dottedname : tpsystem.OLFTPLListener  
WebOTX キュー情報  
j2eeType : [WebOTXQueue](#)  
dottedname : tpsystem.Queue  
WebOTX VD リスト  
j2eeType : [WebOTXVDResource](#)  
dottedname : tpsystem.VDInformation  
WebOTX VD

j2eeType : [WebOTXVD](#)  
dottedname : tpsystem.VDInformation.vdname  
WebOTX Adm リスナ  
j2eeType : [WebOTXAdmListener](#)  
dottedname : tpsystem.admListener  
WebOTX マルチサーバリスト  
j2eeType : [WebOTXMultiServerList](#)  
dottedname : tpsystem.multiServers  
WebOTX マルチサーバグループ  
j2eeType : [WebOTXMultiServerGroup](#)  
dottedname : tpsystem.multiServers.*multiserver-group-name*  
WebOTX マルチサーバセット  
j2eeType : [WebOTXMultiServerSet](#)  
dottedname : tpsystem.multiServers.*multiserver-group-name.multiserver-set-name*  
WebOTX systpp プロセス  
j2eeType : [WebOTXSYSTPP](#)  
dottedname : tpsystem.systpp  
WebOTX CORBA アプリケーション  
j2eeType : [WebOTXCORBAApplication](#)  
dottedname : applications.corba-applications.ap-name  
WebOTX CORBA モジュール  
j2eeType : [WebOTXCORBAModule](#)  
dottedname : applications.corba-applications.ap-name.module-name  
WebOTX CORBA オブジェクト(インタフェース)  
j2eeType : [WebOTXCORBAObject](#)  
dottedname : applications.corba-applications.*ap-name.module-name.interface-name*  
WebOTX CORBA オペレーション  
j2eeType : [WebOTXCORBAOperation](#)  
dottedname : applications.corba-applications.ap-name.module-name.interface-name.op-name  
WebOTX J2EE アプリケーション  
j2eeType : [WebOTXJ2EEApplication](#)  
dottedname : applications.j2ee-applications.ap-name  
WebOTX EJB モジュール  
j2eeType : [WebOTXEJBModule](#)  
dottedname : applications.j2ee-applications.ap-name.module-name  
WebOTX EJBBean オブジェクト(インタフェース)  
j2eeType : [WebOTXEntityBeanHome](#)  
j2eeType : [WebOTXEntityBeanRemote](#)  
j2eeType : [WebOTXMessageDrivenBeanClass](#)  
j2eeType : [WebOTXStatefulSessionBeanHome](#)

j2eeType : [WebOTXStatefulSessionBeanRemote](#)

j2eeType : [WebOTXStatelessSessionBeanHome](#)

j2eeType : [WebOTXStatelessSessionBeanRemote](#)

dottedname : applications.j2ee-applications.ap-name.module-name.bean-name.class-name

WebOTX EJBBean メソッド

j2eeType : [WebOTXEntityBeanHomeOperation](#)

j2eeType : [WebOTXEntityBeanRemoteOperation](#)

j2eeType : [WebOTXMessageDrivenBeanOperation](#)

j2eeType : [WebOTXStatefulSessionBeanHomeOperation](#)

j2eeType : [WebOTXStatefulSessionBeanRemoteOperation](#)

j2eeType : [WebOTXStatelessSessionBeanHomeOperation](#)

j2eeType : [WebOTXStatelessSessionBeanRemoteOperation](#)

dottedname : applications.j2ee-applications.ap-name.module-name.bean-name.class-name.method-name

WebOTX Web モジュール

j2eeType : [WebOTXWebModule](#)

dottedname : applications.j2ee-applications.ap-name.module-name

WebOTX Web コンテキストオブジェクト

j2eeType : [WebOTXWebContext](#)

j2eeType : [WebOTXWebContextInEar](#)

dottedname : applications.j2ee-applications.ap-name.module-name.context-name

WebOTX Web メソッド

j2eeType : [WebOTXWebMethod](#)

j2eeType : [WebOTXWebMethodInEar](#)

dottedname : applications.j2ee-applications.ap-name.module-name.context-name.method-name

WebOTX 共有コンポーネント

j2eeType : [WebOTXSharedComponent](#)

dottedname : applications.sharedComponents.*component-name*

# 13.OLF/TP Adapterに関する設定

## 13.1.OLF/TP Adapterに関する設定

OLF/TP Adapterを使用するには、OLF/TP Adapterの動作環境についての設定を行う必要があります。OLF/TP Adapterでは、実行時に通信環境定義(OLFAdapter.ini)、端末定義(OLFAdapter.xml)を参照して動作します。これらのファイルは通信に使用するリソース及び接続先の情報を定義しています。

これらの設定に加え、配備記述子(ra.xml)とよばれる OLF/TP Adapterのプロパティを設定する必要があります。配備記述子はJ2EEコネクタアーキテクチャ準拠の設定ファイルで、リソースアダプタのプロパティ情報を設定します。

## 13.2.通信環境定義(OLFAdapter.ini)

OLF/TP Adapterを使用してバックエンドサーバと通信するには、まず通信に使用するリソースを定義します。通信環境定義は以下のファイルを編集してください。

<OLF/TP Adapter インストールディレクトリ>/Adapter/OLFTP/Run/conf/OlfAdapter.ini

本定義では、以下の情報を定義します。

プロパティ名	既定値	説明
RmtDefinition	なし	【必須パラメータ】 端末定義ファイルパス。 例: RmtDefinition = E:¥¥winnt¥¥OlfAdapter.xml
UTConnection	4	OLF/TP-UT プロトコルの接続の開設可能総数の上限。UT プロトコルを使用して通信する場合に指定してください。
UWConnection	4	OLF/TP-UW プロトコルの接続の開設可能総数の上限。UW プロトコルを使用して通信する場合に指定してください。
MaxRecv	4	受信スレッドの最大値。リモート定義における delivery=realtime の接続の開設可能総数の上限値である
MaxSession	4	同時開設可能な最大セッション数。UTConnection+UWConnection の値以上を指定してください。
RequestNum	8	同時送受信可能なオブジェクトの最大数。UTConnection+UWConnection の値以上を指定してください。
MaxListenerThread	3	非同期受信を実行するスレッド数。1 以上を指定してください。
OlfByteBufferNum	8	送受信バッファの最大総数。(UTConnection+UWConnection) × 2 の値を指定してください。
OlfByteBufferSize	10000	送受信バッファサイズ。送受信データの最大サイズ + 64 バイトを指定してください





TraceLogBufferSize	1024	バッファリングするサイズ。【10～1,048,576】 なお、ログファイルは行単位で即時出力され、バッファリングされません。
TraceFlushingInterval	5	トレースファイルがバッファ内に滞留する最大時間(秒)【1以上】 なお、実際は本値 x1000 が PollingInterval(ms)の整数倍となります。
UTport		OLF/TP-UT プロトコルでの着信ポート番号 ※着信接続を行う場合に設定します。
UWport		OLF/TP-UW プロトコルでの着信ポート番号 ※着信接続を行う場合に設定します。
PortError	"terminate" または" ignore"	指定した着信ポートがすでに使用されている、もしくは着信ポートの準備に失敗した場合の動作を指定します。 terminate OLF ライブラリの初期化に失敗します。 ignore 失敗した着信ポートは着信不可となり、初期化は成功します。 ※着信接続を行う場合に設定します。

(注意: 定義を変更する場合は、AP サーバを停止してください。また、TraceLogDirなどを既定値から変更した場合は server.policy ファイルに追記が必要です。)

### 13.3. 端末定義(OLFAdapter.xml)

OLF/TP Adapter を使用しての通信相手となるホストの情報を定義します。端末定義は以下のファイルを編集してください。

<OLF/TP Adapter インストールディレクトリ>/Adapter/OLFTP/Run/conf/OlfAdapter.xml

本定義では、以下の情報を定義します

タグ名	既定値	説明				
AckTimeout	5 秒	Ack/Conf 監視時間を設定します(単位: 秒、あるいは nolim、0 は nolim と同じ、既定値=5)。 送信時の送達確認の受信待ち時間であり、指定時間以上経過した場合、コネクションが切断されます。また、UW における PLstop も、本時間で監視します。				
DefaultRmtDef		RmtDef 配下で省略された各エレメントの既定値を設定します。 本定義は Rmt 定義内に一回だけ記述できますが、省略することもできます。記述する場合、後続の RmtDef の既定値が、本設定値になります。 配下にもてるエレメントは、次のとおり。RmtAddr, RmtPort, MaxConnection, Type, InitTimeout, AckTimeout, ResendInterval, ResendCount, ReplyTimeout				
Delivery	polling	非同期メッセージの受信監視の方式を指定します。 <table border="1"> <tr> <td>polling</td> <td>OLF/TP Adapter の polling 監視スレッドで、対象となるコネクションをすべて受信監視します。</td> </tr> <tr> <td>realtime</td> <td>対象となるコネクションごとに監視スレッドを用意します。</td> </tr> </table>	polling	OLF/TP Adapter の polling 監視スレッドで、対象となるコネクションをすべて受信監視します。	realtime	対象となるコネクションごとに監視スレッドを用意します。
polling	OLF/TP Adapter の polling 監視スレッドで、対象となるコネクションをすべて受信監視します。					
realtime	対象となるコネクションごとに監視スレッドを用意します。					

InitTimeout	30	接続時の活性監視時間を設定します(単位: 秒、あるいは nolim、0 は nolim と同じ、既定値=30)。コネクション接続時の DFinet、もしくは Pstart、ならびに UW のセッション取得時の PLstart の監視を行います。指定時間以上経過した場合、コネクションは切断され、そのとき発行している getSession は失敗します。
MaxConnection	1	最大接続コネクション数です。 1 以上の整数を指定してください。
PoolConnect	atRequest	OLF/TP-UT プロトコルを使用する場合は、atRequest を指定してください。OLF/TP-UW プロトコルを使用する場合は、initial を指定してください。
ReplyTimeout	30	応答時間を設定します(単位: 秒、あるいは nolim、0 は nolim と同じ)
ResendCount	5	送信再送回数を指定します。 指定回数再送し、受け付けられなければ、送信失敗となります。
ResendInterval	30	再送間隔を指定します(単位: 秒、0 は不可)。 TXack あるいは TXconf で、RETRY を受信したとき再送する間隔です
RmtAddr	connect	接続先のアドレスを指定します。 ホスト名(hosts もしくは DNS 登録されたもの) あるいは IP アドレス形式で指定必須。
RmtDef		一つの接続先を定義します。各種プロパティで指定していないものは、DefaultRmt で既定値を指定できます。 配下にもてるエレメントは、次のとおり。 RmtName, RmtAddr, RmtPort, MaxConnection, Type, Delivery, PoolConnect, InitTimeout, AckTimeout, ResendInterval, ResendCount, ReplyTimeout
RmtName		接続先のシンボリック名を指定します。 本値は配備記述子の ServerName で指定します。 本エレメントは RmtDef 配下で必須であり、値はリモート定義全体で一意でなければなりません。 本エレメントは DefaultRmtDef 配下で指定できません。
RmtPort	設定していても参照されない。 (ConnectWay="connect"時 のみ必須である)	接続先の着信ポートを指定します。
ConnectWay	"accept"	接続方法を指定します。 発信時は connect(既定値)、着信時は accept を指定します。

Type	セッションの種別を指定します。	
	UT-stateful	OLF/TP-UT プロトコルを使用する場合に指定します。
	UW	OLF/TP-UW プロトコルを使用する場合に指定します。

(注意: 定義を変更する場合は、AP サーバを停止してください。)

## 13.4. 配備記述子(ra.xml)

OLF/TP Adapter のプロパティを指定するファイルです。OLF/TP Adapter のプロパティは、<config-property>タグで指定します。管理環境に OLF/TP Adapter を配備する場合は、配備ツールでプロパティを設定します。非管理環境で OLF/TP Adapter を利用する場合は、サンプルの ra.xml をテキストエディタで編集してください。

OLF/TP Adapter のプロパティを以下に示します。

(注: 大文字/小文字は区別されます)

プロパティ名	型	既定値	説明
ASync	java.lang.Boolean	false	OLF/TP Adapter で非同期電文受信を行う場合に true を指定します。
CodeType	java.lang.String	"JIPSE"	接続先ホストが使用しているコード種別を指定します。以下のコード種別が指定できます。(省略可) <ul style="list-style-type: none"> <li>• JIPSE</li> <li>• JIPS</li> <li>• SJIS</li> <li>• EUC</li> </ul>
EndianType	java.lang.String	"big-endian"	接続先ホストの endian タイプを指定します。指定できる値は以下の通りです。 <ul style="list-style-type: none"> <li>• big-endian</li> <li>• little-endian</li> </ul>
ExecutionTimeout	java.lang.Integer	60000	受信待ち時間をミリ秒単位で指定します(省略可)。InteractionSpec の ExecutionTimeout に-1 を指定している場合、本指定を受信待ち時間として使用します。OLF/TP Adapter では秒単位で受信を待ち合わせます。よって、待ち時間は ExecutionTimeout/1000(秒)となります。0 を指定した場合は受信が完了するまで AP コンポーネントに制御が戻りません。
HostType	java.lang.String	"ACOS4"	接続先のホスト種別を指定します(省略可)。以下のホスト種別が指定できます。 <ul style="list-style-type: none"> <li>• ACOS2</li> <li>• ACOS4</li> <li>• ACOS6</li> <li>• Windows</li> <li>• UNIX</li> </ul>
JISType	java.lang.String	"JIS78"	JIS 年度を指定します(省略可)。本指定は CodeType が "JIPSE"、"JIPS" のときのみ有効です。以下の値が指定できます。 <ul style="list-style-type: none"> <li>• JIS78</li> <li>• JIS90</li> </ul>

JMSConnectionFactoryName	java.lang.String	""	非同期電文受信機能を使用する場合、JMS の QueueConnectionFactory の JNDI 名を指定します。
LIName	java.lang.String	""	業務ログイン済みのコネクションをプーリングする場合に指定します(省略可)。接続先業務名を指定します。
VDName	java.lang.String	""	VD ログイン済みのコネクションを作成する場合に指定します(省略可)。VD 名を指定します。
OLFConfigFile	java.lang.String	""	通信環境定義(OLFAdapter.ini)ファイルのパスを絶対パスで指定します。日本語を含むファイルパスを指定してはいけません。 例: E:\winnt\OLFAdapter.ini
PackageName	java.lang.String	""	電文フォーマットクラス、レコードマッピングクラスのパッケージ名を指定します(省略可)。
ProtocolType	java.lang.String	"UT"	通信プロトコルを指定します(省略可)。指定できる値は以下の通りです。 <ul style="list-style-type: none"> <li>• UT</li> <li>• UW</li> </ul>
QueueName	java.lang.String	""	非同期電文受信機能を使用する場合、JMS の Queue の JNDI 名を指定します。
ServerName	java.lang.String	""	接続先ホスト名を指定します。接続先ホスト名は、端末定義(OLFAdapter.xml)で定義しておく必要があります。着信接続を行う場合にはタグそのものを記述しません。
TraceLevel	java.lang.Integer	0	トレースレベルを指定します(省略可)。指定できる値は以下の通りです。 <ul style="list-style-type: none"> <li>• 0…エラートレース</li> <li>• 1…エラートレース+CCIトレース</li> <li>• 2…エラートレース+CCIトレース+SPIトレース</li> </ul> AP サーバが J2EE コネクタのトレース機能をサポートしている場合、AP サーバで指定したファイルにトレースを出力します。AP サーバがトレース機能をサポートしていない場合は標準出力にトレースを出力します。
TXIDLength	java.lang.Integer	8	OLF/TP Adapter では、OLF/TP の拡張領域に含まれているトランザクション ID を使用して受信データの解析を行います。受信電文に OLF/TP の拡張領域が含まれていない場合、OLF/TP Adapter では、ユーザデータ中の任意のフィールドをトランザクション ID として扱います。本プロパティでは、電文中に含まれるトランザクション ID の長さを指定します(省略可)。

TXIDPosition	java.lang.Integer	0	OLF/TP Adapter では、OLF/TP の拡張領域に含まれているトランザクション ID を使用して受信データの解析を行います。受信電文に OLF/TP の拡張領域が含まれていない場合、OLF/TP Adapter では、ユーザデータ中の任意のフィールドをトランザクション ID として扱います。本プロパティでは、電文中に含まれるトランザクション ID の位置を指定します(省略可)。
UserName	java.lang.String	""	QueueConnection を作成時のユーザー名を指定します(指定がない場合でも動作可能)
Password	java.lang.String	""	QueueConnection を作成時のパスワードを指定します(指定がない場合でも動作可能)
RecordType	java.lang.String	"Indexed Record"	非同期受信データ型を指定します。IndexedRecord と MappedRecord を指定可能です。デフォルトは、IndexedRecord です。
UnpackType	java.lang.String	" EBCDIC_EBCDIC "	文字集合と符号の組み合わせを指定します。指定できる組み合わせは以下の通りです。 <ul style="list-style-type: none"> <li>• EBCDIC_EBCDIC…文字集合=EBCDIC／符号=EBCDIC(ACOS4)</li> <li>• ASCII_EBCDIC…文字集合=ASCII／符号=EBCDIC(ACOS4 以外)</li> <li>• ASCII_ASCII…文字集合=ASCII／符号=ASCII(ACOS4 以外 MF-COBOL)</li> </ul>

# 14. Working Domain Coordinatorに関する設定

Working Domain Coordinatorに関する設定について説明します。

## 14.1. Working Domain Coordinator全般の設定項目・設定方法

Working Domain Coordinatorの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

補足：

設定された属性はそれぞれ以下のディレクトリ配下に格納されます。

Working Domain coordinator	<code>\${INSTANCE_ROOT}/config/persistent-mbean/runtime/WorkingDomainCoordinator</code>
ビジネスロジックグループ	<code>\${INSTANCE_ROOT}/config/persistent-mbean/runtime/WDCControlledBusinessLogicGroup</code>
ビジネスロジック	<code>\${INSTANCE_ROOT}/config/persistent-mbean/runtime/WDCControlledBusinessLogic</code>
ロードバランサ	<code>\${INSTANCE_ROOT}/config/persistent-mbean/runtime/WDCLoadBalancer</code>
制御対象サーバ	<code>\${INSTANCE_ROOT}/config/persistent-mbean/runtime/WDCControlledServer</code>
制御対象ドメイン	<code>\${INSTANCE_ROOT}/config/persistent-mbean/runtime/WDCControlledDomain</code>

また、各 MO の名前の一覧情報を以下のファイルに格納しています。

ビジネスロジックグループ	<code>\${INSTANCE_ROOT}/config/wowdc/businesslogicgroups.properties</code>
ビジネスロジック	<code>\${INSTANCE_ROOT}/config/wowdc/businesslogics.properties</code>
ロードバランサ	<code>\${INSTANCE_ROOT}/config/wowdc/loadbalancers.properties</code>
制御対象サーバ	<code>\${INSTANCE_ROOT}/config/wowdc/servers.properties</code>
制御対象ドメイン	<code>\${INSTANCE_ROOT}/config/wowdc/domains.properties</code>

これらのファイルの内容には、手を加えないでください。

## 14.2. Working Domain Coordinatorに関する設定

以下は MO で設定可能な Working Domain Coordinator 設定項目の一覧です。

Dottedname : `server.workingDomainCoordinator`

項目 [属性名: attribute-name]	説明	既定値
ドメイン停止の待ち合わせ時間 [domainStopWaitingTime]	制御対象ドメイン停止時の待ち合わせ時間です。(単位:ミリ秒) 現在、この設定は無効です。この値を設定しても、ドメイン停止の待ち合わせは行われません。	1
ログレベル [logLevel]	Working Domain Coordinator のログレベルです。 指定されたレベルでのロギングを有効にすると、より高いすべてのレベルでのロギングも有効になります。レベルは降順で次のとおりです。 <b>SEVERE</b> : 重大な障害を示すメッセージを出力します。 <b>WARNING</b> : 潜在的な問題を示すメッセージを出力します。	INFO

	<p><b>INFO</b> : 動作に関する主なメッセージを出力します。</p> <p><b>CONFIG</b> : 静的な構成メッセージを出力します。</p> <p><b>FINE</b> : トレースメッセージを出力します。</p> <p><b>FINER</b> : 詳細なトレースメッセージを出力します。</p> <p><b>FINEST</b> : <b>FINER</b> よりも詳細なトレースメッセージを出力します。</p> <p>なお、<b>OFF</b> を指定するとログは出力されません。</p>	
ドメイン切替時のワークスレッド最大値 [maxThreads]	ドメイン切り替え時のワークスレッド数の最大値です。	10
ドメイン切替時のワークスレッド最小値 [minThreads]	ドメイン切り替え時のワークスレッド数の最小値です。	2

### 14.3.ビジネスロジックグループに関する設定

以下は MO で設定可能なビジネスロジックグループの設定項目の一覧です。

Dottedname : server.workingDomainCoordinator.businessLogicGroups.*group-name*

*group-name* には、ビジネスロジックグループ名が設定されます。

項目 [属性名: attribute-name]	説明	既定値
ビジネスロジックグループ名 [businessLogicGroupName]	ビジネスロジックグループ名です。この名前を変更しないでください。	なし
キュー滞留数上限値 [maxQueuingRequests]	キュー滞留数上限値を指定します。定期監視および、サンプリング監視時、キュー滞留数がキュー滞留数上限値を上回った場合、高負荷状態とみなされます。	20
定期監視間隔 [observationInterval]	キュー滞留数の定期監視間隔を指定します。(単位: 秒)	600
高負荷時監視間隔 [sampleObservationInterval]	キュー滞留数の上限値を超える高負荷時監視間隔を指定します。定期監視で高負荷とみなされた場合、監視の間隔をこの高負荷時監視間隔に切り替え、キュー滞留数のサンプリングの採取を行います。(単位: 秒)	60
高負荷時サンプリング回数 [sampleObservationCount]	高負荷時の監視回数を指定します。定期監視で高負荷とみなされた場合、監視の間隔を高負荷時監視間隔に切り替え、キュー滞留数のサンプリング採取を行います。	10
高負荷検出率 [highLoadRate]	高負荷検出率を指定します。高負荷時の監視回数分、キュー滞留数の監視を行い、キュー滞留数上限値を超えた回数がこの率を超えた場合、継続的な高負荷状態とみなし、ビジネスロジックの切り替えを行います。	60
優先度 [priority]	優先度を指定します。切り替え処理が行われる際、停止対象となるビジネスロジックの選択のパラメータとなります。この値が低いビジネスロジックから切り替えの対象となります。また、この値に 5 が設定されている場合は、切り替えの対象となりません。	1
稼動サーバ数上限値	稼動サーバ数の上限値を指定します。継続的な高負荷状態にあるビジネス	1

[maxServers]	スロジックが動作するサーバ数が、既にこの稼動サーバ数上限値に達している場合、切り替え処理は行われません。	
稼動サーバ数下限値 [minServers]	稼動サーバ数の下限値を指定します。切り替え処理で停止対象となったビジネスロジックが動作するサーバ数が、停止後、稼動サーバ数下限値を下回る場合、切り替え処理は行われません。	1

## 14.4.ビジネスロジックに関する設定

以下は MO で設定可能なビジネスロジックの設定項目の一覧です。

Dottedname : server.workingDomainCoordinator.businessLogicGroups.*group-name*.businessLogic.*logic-name*

*group-name* には、ビジネスロジックグループ名が設定されます。

*logic-name* には、ビジネスロジック名が設定されます。

項目 [属性名: attribute-name]	説明	既定値
ビジネスロジック名 [businessLogicName]	監視の対象となるプロセスグループ名を指定します。	なし

## 14.5.ロードバランサに関する設定

以下は MO で設定可能なロードバランサの設定項目の一覧です。

Dottedname : server.workingDomainCoordinator.loadBalancers.*loadbalancer-name*

*loadbalancer-name* には、ロードバランサ名が設定されます。

項目 [属性名: attribute-name]	説明	既定値
ロードバランサ名 [loadBalancerName]	ロードバランサ名です。この名前を変更しないでください。	なし
ロードバランサのホスト名 [hostName]	ロードバランサに接続する時のホスト名です。 ロードバランサが BIG-IP の場合、iControl のエンドポイントのホスト名を指定してください。	なし
ロードバランサのポート番号 [portNumber]	ロードバランサに接続する時のポート番号です。 ロードバランサが BIG-IP の場合、iControl のエンドポイントのポート番号を指定してください。	443
ロードバランサのバージョン [version]	ロードバランサのバージョンです。	なし
ロードバランサのユーザ名 [userName]	ロードバランサの管理者のユーザ名です。 ロードバランサが BIG-IP の場合、iControl のエンドポイントのベーシック認証用ユーザ名を指定してください。	なし
ロードバランサのパスワード [password]	ロードバランサの管理者のパスワードです。 ロードバランサが BIG-IP の場合、iControl のエンドポイントのベーシック認証用パスワードを指定してください。	なし
ロードバランサのタイプ	ロードバランサのタイプです。	999 (OTHER)



[lbType]	(1:BIG-IP 999:OTHER)	
ロードバランサ制御モード [lbControlMode]	ロードバランサの制御モードです。  1 : LB 制御あり  制御対象ドメインに設定された、ロードバランサの振分先の制御を自動的に行います。ドメイン停止処理時、ロードバランサから振分先を削除するコマンドを発行し、そのドメインに滞留するリクエストがすべて処理されたことを確認した後で、実際の停止を行います。  2 : LB 制御なし  ロードバランサに対して振分先の制御を行いません。振分先の制御は、ロードバランサのヘルスチェック機能により行われます。  3 : LB 制御依頼  ロードバランサの振分先の制御を運用担当者に依頼するモードです。ドメインの切り替えを行う際に、ロードバランサの振分先の制御依頼メッセージをイベントログ(Unix では syslog)に出力します。運用担当者は、そのメッセージに記載された制御対象ドメインの振分先設定を行ってください。	1 (LB 制御あり)
振分先追加処理のリトライ回数 [registerToLbCommandRetryMax]	振分先追加処理のリトライ回数を指定します。	1
振分先追加処理の終了の待ち合わせ時間 [registerToLbCommandTimeout]	振分先追加処理の終了の待ち合わせ時間を指定します。(単位:ミリ秒)	30
振分先削除処理のリトライ回数 [removeFromLbCommandRetryMax]	振分先削除処理のリトライ回数を指定します。	1
振分先削除処理の終了の待ち合わせ時間 [removeFromLbCommandTimeout]	振分先削除処理の終了の待ち合わせ時間を指定します。(単位:ミリ秒)	30

## 14.6. 制御対象サーバに関する設定

以下は MO で設定可能な制御対象サーバの設定項目の一覧です。

Dottedname : server.workingDomainCoordinator.controlledServers.*server-name*

*server-name* には、制御対象サーバ名が設定されます。

項目 [属性名: attribute-name]	説明	既定値
制御対象サーバ名 [serverName]	制御対象サーバ名を指定します。TCP ホスト名などの識別名を指定してください。	localhost

JMX Remote URL [jmxAdminRemoteURL]	制御対象サーバで動作する管理ドメインと接続するための JMX Remote URL を指定します。規定値のホスト名やポート番号を必要に応じて変更してください。	service:jmx:jmxmp ://localhost:6202
デフォルトドメイン名 [defaultDomainName]	Working Domain Coordinator の起動時に、制御対象サーバで制御対象ドメインが1つも起動されていない場合に起動する制御対象ドメイン名を指定します。  Working Domain Coordinator の停止時、デフォルトドメインの停止は行われません。	domain1
サーバの制御状態 [controlledStatus]	サーバに対する Working Domain Coordinator の負荷分散制御状態を表します。読み取り専用の属性です。  0: 負荷分散制御中 1: 停止制御再開指示待ち	なし
サーバの負荷状態 [serverState]	最大キュー滞留数に対する、制御対象サーバの現在のキュー滞留数の割合の最大値です。読み取り専用の属性です。	なし

## 14.7. 制御対象ドメインに関する設定

以下は MO で設定可能な制御対象ドメインの設定項目の一覧です。

Dottedname : `server.workingDomainCoordinator.controlledServers.server-name.controlledDomains.domain-name`

`server-name` には、制御対象サーバ名が設定されます。

`domain-name` には、制御対象ドメイン名が設定されます。

項目 [属性名: attribute-name]	説明	既定値
制御対象ドメイン名 [domainName]	制御対象ドメイン名を指定します。この名前を変更しないでください。	domain1
ビジネスロジックグループ名 [businessLogicGroupName]	ビジネスロジックを束ねるビジネスロジックグループの名前を指定します。	なし
LB の振分先に追加するコマンド [domainAdditionalCommands]	LB の振分先に追加するコマンドを指定します。LB 制御モードが LB 制御ありに設定されており、切り替え処理が起動した場合に、発行されるコマンドです。複数のコマンドを登録することができます。コマンドは登録された順番に発行されます。	なし
LB の振分先から削除するコマンド [domainDeleteCommands]	LB の振分先から削除するコマンドを指定します。LB 制御モードが LB 制御ありに設定されており、切り替え処理が起動した場合に、発行されるコマンドです。複数のコマンドを登録することができます。コマンドは登録された順番に発行されます。	なし
JMX Remote URL [jmxRemoteURL]	ユーザドメインと接続するための JMX Remote URL を指定します。規定値のホスト名やポート番号を必要に応じて変更してください。	service:jmx:jmxmp ://localhost:6212
ロードバランサ名 [loadBalancerName]	振分先制御が行われるロードバランサ名を指定します。	なし

振分先のグループ名 [ lbGroupName]	ロードバランサの振分先の論理的なグループ名を指定します。 ロードバランサが BIG-IP の場合、プール名を指定してください。プール名が指定されていない場合は、ビジネスロジックグループ名がプール名として使用されます。	なし
振分先ポート番号 [ lbPortNumbers]	ロードバランサによって振分が行われるアプリケーションサーバ側のポート番号を指定します。複数設定する場合は、ポート番号をカンマ区切りで入力してください。	80

# 15. データベースコントローラに関する設定

データベースコントローラに関する設定について説明します。

## 15.1. データベースコントローラ全般の設定項目・設定方法

データベースコントローラの設定項目については全てMO化されています。項目の設定方法については「2各サービスの属性の設定」を参照してください。

補足:

設定された属性は、以下のディレクトリ配下に格納されます。

```
データベースコントローラ    ${INSTANCE_ROOT}/config/persistent-mbean/runtime/DatabaseController
```

これらのファイルの内容には、手を加えないでください。

## 15.2. データベースコントローラに関する設定

以下は MO で設定可能な データベースコントローラ 設定項目の一覧です。

Dottedname : server.databaseController

項目 [属性名: attribute-name]	説明	既定値
JDBC データソース名 [checkDataSourceName]	状態監視を行う JDBC データソースの JNDI サーバへの登録名を指定します。	なし
データベースの状態監視コマンド [checkDatabaseCommand]	データベースの状態監視のために使用する SQL 命令を指定します。 JDBC データソースの状態監視コマンドが設定されていない場合に指定してください。SELECT 命令など、ロックの競合が発生しない SQL 命令を指定してください。	なし
データベースの起動コマンド [startCommands]	データベースサーバの起動コマンドを指定します。 データベースを開始する場合に、発行されるコマンドです。 複数のコマンドを登録することができます。コマンドは登録された順番に発行されます。	なし
データベースの停止コマンド [stopCommands]	データベースサーバの停止コマンドを指定します。 データベースを停止する場合に、発行されるコマンドです。 複数のコマンドを登録することができます。コマンドは登録された順番に発行されます。	なし
コマンド発行の待ち合わせ時間 [commandWaitTime]	開始または終了コマンドの待ち合わせ時間を指定します。(単位:ミリ秒)	1000
ドメイン起動時の連動起動使用有無 [startup]	ドメイン起動に連動してデータベース起動を行うかどうかを指定します。	false

既定値について

JavaSE6 (JavaDB インストール済) でかつ、WebOTX Developer のテストサーバの場合、インストール時又はドメイン生成時に JavaDB (Apache

Derby)をデフォルトの制御対象データベースとして既定値を自動生成します。

既定値の自動生成に合わせて、JDBC データソース(jdbc/Derby)の自動登録、JavaDB(Apache Derby)に必要な環境変数(DERBY\_HOME)の自動設定、JavaDB(Apache Derby)のJDBCドライバの自動コピー(>\${INSTANCE\_ROOT}/lib/ext/derbyclient.jar)が実施されます。

自動生成するデータベースコントローラの既定値は以下の通りです。

項目 [属性名: attribute-name]	既定値
JDBC データソース名 [checkDataSourceName]	jdbc/Derby
データベースの状態監視コマンド [checkDatabaseCommand]	なし
データベースの起動コマンド [startCommands]	<p>Windows の場合</p> <pre> \${ProgramFiles}¥Sun¥JavaDB¥frameworks¥NetworkServer¥bin¥startNetworkServer.bat localhost 1527 </pre> <p>または、</p> <pre> \${JAVA_HOME}¥frameworks¥NetworkServer¥bin¥startNetworkServer.bat localhost 1527 </pre> <p>Unix の場合</p> <pre> /opt/sun/javadb/frameworks/NetworkServer/bin/startNetworkServer.ksh localhost 1527 </pre> <p>または、</p> <pre> \${JAVA_HOME}/frameworks/NetworkServer/bin/startNetworkServer.ksh localhost 1527 </pre>
データベースの停止コマンド [stopCommands]	<p>Windows の場合</p> <pre> \${ProgramFiles}¥Sun¥JavaDB¥frameworks¥NetworkServer¥bin¥stopNetworkServer.bat localhost 1527 </pre> <p>または、</p> <pre> \${JAVA_HOME}¥frameworks¥NetworkServer¥bin¥stopNetworkServer.bat localhost 1527 </pre> <p>Unix の場合</p> <pre> /opt/sun/javadb/frameworks/NetworkServer/bin/stopNetworkServer.ksh localhost 1527 </pre> <p>または、</p> <pre> \${JAVA_HOME}/frameworks/NetworkServer/bin/stopNetworkServer.ksh localhost 1527 </pre>
コマンド発行の待ち合わせ時間 [commandWaitTime]	1000
ドメイン起動時の連動起動使用有無 [startup]	true