

WebOTX 運用編(ドメインの運用)

WebOTX 運用編

バージョン: 7.1

版数: 第 11 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. サービスの起動・停止	2
2.1. WEBOTXのサービスについて	2
2.2. サービスの起動・停止方法	5
2.3. サービスの設定	6
3. システムに関する設定	7
3.1. システム環境変数	7
3.2. 使用するポート番号	9
3.3. 動作プロセスについて	12
3.4. ディレクトリ構成	18
3.5. JAVA J2SEの設定	19
3.6. WEBOTX管理ユーザ設定 (HP-UX,SOLARIS,LINUX)	21
3.6.1. WebOTX管理ユーザ(運用ユーザ)の設定	21
3.6.2. 複数のドメインを異なるユーザで運用するための手順	22
3.7. ホスト名の変更	27
3.8. WINDOWSサービスアカウントの変更	30
4. ドメインの運用操作	31
4.1. ドメインの作成・削除	31
4.2. ドメインの確認	36
4.2.1. ドメインの一覧確認	36
4.2.2. ドメインの起動完了確認	36
4.3. ドメインの起動・停止	38
4.4. 運用ユーザの設定	40
4.5. 各サービスの起動・停止	40
4.6. ポリシーファイルの設定	43
4.7. JAVA VM環境の設定	43
4.7.1. クラスパスの設定	44
4.7.2. VMオプションの設定	45
4.8. 統計情報の取得	45
4.8.1. 統計情報の出力レベル設定	45
4.8.2. 統計情報の出力レベル確認	46
4.8.3. 統計情報の取得	47
4.8.4. Transactionサービス (JTA) 統計情報の取得	48
4.9. コード変換について	51
4.9.1. CORBA通信におけるコード変換	51

4.9.2.	Webアプリケーションの文字エンコーディングの設定について	52
4.10.	JCONSOLEを利用したJAVAプロセスの監視・管理	53
4.10.1.	WebOTXに必要な設定	53
4.10.2.	WebOTXで動作するJavaプロセスへの接続	56
4.10.3.	接続できない場合の対処	58
5.	リソースの追加・削除	59
6.	バックアップ/リストア	61
6.1.	ドメイン環境の自動バックアップ	61
6.2.	バックアップ手順	63
6.2.1.	Windows	63
6.2.2.	HP-UX版	63
6.2.3.	Solaris	64
6.2.4.	Linux	64
6.3.	同一環境にリストアする手順	64
6.3.1.	Windows	64
6.3.2.	HP-UX版	64
6.3.3.	Solaris	65
6.3.4.	Linux	65
6.4.	異なる環境にリストアする手順	65
6.5.	ドメイン構成ファイル(DOMAIN.XMLとDOMAINS-CONFIG.XML)の自動復旧機能	67

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す。

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. サービスの起動・停止

システムを構築、運用するための手順について説明します。

2.1. WebOTXのサービスについて

WebOTX のサービスについて説明します。

Windows:

Windows のサービスとして以下のサービスがあります。

- WebOTX Agent Service
WebOTX 全体を管理するエージェントサービスです。既定値では自動起動するように設定しています。
- WebOTX WebServer *domainname*
Apache HTTP Server 1.3 ベースの WebOTX Web サーバのサービスです。インストール時に「Web サーバ 1.3」を選択した場合に登録され、WebOTX Web サーバを利用するドメイン毎に作成されます。既定値では手動起動です。該当ドメイン起動時に自動的に起動します。サービスマネージャから起動する必要はありません。
- WebOTX WebServer2 *domainname*
Apache HTTP Server 2.0 ベースの WebOTX Web サーバのサービスです。インストール時に「Web サーバ 2.0」を選択した場合に登録され、WebOTX Web サーバを利用するドメイン毎に作成されます。既定値では手動起動です。該当ドメイン起動時に自動的に起動します。サービスマネージャから起動する必要はありません。
- ObjectSpinner
WebOTX としては利用しませんが、WebOTX 以外のモジュールがドメイン外で CORBA サービスとして利用する場合に利用します。既定値では手動起動で通常は起動しません。
- WebOTX TPBASEadm
WebOTX Standard Edition もしくは Enterprise Edition を利用する場合、TP モニタに対する制御を行なうサービスです。「WebOTX Agent Service」と依存関係があり、「WebOTX Agent Service」を起動すると自動的に起動されます。
- WebOTX WatchServer
Enterprise Edition を利用する場合、マルチサーバ構成の負荷分散構成とした場合に、アプリケーションサーバの監視を行なうサービスです。Enterprise Edition、もしくは StandardEdition とクラスタ環境を組み合わせることでできます。
- WebOTX Transaction Service
Transaction サービスです。WebOTX 以外のモジュールがドメイン外で Transaction サービス管理用プロセス(RCD)を使用する場合に起動します。初期状態ではサービス登録されていません。使用する場合は「wotsrpd -a」コマンドでサービスを登録する必要があります。既定値では手動起動で通常は起動しません。
- WebOTX TS RecoveryServer
Transaction サービスです。リカバリサーバを使用する場合に起動します。インストール時に「AP サーバ環境」 - 「TransactionService」 - 「旧バージョン互換サービス」を選択する必要があります。既定値では手動起動で通常は起動しません。

HP-UX:

HP-UX ではマシン起動時にサービスが起動できるよう以下の起動スクリプトを提供しています。ただし ObjectSpinner、WebOTX_TS に関しては Windows と同様マシン起動時には起動しません。

- /sbin/init.d/WOAgentSvc
- /sbin/rc1.d/K200WOAgentSvc (/sbin/init.d/WOAgentSvc へのシンボリックリンク)
- /sbin/rc2.d/S800WOAgentSvc (/sbin/init.d/WOAgentSvc へのシンボリックリンク)

- /sbin/init.d/ObjectSpinner
- /sbin/rc1.d/K250ObjectSpinner (/sbin/init.d/ObjectSpinner へのシンボリックリンク)
- /sbin/rc2.d/S750ObjectSpinner (/sbin/init.d/ObjectSpinner へのシンボリックリンク)
- /sbin/init.d/WebOTXinit
- /sbin/rc1.d/K215WebOTXinit (/sbin/init.d/WebOTXinit へのシンボリックリンク)
- /sbin/rc2.d/S790WebOTXinit (/sbin/init.d/WebOTXinit へのシンボリックリンク)
- /sbin/init.d/WatchServer
- /sbin/rc1.d/K210WatchServer (/sbin/init.d/WatchServer へのシンボリックリンク)
- /sbin/rc2.d/S795WatchServer (/sbin/init.d/WatchServer へのシンボリックリンク)
- /sbin/init.d/WebOTX_TS
- /sbin/rc1.d/K230WebOTX_TS (/sbin/init.d/WebOTX_TS へのシンボリックリンク)
- /sbin/rc2.d/S770WebOTX_TS (/sbin/init.d/WebOTX_TS へのシンボリックリンク)

Solaris:

Solaris ではマシン起動時にサービスが起動できるように以下の起動スクリプトを提供しています。ただし ObjectSpinner、WebOTX_TS に関しては Windows と同様マシン起動時には起動しません。

- /etc/init.d/WOAgentSvc
- /etc/rc0.d/K30WOAgentSvc
- /etc/rc1.d/K30WOAgentSvc
- /etc/rc2.d/S81WOAgentSvc
- /etc/init.d/ObjectSpinner
- /etc/rc0.d/K50ObjectSpinner
- /etc/rc1.d/K50ObjectSpinner
- /etc/rc2.d/S75ObjectSpinner
- /etc/init.d/WebOTXinit
- /etc/rc0.d/K35WebOTXinit
- /etc/rc1.d/K35WebOTXinit
- /etc/rc2.d/S79WebOTXinit
- /etc/init.d/WatchServer
- /etc/rc0.d/K33WatchServer
- /etc/rc1.d/K33WatchServer
- /etc/rc2.d/S80WatchServer
- /etc/init.d/WebOTX_TS
- /etc/rc0.d/K40WebOTX_TS
- /etc/rc1.d/K40WebOTX_TS
- /etc/rc2.d/S77WebOTX_TS

Linux:

Linux ではマシン起動時にサービスが起動できるように以下の起動スクリプトを提供しています。ただし ObjectSpinner、

WebOTX_TS に関しては Windows と同様マシン起動時には起動しません。

- /etc/init.d/WOAgentSvc
- /etc/rc0.d/K30WOAgentSvc
- /etc/rc1.d/K30WOAgentSvc
- /etc/rc2.d/S81WOAgentSvc
- /etc/rc3.d/S81WOAgentSvc
- /etc/rc5.d/S81WOAgentSvc
- /etc/rc6.d/K30WOAgentSvc
- /etc/init.d/ObjectSpinner
- /etc/rc0.d/K50ObjectSpinner
- /etc/rc1.d/K50ObjectSpinner
- /etc/rc2.d/S75ObjectSpinner
- /etc/rc3.d/S75ObjectSpinner
- /etc/rc4.d/S75ObjectSpinner
- /etc/rc5.d/S75ObjectSpinner
- /etc/rc6.d/K50ObjectSpinner
- /etc/init.d/WebOTXinit
- /etc/rc0.d/K35WebOTXinit
- /etc/rc1.d/K35WebOTXinit
- /etc/rc2.d/S79WebOTXinit
- /etc/rc3.d/S79WebOTXinit
- /etc/rc5.d/S79WebOTXinit
- /etc/init.d/WatchServer
- /etc/rc0.d/K33WatchServer
- /etc/rc1.d/K33WatchServer
- /etc/rc2.d/S80WatchServer
- /etc/rc3.d/S80WatchServer
- /etc/rc5.d/S80WatchServer
- /etc/init.d/WebOTX_TS
- /etc/rc0.d/K40WebOTX_TS
- /etc/rc1.d/K40WebOTX_TS
- /etc/rc2.d/S77WebOTX_TS
- /etc/rc3.d/S77WebOTX_TS
- /etc/rc4.d/S77WebOTX_TS
- /etc/rc5.d/S77WebOTX_TS
- /etc/rc6.d/K40WebOTX_TS

2.2. サービスの起動・停止方法

WebOTX のサービス全体を起動および停止する方法について説明します。デフォルトの設定ではマシン起動時にサービスが起動するようになっています。ObjectSpinner、WebOTX_TS のサービスは WebOTX としては利用しませんが、下記と同様の方法で起動・停止することができます。

注意: UNIX での起動停止の操作は root ユーザで実行してください。

Windows:

「WebOTX AS Agent Service」サービスを起動・停止します。サービスコントロールマネージャもしくは net start、net stop コマンドより実行できます。

例) サービスの起動

```
net start "WebOTX AS Agent Service"
```

例) サービスの停止

```
net stop "WebOTX AS Agent Service"
```

サービス起動直後はまだ起動処理が行われています。サービスの起動完了を確認する場合は「[ドメインの起動完了確認](#)」を参考にしてドメインの起動処理が完了していることを確認してください。

サービス停止時、ドメインの停止に時間がかかるとサービス制御マネージャがタイムアウトして「ローカルコンピュータの WebOTX Agent Service サービスを停止できません。」と出力されることがあります。

ただし、停止処理は続けられ、最終的には停止します。

HP-UX:

/sbin/init.d/WOAgentSvc に起動・停止スクリプトがあります。スクリプトを実行することによりサービスの起動、停止が行えます。

例) サービスの起動

```
/sbin/init.d/WOAgentSvc start
```

例) サービスの停止

```
/sbin/init.d/WOAgentSvc stop
```

Solaris:

/etc/init.d/WOAgentSvc に起動・停止スクリプトがあります。スクリプトを実行することによりサービスの起動、停止が行えます。

例) サービスの起動

```
/etc/init.d/WOAgentSvc start
```

例) サービスの起動

```
/etc/init.d/WOAgentSvc stop
```

Linux:

/etc/init.d/WOAgentSvc に起動・停止スクリプトがあります。スクリプトを実行することによりサービスの起動、停止が行えます。

例) サービスの起動

```
/etc/init.d/WOAgentSvc start
```

例) サービスの起動

/etc/init.d/WOAgentSvc stop

2.3. サービスの設定

Windows 版のサービスは、プロパティファイルを持っています。ここに記述されたコマンドを実行してドメイン起動を行っています。

`#{AS_INSTALL}¥config¥woasinfo.properties`

この設定を変更することにより、コマンド実行待ち時間を変更することが可能です。

項目	説明	既定値
startcmd	ドメイン起動コマンドを記述します。 記述がない、不正記述などの場合はサービス起動に失敗します。	"#{AS_INSTALL}¥bin¥WOAgentSvc.bat" start-domain
stopcmd	ドメイン停止コマンドを記述します。 記述がない、不正記述などの場合はサービス停止に失敗します。 停止時に設定ファイルを読み直します。	"#{AS_INSTALL}¥bin¥WOAgentSvc.bat" stop-domain
woastrace	詳細トレース(woastrace.log) 採取フラグです。 Agent サービス自体のトレースを出力するときに使用します。 1:採取する 0:採取しない(既定値) トレースファイルは以下の位置に出力されます。 <code>#{AS_INSTALL}¥bin¥woastrace.log</code> 設定は再起動時に有効になります。	0
woastracesize	詳細トレースのサイズをキロバイトで指定します。 このサイズを超過するか、サービス再起動時にバックアップファイル(woastrace.log.bak) を作成します。バックアップファイルがあった場合は上書きします。 設定は再起動時に有効になります。	32
woascmdtimeout	startcmd、stopcmd で指定したコマンド実行時のタイムアウト設定を秒単位で指定します。 -1 を設定すると非同期実行となり、コマンド終了を待ち合わせません。	600

3. システムに関する設定

システムに関する設定項目について説明します。

3.1. システム環境変数

WebOTX共通のシステム環境変数

WebOTX では動作環境についての情報を環境変数や Java システムプロパティとして保持しています。

WebOTXドメイン共通の環境変数は、以下の設定ファイルで定義されます。

Windows 版

```
${INSTALL_ROOT}\%config%\asenv.bat
```

UNIX 版

```
${INSTALL_ROOT}%config%asenv.conf
```

以下にその一覧を示します。

ドメイン共通の環境変数:

変数名	説明	既定値
AS_INSTALL	WebOTX インストールディレクトリ インストール時に指定する。	(Windows) (システムドライブ):\%WebOTX (UNIX) /opt/WebOTX
AS_ANT	ANT ディレクトリ	\$(AS_INSTALL)/lib/ant
AS_ANT_LIB	ANT ライブラリディレクトリ	\$(AS_ANT)/lib
AS_WEBSERVICES_LIB	WEB サービスライブラリディレクトリ	\$(AS_INSTALL)/lib
AS_IMQ_LIB	JMS ライブラリディレクトリ	\$(AS_INSTALL)/wojms/lib
AS_IMQ_BIN	JMS 実行ディレクトリ	\$(AS_INSTALL)/wojms/bin
AS_CONFIG	コンフィグディレクトリ	\$(AS_INSTALL)/config
AS_JAVA	JAVA インストールディレクトリ	\$(JAVA_HOME)
AS_ACC_CONFIG	アプリケーションクライアントが使用する XML ファイル	\$(AS_INSTALL)/lib/appclient/nec-acc.xml
AS_ICU_LIB	WebOTX 実行時に必要なネイティブライ ブラリディレクトリ	\$(AS_INSTALL)/lib
AS_OSPI_LIB	Object Broker ライブラリパス	\$(AS_INSTALL)/ObjectBroker/lib
AS_DEF_DOMAINS_PATH	domains ディレクトリ	\$(AS_INSTALL)/domains
AS_BOOT_DOMAIN	サーバ起動時に自動起動されるドメイン	マルチドメイン時: WebOTXAdmin

	名	シングルドメイン時: domain1
--	---	--------------------

ドメイン別の環境変数:

変数名	説明	既定値
DOMAIN_NAME	ドメイン名 ドメイン作成時に指定	domain1(インストール時作成されるドメイン)
SERVER_NAME	サーバ名 “server”で固定	server
INSTANCE_ROOT	ドメインのルートディレクトリ	\$(AS_INSTALL)/domains/\$(DOMAIN_NAME)

ユーザ定義の共通システム環境変数

任意の環境変数を設定する場合、または、システムで定義される環境変数を WebOTX のみ変更する必要がある場合には、設定ファイルの最後に環境変数の定義を追記します。Unix の場合、設定した環境変数をドメイン環境で利用する場合は export を行う必要がありますので注意してください。

Windows 版

```
set 環境変数=値
```

Unix 版

```
環境変数名=値
export 環境変数名
```

プロセスグループ個別のシステム環境変数

Standard Edition 及び、Enterprise Edition では、プロセスグループ個別にシステム環境変数を定義することも可能です。プロセスグループ個別に環境変数を定義する場合は、プロセスグループのプロパティ “環境変数” に設定します。

環境変数の影響範囲

WebOTX のシステム環境変数は、環境変数や Edition、Web コンテナのシングルプロセスモード/マルチプロセスモードの違いにより、アプリケーションへの影響範囲が異なります。

以下に WebOTX の設定とアプリケーションへの影響についてまとめます。

パス環境変数(Windows:PATH、HP-IPF/Linux/Solaris:LD_LIBRARY_PATH、HP(PA):SHLIB)

変数名	説明	製品名(Edition)	Web	EJB
PATH	OS の環境変数(運用ユーザの環境変数)に定義	全 Edition 共通	○	○
LD_LIBRARY_PATH	WebOTX の共通設定に定義	Web Edition、Standard-J Edition	○	○

		Standard Edition、Enterprise Edition (Web コンテナ シングルプロセスモード)	○	○
		Standard Edition、Enterprise Edition (Web コンテナ マルチプロセスモード)	○	○
ドメインのプロパティ ”native-library-path-prefix” 及び ”native-library-path-suffix” に 定義		Web Edition、Standard-J Edition	○	○
		Standard Edition、Enterprise Edition (Web コンテナ シングルプロセスモード)	○	×
		Standard Edition、Enterprise Edition (Web コンテナ マルチプロセスモード)	×	×
プロセスグループのプロパティ”環境 変数”に定義		Web Edition、Standard-J Edition	—	—
		Standard Edition、Enterprise Edition (Web コンテナ シングルプロセスモード)	×	○
		Standard Edition、Enterprise Edition (Web コンテナ マルチプロセスモード)	○	○

3.2. 使用するポート番号

管理ドメインで使用するポート番号

管理ドメインで使用するポート番号について以下に説明します。

ドメイン作成時のプロパティ名	既定値	説明
admdomain.admin.port	6202	エージェントが利用する JMXMP のポート番号

TPモニタ運用管理プロセスで使用するポート番号

Standard/Enterprise Edition を利用する場合、運用管理エージェントとTPモニタ運用管理プロセスが利用するポートの設定が必要です。このポートの設定は直接/etc/services に定義します。Windows の場合 WebOTX インストール時に自動的に設定します。UNIX の場合は、WebOTX をインストール後必ず設定してください。なおこのポート番号はローカルマシン間でのみ利用されます。

サービス名	既定値	説明
w-tpadm	8899	TPモニタ管理プロセスで使用するポート番号

一般ドメインで使用するポート番号

一般ドメインで使用するポート番号について以下に説明します。なお一般ドメインについてはサーバ上に複数作成することが出来、定義した全てのポート番号が利用されます。

ドメイン作成時のプロパティ名	既定値	説明
domain.admin.port	6212	エージェントが利用する JMXMP のポート番号

dottedname	domain1 での既定値	説明
server.embedded-iiop-service.port	7780	エージェント上で動作する IIOP リスナ。JNDI サーバと Standard-J Edition における EJB の通信リスナとなります。
server.http-service.http-listener.http-listener-1.port	80	HTTP サーバが利用する HTTP ポートの番号 HTTP サーバを起動した場合利用
server.http-service.http-listener.http-listener-2.port	443	SSL で保護された HTTPS ポートの番号 HTTP サーバを起動した場合利用
server.http-service.http-listener.admin-listener.port	4848	エージェントが利用する管理用ポートの番号 Web管理コンソールとの通信に利用
server.http-service.http-listener.ajp-listener-1.port	8009	外部の HTTP サーバ・Web コンテナ連携ポートの番号 HTTP サーバ、Web コンテナを起動した場合利用
server.objectbrokerservice.oad.OadPort	9825	oad が使用するポート番号 Object Broker サービスを起動した場合利用
server.objectbrokerservice.oadj.Port	9826	oadj が使用するポート番号 Object Broker サービスを起動した場合利用
server.objectbrokerservice.corbaloc.CorbalocServerPort	9827	corbaloc サーバが使用するポート番号 Object Broker サービスを起動した場合利用
server.objectbrokerservice.namesv.NameServicePort	2809	名前サーバが使用するポート番号 Object Broker サービスを起動した場合利用
server.objectbrokerservice.irsv.InterfaceRepositoryPort	9830	インタフェースリポジトリが使用するポート番号 Object Broker サービスを起動した場合利用
server.objectbrokerservice.cnamesv.CacheNameServicePort	9829	キャッシュ名前サーバが使用するポート番号 Object Broker サービスを起動した場合利用 (キャッシュ名前サーバは、Enterprise Edition で選択した場合のみインストールされます)
server.jms-service.jms-host.default_JMS_host.port	9700	JMS が使用するポート番号 JMS を起動した場合利用
server.transactionservice.rcs-cpp-port	5965	C++版の RCS が使用するポート番号 (Web Edition、および Standard-J Edition では使用されません)。
tpsystem.IIOPListener.listenerPortNumber	5151	IIOPリスナが使用するポート番号 (平文ポート) Standard/Enterprise Edition の場合で平文ポートを利用する場合利用

tpsystem.IIOPListener. sslPortNumberCert	5751	IIOPリスナが使用するポート番号(SSL クライアント認証ありポート) Standard/Enterprise Edition の場合でSSLクライアント認証ありを利用する場合利用
tpsystem.IIOPListener. sslPortNumberNoCert	5751	IIOPリスナが使用するポート番号(SSL クライアント認証なしポート) Standard/Enterprise Edition の場合でSSLクライアント認証なしを利用する場合利用
tpsystem.OLFTPLListener. listenerPortNumber	5251	OLF/TP リスナが使用するポート番号 Standard/Enterprise Edition の場合利用
tpsystem.OLFTPLListener. listenerPortNumber	5251	OLF/TP リスナが使用するポート番号 Standard/Enterprise Edition の場合利用
tpsystem.admListener.portNumber	5202	Adm リスナが使用するポート番号 Standard/Enterprise Edition の場合利用
server.http-service.sip-listener. sip-listener-1.port	5060	SIP リスナが使用するポート番号 SIP Application Server の場合利用
server.http-service.sip-listener. sip-listener-2.port	5060	SIP リスナが使用するポート番号 SIP Application Server の場合利用
server.http-service.sip-listener. sip-listener-3.port	5061	SIP TLS リスナが使用するポート番号 SIP Application Server の場合利用

自動で割り当てられる一時ポート番号に関する注意制限事項

OS により以下のポート番号の範囲が自動で割り当てられる一時ポート番号として設定されています。

OS	一時ポートの割り当て範囲(デフォルト)
Windows	1024 から 5000
HP-UX	49152 から 65535
Linux	32768 から 61000
Solaris	32768 から 65535

WebOTX が使用するポート番号が一時ポートの割り当て範囲と重複していた場合、別のアプリケーションが同じポート番号を一時ポートとして使用し WebOTX の起動が失敗するなどの問題を引き起こすことがあります。

Windows 2000/2003 においては、ポート番号 1024~5000 の範囲となり、WebOTX が既定値として使用する、名前サーバのポート(2809)と運用管理用ポート(4848)が該当します。

対処法としては、WebOTX で利用するポートを OS の一時ポートの割り当て対象からはずします。設定方法として以下の手順に従い、設定用のバッチスクリプトを実行してください。

1. 設定用コマンドによる登録

コマンドプロンプトを起動し、以下のコマンドを実行します。

```
> ${INSTANCE_ROOT}\%bin%\ReservPorts.bat install
Microsoft (R) Windows Script Host Version 5.6
Copyright (C) Microsoft Corporation 1996-2001. All rights reserved.
Add ReservPorts :[2809-2809 4848-4848 ]
ReservPorts registerd Succeed! Please reboot the machine.
```

一時ポート範囲から除外されたポート番号が、Add ReservPorts:[開始ポート番号-終了ポート番号 ...]として表示されます。設定変更後はマシンの再起動が必要です。

ReservedPort.bat コマンド: \${INSTANCE_ROOT}\%bin%\ReservPorts.bat [list | install | uninstall]

表示項目	説明
Reserved Ports :[ポート番号 ...]	割り当て対象外として登録されているポート番号 (現在のレジストリ値)
WebOTX Ports :[ポート番号 ...]	WebOTX が利用する 1024 から 5000 までのポート番号
Add Ports :[ポート番号 ...]	ReservPort コマンドにより登録された割り当て対象外のポート番号

2. レジストリエディタによる登録方法

レジストリエディタを起動します。

レジストリキー

HKEY_LOCAL_MACHINE\SYSTEM\

CurrentControlSet\Services\Tcpip\Parameters 配下に以下の項目を追加します。

名前: ReservedPorts

種類: REG_MULTI_SZ (複数行文字列値)

値: 以下を 2 行で入力してください

2809-2809

4848-4848

設定変更後はマシンの再起動が必要です。

3.3. 動作プロセスについて

サービスプロセス

プロセス名	説明
WOAgentSvc.exe (Windows)	サービス管理用のプロセスです。 「WebOTX Agent Service」を起動すると動作し、停止すると終了します。
tpadmd.exe (Windows) tpadmd (UNIX)	Standard/Enterprise Edition における TP モニタ運用管理用のサービスプロセスです。 「WebOTX TPBASEadm」を起動すると動作し、停止すると終了します。 UNIX におけるカレントディレクトリは \${AS_INSTALL}/Trnsv です。

管理ドメインで動作するプロセス

プロセス名	説明
java.exe(Windows) java(UNIX)	<p>エージェント(管理ドメイン)を起動するための JavaVM です。エージェントが停止すると終了します。</p> <p>異常終了した場合、Windows 系の場合はエージェント(管理ドメイン)の JavaVM に直接影響はありませんが、エージェントの JavaVM から直接出力されるエラーメッセージが server.log に出力されなくなります。UNIX 系の場合はエージェントの JavaVM(後述)、および一般ドメインにおける全てのプロセスが終了します。本プロセスを再起動するにはドメインの再起動が必要です。</p> <p>引数に「-Dwebotx.funcid=startserv -Ddomain.name=WebOTXAdmin」という文字列が指定されますのでそれが目印になります。</p>
javaw.exe(Windows) java(UNIX)	<p>エージェントの JavaVM です。</p> <p>異常終了した場合、管理ドメインへのアクセスができなくなりますが、一般ドメインは利用できます。復旧は WebOTX サービス再起動を行う必要があります。なお再起動方法については「2.2 サービスの起動・停止方法」を参照ください。</p> <p>引数に「-Dwebotx.funcid=agent -Ddomain.name=WebOTXAdmin」という文字列が指定されますのでそれが目印になります。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/config です。</p>

一般ドメインで動作するプロセス

ドメインを起動すると以下のプロセスが起動します。複数ドメインを起動した場合は、同一名のプロセスが複数起動します。

プロセス名	説明
java.exe(Windows) java(UNIX)	<p>エージェント(ドメイン)を起動するための JavaVM です。エージェントが停止すると終了します。</p> <p>異常終了した場合、Windows 系の場合はエージェント(ドメイン)の JavaVM に直接影響はありませんが、エージェントの JavaVM から直接出力されるエラーメッセージが server.log に出力されなくなります。UNIX 系の場合はエージェントの JavaVM(後述)も終了します。本プロセスを再起動するにはドメインの再起動が必要です。</p> <p>引数に「-Dwebotx.funcid=startserv -Ddomain.name=\${DOMAIN_NAME}」という文字列が指定されますのでそれが目印になります。</p>
javaw.exe(Windows) java(UNIX)	<p>エージェントの JavaVM です。</p> <p>異常終了した場合、該当ドメインへのアクセスができなくなります。速やかにドメイン再起動を行う必要があります。なお再起動方法については「3.3 ドメインの起</p>

	<p>動・停止」を参照ください。</p> <p>引数に「-Dwebotx.funcid=agent -Ddomain.name=\${DOMAIN_NAME}」という文字列が指定されますのでそれが目印になります。UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/config です。</p>
<p>apache.exe(Windows)</p> <p>httpd(UNIX)</p>	<p>HTTP サーバのデーモンプロセス。インストール時に「WebOTX Web サーバ」を選択した場合に有効です。Web サーバを起動すると動作します。</p> <p>常時親プロセス(監視プロセス)と子プロセス(HTTP サービスデーモン)で構成されており、子プロセスが異常終了した場合、親プロセスは子プロセスを再起動します。</p> <p>異常終了時は HTTP サーバの再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/です。</p>
<p>wojmsbrokersvc.exe (Windows)</p> <p>wojmsbrokerd (UNIX)</p>	<p>JMS 管理プロセス。</p> <p>JMS を起動すると動作します。</p> <p>異常終了時は JMS の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p>
<p>java.exe (Windows)</p> <p>java (UNIX)</p>	<p>JMS デーモンプロセス。</p> <p>JMS を起動すると動作します。</p> <p>異常終了時は JMS の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>引数に「-Dwebotx.funcid=jms -Ddomain.name=\${DOMAIN_NAME}」という文字列が指定されますのでそれが目印になります。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/config です。</p>
<p>oad.exe (Windows)</p> <p>oad (UNIX)</p>	<p>CORBA オブジェクト活性化デーモンプロセス。</p> <p>Object Broker を起動すると動作します。異常終了時は新たな IIOP 通信(RMI/IIOP)が行えなくなります。</p> <p>異常終了時は Object Broker の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/config です。</p> <p>Standard-J/Standard/Enterprise Edition で動作します</p>
<p>namesv.exe (Windows)</p> <p>namesv (UNIX)</p>	<p>CORBA 名前サーバデーモンプロセス。</p> <p>Object Broker を起動すると動作します。異常終了時はオブジェクトの取得が行えなくなり IIOP 通信ができなくなります。</p> <p>異常終了時は Object Broker の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/config です。</p>

	Standard-J/Standard/Enterprise Edition で動作します
irsv.exe (Windows) irsv (UNIX)	<p>CORBA インターフェースリポジトリデーモンプロセス。 Object Broker を起動すると動作します。異常終了時はオブジェクトのインタフェース情報の取得が行えなくなりますが WebOTX では irsv を利用していないため影響はありません。</p> <p>異常終了時は Object Broker の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>Standard-J/Standard/Enterprise Edition で動作します</p>
corbaloc.exe (Windows) corbaloc (UNIX)	<p>CORBALOC サーバデーモンプロセス。 Object Broker を起動すると動作します。異常終了時は CORBALOC サーバとして利用している場合、新たな IIOP 通信(RMI/IIOP)が行えなくなります。</p> <p>異常終了時は Object Broker の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>Standard-J/Standard/Enterprise Edition で動作します</p>
java.exe (Windows) java (UNIX)	<p>CORBA Java 自動起動デーモンプロセス。 Object Broker を起動すると動作します。異常終了時は Java アプリケーションの自動起動ができなくなります。また、ライセンスの取得もできなくなるため、IIOP 通信のコネクション数に制限が発生します。</p> <p>異常終了時は Object Broker の再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>引数に「-Dwebotx.funcid=oadj -Ddomain.name=\${DOMAIN_NAME}」という文字列が指定されますのでそれが目印になります。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/config です。</p> <p>Standard-J/Standard/Enterprise Edition で動作します</p>
rcssv.exe (Windows) rcssv (UNIX)	<p>Transaction サービスで提供する C++アプリケーション用のリカバリプロセス (RCS) (Web Edition、および Standard-J Edition では使用しません)</p> <p>Transaction サービスを起動すると動作します。異常終了時はトランザクションの開始が行えなくなります。</p> <p>異常終了時は Transaction サービスの再起動が必要です。なお再起動方法については「3.5 各サービスの起動・停止」を参照ください。</p> <p>引数に「rcsid=XXX」という文字列が指定されますのでそれが目印になります。</p> <p>UNIX におけるカレントディレクトリは\${INSTANCE_ROOT}/logs/TS です。</p>
tpmMain.exe (Windows) tpmMain (UNIX)	<p>アプリケーションプロセスの障害検出や異常終了時の自動復旧など、WebOTX の高信頼性を実現させているプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了した場合 TP モニタの機能が利用できなくなりますが、親プロセスが監</p>

	<p>視を行なっているのでプロセスの監視は親プロセス(tpmmonitor)の方を監視してください。</p> <p>UNIX におけるカレントディレクトリは\${AS_INSTALL}/Trnsv です。</p>
<p>tpmonitor.exe (Windows)</p> <p>tpmonitor (UNIX)</p>	<p>tpmMain プロセスの親プロセスで主な役割は tpmMain プロセスの監視です。Standard/Enterprise Edition で動作します。</p> <p>異常終了した場合、TP モニタの機能が利用できなくなるため、プロセス監視を行なう場合は、監視対象にしてください。</p> <p>復旧方法は TP システムの再起動です。それでも復旧しない場合はマシン再起動を行なってください。</p> <p>UNIX におけるカレントディレクトリは\${AS_INSTALL}/Trnsv です。</p>
<p>tpadmdMain.exe (Windows)</p> <p>tpadmd (UNIX)</p>	<p>WebOTX エージェントプロセス(javaw)からの要求を受けるなど、TP モニタの運用操作に必要なプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了した場合、一時的に Standard/Enterprise Edition の一部運用操作がエラーとなりますが、自動的にプロセスが再起動され復旧します。</p> <p>UNIX におけるカレントディレクトリは\${AS_INSTALL}/Trnsv です。</p>
<p>tpssendtp.exe (Windows)</p> <p>tpssendtp (UNIX)</p>	<p>クライアントへのメッセージ送信等を行うプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了した場合、統合運用管理ツールからクライアントへメッセージ送信や動的ログレベル変更などが不可となります。復旧方法は TP システムの再起動です。</p>
<p>systpp.exe (Windows)</p> <p>systpp (UNIX)</p>	<p>アプリケーショングループの起動や停止などを実行するためのプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了した場合、一部運用操作(起動、停止など)が不可となります。復旧方法は TP システムの再起動です。</p>
<p>jnlwrt.exe (Windows)</p> <p>jnlwrt (UNIX)</p>	<p>ジャーナルを収集しているプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了した場合、異常終了した場合、ジャーナルの採取が不可となります。復旧方法は TP システムの再起動です。</p>
<p>olftplsn.exe (Windows)</p> <p>olftplsn (UNIX)</p>	<p>WebOTX Print Kit を利用する場合に必要なプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了時は OLF リスナとの通信が不可となり、WebOTX Print Kit が使用できなくなります。復旧方法は TP システムの再起動です。</p>
<p>iioplsn.exe (Windows)</p> <p>iioplsn (UNIX)</p>	<p>クライアントとの接続切断や、電文の送受信を行っているプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了時はIIOPリスナとの通信が不可となります。全てのクライアントからの</p>

	<p>アクセスができなくなります。復旧方法は TP システムの再起動です。</p> <p>UNIX におけるカレントディレクトリは\${AS_INSTALL}/Trnsv /logs です。</p>																																																			
<p>TIMMSGSEND.exe(Windows)</p> <p>TIMMSGSEND(UNIX)</p>	<p>メッセージ送信を行う時間の管理を行っています。送信時間になるとメッセージを送信するように tpssendtp プロセスに指示します。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了時はクライアントへメッセージ送信や動的ログレベル変更などが不可となります。復旧方法は TP システムの再起動です。</p>																																																			
<p>wosystpp.exe(Windows)</p> <p>wosystpp(UNIX)</p>	<p>トレースレベルとサイズの変更、モジュール動的追加と削除、サーバプロセスメッセージ通知、モジュールの活性化と非活性化、スタックトレースの採取を行うためのプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>異常終了時は上記機能が使用できなくなります。復旧方法は TP システムの再起動です。</p>																																																			
<p>THTPPCTL.exe</p> <p>THTPPCTL_N.exe</p> <p>THTPPCTL_N2003.exe</p> <p>THTPPCTL_2005.exe</p> <p>(Windows)</p> <p>THTPPCTL7</p> <p>THTPPCTL8</p> <p>THTPPCTL9</p> <p>THTPPJAVA2</p> <p>THTPPOTS7</p> <p>THTPPOTS8</p> <p>THTPPOTS9</p> <p>THTPPDB7</p> <p>THTPPDB8</p> <p>THTPPDB9</p> <p>(UNIX)</p>	<p>サーバアプリケーションのプロセスです。</p> <p>Standard/Enterprise Edition で動作します。</p> <p>言語や利用機能によりモジュール名が変わります。</p> <p>(Windows)</p> <table border="1"> <thead> <tr> <th>モジュール名</th> <th>言語、利用機能</th> <th>バージョン</th> </tr> </thead> <tbody> <tr> <td>THTPPCTL.exe</td> <td>Java</td> <td>全バージョン</td> </tr> <tr> <td></td> <td>C++(VC6.0)</td> <td>V5</td> </tr> <tr> <td>THTPPCTL_N.exe</td> <td>C++(VC .net 2002)</td> <td>V5</td> </tr> <tr> <td>THTPPCTL_N2003.exe</td> <td>C++(VC .net 2003)</td> <td>全バージョン</td> </tr> <tr> <td>THTPPCTL_2005.exe</td> <td>C++(VC 2005)</td> <td>V6 以上</td> </tr> </tbody> </table> <p>(UNIX)</p> <table border="1"> <thead> <tr> <th>モジュール名</th> <th>言語、利用機能</th> <th>バージョン</th> </tr> </thead> <tbody> <tr> <td>THTPPCTL7</td> <td>C++</td> <td>V5</td> </tr> <tr> <td>THTPPCTL8</td> <td>C++</td> <td>V6</td> </tr> <tr> <td>THTPPCTL9</td> <td>C++</td> <td>V7</td> </tr> <tr> <td>THTPPJAVA2</td> <td>Java</td> <td>全バージョン</td> </tr> <tr> <td>THTPPOTS7</td> <td>C++(TS 連携あり)</td> <td>V5</td> </tr> <tr> <td>THTPPOTS8</td> <td>C++(TS 連携あり)</td> <td>V6</td> </tr> <tr> <td>THTPPOTS9</td> <td>C++(TS 連携あり)</td> <td>V7</td> </tr> <tr> <td>THTPPDB7</td> <td>C++(DB連携あり)</td> <td>V5</td> </tr> <tr> <td>THTPPDB8</td> <td>C++(DB連携あり)</td> <td>V6</td> </tr> <tr> <td>THTPPDB9</td> <td>C++(DB連携あり)</td> <td>V7</td> </tr> </tbody> </table> <p>異常終了時はそのプロセスグループが機能しなくなります。ただし、マルチプロセス構成や再起動設定により、自動的に復旧されます。</p> <p>UNIX におけるカレントディレクトリは\${AS_INSTALL}/Trnsv です。</p>	モジュール名	言語、利用機能	バージョン	THTPPCTL.exe	Java	全バージョン		C++(VC6.0)	V5	THTPPCTL_N.exe	C++(VC .net 2002)	V5	THTPPCTL_N2003.exe	C++(VC .net 2003)	全バージョン	THTPPCTL_2005.exe	C++(VC 2005)	V6 以上	モジュール名	言語、利用機能	バージョン	THTPPCTL7	C++	V5	THTPPCTL8	C++	V6	THTPPCTL9	C++	V7	THTPPJAVA2	Java	全バージョン	THTPPOTS7	C++(TS 連携あり)	V5	THTPPOTS8	C++(TS 連携あり)	V6	THTPPOTS9	C++(TS 連携あり)	V7	THTPPDB7	C++(DB連携あり)	V5	THTPPDB8	C++(DB連携あり)	V6	THTPPDB9	C++(DB連携あり)	V7
モジュール名	言語、利用機能	バージョン																																																		
THTPPCTL.exe	Java	全バージョン																																																		
	C++(VC6.0)	V5																																																		
THTPPCTL_N.exe	C++(VC .net 2002)	V5																																																		
THTPPCTL_N2003.exe	C++(VC .net 2003)	全バージョン																																																		
THTPPCTL_2005.exe	C++(VC 2005)	V6 以上																																																		
モジュール名	言語、利用機能	バージョン																																																		
THTPPCTL7	C++	V5																																																		
THTPPCTL8	C++	V6																																																		
THTPPCTL9	C++	V7																																																		
THTPPJAVA2	Java	全バージョン																																																		
THTPPOTS7	C++(TS 連携あり)	V5																																																		
THTPPOTS8	C++(TS 連携あり)	V6																																																		
THTPPOTS9	C++(TS 連携あり)	V7																																																		
THTPPDB7	C++(DB連携あり)	V5																																																		
THTPPDB8	C++(DB連携あり)	V6																																																		
THTPPDB9	C++(DB連携あり)	V7																																																		

単体で動作するプロセス

ツールなど単独で動作可能なプロセスです。複数起動した場合は、同一名のプロセスが複数起動します。

プロセス名	説明
java.exe(Windows) java(UNIX)	運用管理コマンド(otxadmin)のプロセスです。 引数に「-Dwebotx.funcid=otxadmin」という文字列が指定されますのでそれが目印になります。
javaw.exe(Windows) javaw(UNIX)	統合運用管理ツール(otxadmingui)のプロセスです。 引数に「-Dwebotx.funcid=otxadmingui」という文字列が指定されますのでそれが目印になります。
javaw.exe(Windows) javaw(UNIX)	配備ツール(deploytool)のプロセスです。 引数に「-Dwebotx.funcid=deploytool」という文字列が指定されますのでそれが目印になります。
javaw.exe(Windows) javaw(UNIX)	JNDI管理ツール(jndiadm)のプロセスです。 引数に「-Dwebotx.funcid=jndiadm」という文字列が指定されますのでそれが目印になります。

3.4. ディレクトリ構成

インストールディレクトリのディレクトリ構成

WebOTX をインストールしたディレクトリ($\${AS_INSTALL}$)のディレクトリ構成について説明します。

以下にそれぞれのディレクトリについて説明します。

ディレクトリ名	説明
bin	WebOTX エージェントのバイナリファイル、コマンド、バッチファイルを格納します。
config	WebOTX 全体に関する構成情報ファイルを格納します。
domains	ドメインのルートディレクトリの親ディレクトリです。ドメインを新規に作成するとこの下にドメインルートが作成されます。
downloader(*1)	Downloader のモジュールに関するファイルを格納しています。
jbi(*3)	Enterprise Service Bus に関するモジュールを格納しています。
lib	WebOTX エージェントが利用するライブラリやツール、各ドメイン作成時に利用するテンプレートファイルを格納しています。
ObjectBroker	Object Broker のモジュールに関するファイルを格納しています。
ProcessConductor(*3)	Process Conductor のモジュールに関するファイルを格納しています。
samples(*1)	サンプルアプリケーションを格納しています。
share(*2)	ライセンスチェックのモジュールに関するファイルを格納しています。
template(*1)	画面テンプレートのモジュールに関するファイルを格納しています。
Trnsv	WebOTX 実行環境のモジュールに関するファイルを格納しています。
TS	Transaction サービスのモジュールに関するファイルを格納しています。
uddi(*3)	UDDI Registry のモジュールに関するファイルを格納しています。
WebCont5.0	Web コンテナのモジュールに関するファイルを格納しています。

WebServer(*1)	WebServer(apache1.3)に関するモジュールを格納しています。
WebServer2(*1)	WebServer(apache2.0)に関するモジュールを格納しています。
wojms	JMS のモジュールに関するファイルを格納しています。
wowdc(*3)	Work Load Manager に関するモジュールを格納しています。
WOWS(*1)	WatchServer に関するモジュールを格納しています。

*1: インストール時にオプションとして選択した場合にインストールされます。

*2: UNIX 環境の場合は、/opt/share.nec にインストールされます。

*3: WebOTX 拡張製品をインストールした場合にインストールされます。

ドメインルートのディレクトリ構成

ドメインのルートディレクトリ(\${INSTANCE_ROOT})のディレクトリ構成について説明します。

以下にそれぞれのディレクトリについて説明します。

ディレクトリ名	説明
applications	ドメインに配備されたアプリケーションを格納します。
autodeploy	ドメインのオートデプロイディレクトリ
backup	構成情報(config ディレクトリ)のバックアップを格納します。
bin	ドメイン固有のバッチファイルを格納します。
config	ドメインの設定情報ファイルを格納します。
docroot	HTTP サーバのドキュメントルート
generated	配備のときにコンテナで生成したファイルを格納します。
jbi	該当ドメインで動作する ESB に関するファイルを格納します。
lib	ドメイン固有のライブラリを格納します。
logs	ドメインのログディレクトリ
session-store	Passivate された stateful session bean の情報を格納します。
stats	ドメインの統計情報を格納します。
wojms	該当ドメインで動作する JMS に関するファイルを格納します。

3.5. Java J2SEの設定

WebOTXで利用する J2SE 環境は、インストール時に指定した J2SE のインストールディレクトリ、または、指定されたインストール済みの J2SE ディレクトリを利用して起動するように設定されます。

Windows 版

`${INSTALL_ROOT}%config%asenv.bat` に記述されている AS_JAVA

HP-UX,Solaris, Linux 版

`${INSTALL_ROOT}/config/asenv.conf` に記述されている AS_JAVA

Java J2SEの更新・変更

J2SE のリビジョンアップなどにより J2SE のインストールディレクトリが変更になった場合は、あわせて WebOTX の設定を変更する環境にあわせて変更する必要があります。

Windows 版

1.環境変数設定ファイルの変更

`${INSTALL_ROOT}\%config%\asenv.bat` に記述されている `AS_JAVA` の設定を SDK の環境に合わせて変更します。
`set AS_JAVA=C:\%j2sdk1.4.2_14`

2.JMS 環境設定ファイルの変更(Standard-J Edition 以上)

`${INSTALL_ROOT}\%wojms%\var%\jdk-env.bat` に記述されている `WOJMS_JAVAHOME` の設定を SDK の環境に合わせて変更します。
`set WOJMS_JAVAHOME=C:\%j2sdk1.4.2_14`

3.Java CurrentVersion の確認

WebOTX では、レジストリに登録されている Java の `CurrentVersion` を参照します。

```
HKEY_LOCAL_MACHINE
SOFTWARE
  JavaSoft
    Java Development Kit
```

`CurrentVersion` に正しく SDK のバージョンが設定されていることを確認します。

注意) 同一の端末に、複数バージョンの Java を同時にインストールしないようにしてください。複数のバージョンを同時に使用する場合は、Java `CurrentVersion` が正しくなるよう、WebOTX で使用する Java SDK バージョンを最後にインストールするようにしてください。

HP-UX,Solaris, Linux 版

1.環境変数設定ファイルの変更

`${INSTALL_ROOT}/config/asenv.conf` に記述されている `AS_JAVA` の設定を SDK の環境に合わせて変更します。
`AS_JAVA="/opt/java1.4"`

2.JMS 環境設定ファイルの変更(Standard-J Edition 以上)

`${INSTALL_ROOT}/wojms/var/jdk.env` に記述されている `WOJMS_JAVAHOME` の設定を SDK の環境に合わせて変更します。
`/opt/java1.4/jre`

J2SE 6 へのバージョンアップ設定

Java J2SE1.4 及び J2SE5.0 でインストールを行った WebOTX 環境を J2SE6.0 へバージョンアップする際、WebOTX の設定を変更する必要があります。

1.ドメインに Java システムプロパティ-Dsun.lang.ClassLoader.allowArraySyntax を追加

例)

```
otxadmin> create-jvm-options -Dsun.lang.ClassLoader.allowArraySyntax=true
```

2.JAX-WS の有効化

JAX-WS を有効にする必要があります。以下のコマンドを実行して JAX-WS を有効にしてください。

```
Windows 版
${INSTALL_ROOT}\lib\jaxws\install_jaxws.bat
UNIX 版
${INSTALL_ROOT}/lib/jaxws/install_jaxws.sh
```

3.ドメインに Java システムプロパティ-Djavax.xml.ws.spi.Provider を追加

```
例)
otxadmin> create-jvm-options -Djavax.xml.ws.spi.Provider=com.nec.webotx.webservice.xml.ws.spi.ProviderImpl
```

Standard/Enterprise Edition の場合は、各プロセスグループの Java システムプロパティにも追加する必要があります。

```
例)
otxadmin> add-pg-javasystem-property --apgroup apgroupname1 --javasystemprop sun.lang.ClassLoader.allowArraySyntax
--value true pggroupname1
otxadmin> add-pg-javasystem-property --apgroup apgroupname1 --javasystemprop javax.xml.ws.spi.Provider --value
com.nec.webotx.webservice.xml.ws.spi.ProviderImpl pggroupname1
```

3.6. WebOTX管理ユーザ設定 (HP-UX,Solaris,Linux)

3.6.1. WebOTX管理ユーザ(運用ユーザ)の設定

UNIX 系の OS では、WebOTX のインストール時にシステム管理者権限をもつ root ユーザとは別に、WebOTX のユーザを指定してインストールが可能です。WebOTX 管理ユーザ(運用ユーザ)を設定することで、不用意なシステム権限領域へのアクセスを制御し、より強固なセキュリティの確保と OS のもつユーザ制限機能を利用することができます。

また、運用ユーザを変更するスクリプトを提供していますので、インストール後でも運用ユーザに変更することが可能です。

ただし、WebOTX 管理ユーザを利用する場合には、OS 側の制限により、1024 以下のポート番号を利用することができなくなります。HTTP/HTTPS 用のポート番号を 1024 以下で設定している場合には、ポート番号の再設定等が必要になるため、注意が必要です。

なお、UNIX 系 OS における運用ユーザのパスワード変更による WebOTX への影響はありません。Windows OS の場合は「2.8.Windows サービスアカウントの変更」を参照してください。

以下では、インストール後に運用ユーザを変更する手順を示します。

- 1.root ユーザでログインします。
- 2.WebOTX を停止します。
otxadmin.bat stop-domain WebOTXAdmin
- 3.WebOTX の WebOTX 運用ユーザのユーザ、および、グループアカウントを作成します。ここでは、説明の為に、ユーザ名:"admin" グループ名:"webotx"として設定します。アカウントの作成はシステム管理者に依頼するか、各OSの運用マニュアルを参照して登録を行ってください。
- 4.新しく作成したアカウントが telnet などで正しくログインできることを確認してください。
- 5.root ユーザで以下のコマンドを実行します。

```
# cd /opt/WebOTX/bin
# ./otxown.sh
## Executing WebOTX Operation User Change script.
Would you like to change as WebOTX Operation User? [y, n](default n)
y
Please input WebOTX Operation User name.
```



```

admin
Please input WebOTX Operation User group name.
webotx
*****
* Change of WebOTX Operation User. To continue, input y. *
* Input q to exit the installation. [y, q](default y) *
*****

```

6. 設定がおわりましたら、/opt/WebOTX 配下のファイルオーナーが運用ユーザに設定されていることを確認します。
7. サーバを再起動してください。

3.6.2. 複数のドメインを異なるユーザで運用するための手順

Standard/Enterprise Edition(HP-UX / HP-UX[IPF] / Solaris / Linux) では、複数のドメインを異なるユーザで運用する機能をサポートしています。ここではそのための手順を説明します。

ドメインごとに運用ユーザを変えたい時はこちらの設定をご確認ください。

設定の際の注意点

設定の際は、以下の点に注意してください。

1. 各ドメインに設定するユーザは、すべてのドメイン間で同じグループに属している必要があります。
2. 管理ドメイン「WebOTXAdmin」は、root でなければなりません。
3. WebOTX 内部のソケット通信ポートは、w-tpadm(既定値: 8899)で設定したものを使用します。別ユーザで運用する場合は、各ドメイン用にシステム ID を加えたポートも設定する必要があります。サーバマシン内で使用されていないポート番号となるようにシステム ID を設定してください。
4. システム ID はドメイン間で一意である必要があります。
5. ファイルの直接編集やディレクトリ等のオーナー・権限等の変更は、WebOTX が停止した状態で行ってください。
6. ファイルを直接編集する際は、必ずバックアップを取ってください。

設定の手順

次の節から、以下の環境を例にして設定方法を説明します。

ドメイン種別	名前、値		
管理ドメイン	WebOTXAdmin		
一般ドメイン	domain1		
	システム ID		0
	運用ユーザ	オーナー	otxadm
		グループ	otxadmin
一般ドメイン	domain2		
	システム ID		2

	運用ユーザ	オーナー	otxadm2
		グループ	otxadmin

OS設定上の変更

ここでは、OS に対してポートの設定やサービス登録を行います。

1. /etc/services への設定追加

サービス名とポートの追加を行います。ポート番号には、「w-tpadm(既定値: 8899)+システム ID」の数値を設定します。

w-tpadm	8899/tcp	# WebOTX TPBASEadm(SV)
w-tpadm2	8901/tcp	# WebOTX TPBASEadm(SV)

ここでは、今まで設定されていた w-tpadm の他に、domain2 用として w-tpadm にシステム ID 2 を追加した値を設定しています。サービス名とポート番号は、他と重複しないように設定してください。

2. inetd へのサービス登録

ドメインごとに inetd の設定を行います。ここで指定したドメインのオーナーとグループがドメイン起動時のプロセスに反映されます。

HP-UX の場合

/etc/inetd.conf ファイルにエディタ等で以下の 2 行を追加して下さい。それぞれ一行ずつで記述して下さい。

```
w-tpadm stream tcp6 nowait root /opt/WebOTX/Trnsv/bin/tpadm.sh
w-tpadmd /opt/WebOTX/domains/domain1

w-tpadm2 stream tcp6 nowait root /opt/WebOTX/Trnsv/bin/tpadm.sh
w-tpadmd /opt/WebOTX/domains/domain2
```

第 1 フィールドには、/etc/services に設定したサービス名で inetd に追加します。さらに、サーバ・プログラム引数フィールドは、ドメインのディレクトリを与えます。

編集後は設定を再読み込みさせてください。

```
# /usr/sbin/inetd -c
```

Solaris の場合

/etc/inet/inetd.conf ファイルにエディタ等で以下の 2 行を追加して下さい。それぞれ一行ずつで記述して下さい。

```
w-tpadm stream tcp nowait root /opt/WebOTX/Trnsv/bin/tpadm.sh
w-tpadmd /opt/WebOTX/domains/domain1

w-tpadm2 stream tcp nowait root /opt/WebOTX/Trnsv/bin/tpadm.sh
w-tpadmd /opt/WebOTX/domains/domain2
```

第 1 フィールドには、/etc/services に設定したサービス名で inetd に追加します。さらに、サーバ・プログラム引数フィールドは、ドメインのディレクトリを与えます。

編集後は設定を再読み込みさせてください。

```
# kill -HUP <inetd のプロセス ID>
```

Linux の場合

インストール時に xinetd へのサービスを一つ登録しています。そのサービスに対して引数を設定します。また、新規にサービスを追加します。

/etc/xinetd.d/webotx ファイルをエディタ等で以下のように修正して下さい。

```
service w-tpadm
{
    socket_type    = stream
    protocol      = tcp
    wait          = no
    user          = root
    server        = /opt/WebOTX/Trnsv/bin/tpadm.sh
    server_args    = /opt/WebOTX/domains/domain1
}

service w-tpadm2
{
    socket_type    = stream
    protocol      = tcp
    wait          = no
    user          = root
    server        = /opt/WebOTX/Trnsv/bin/tpadm.sh
    server_args    = /opt/WebOTX/domains/domain2
}
```

インストール時に設定されているサービス w-tpadm に server_args を追加し、ドメインのディレクトリを設定します。また、/etc/services に新たに設定したサービスとその設定を inetd に追加します。

編集後は設定を再読み込みさせてください。

```
# kill -USR2 <xinetd のプロセス ID>
```

WebOTX制御ファイルの変更

ここでは、WebOTX の制御ファイルに関して、値の変更や権限の変更を行います。

1. /opt/WebOTX/Trnsv/bin/tpadm.sh

tpadm.sh 内の記述において、「wosh」に引数を追加するため「\$1」を書き加えます。\$1 の前に半角スペースが必要です。

```
#!/bin/sh
```

```
/opt/WebOTX/Trnsv/bin/wosh /opt/WebOTX/Trnsv/bin/tpadm2.sh start $1
```

Linux のパッチを適用する際、tpadm.sh が上書きされる可能性があります。適用前に tpadm.sh のバックアップをとってパッチ適用後に復元するか、パッチ適用後に再度「\$1」を加えてください。

2. Java システムプロパティの追加

マルチユーザ用の Java VM オプションを追加します。下に示した domain.xml ファイルを編集して、Java VM オプションを 1 行追加してください。

```
/opt/WebOTX/domains/domain1/config/domain.xml
```

/opt/WebOTX/domains/domain2/config/domain.xml

追加編集する箇所は、<java-config>要素の以下のサブ要素です。

<jvm-options>-Dcom.nec.webotx.multiUser=true</jvm-options>

この設定は、パッチモジュールを適用する前にドメインが起動した状態から、統合運用管理ツールを用いて変更することもできます。

ドメイン名

- アプリケーションサーバ
- JVM 構成 --> [一般]タグ

[一般]画面内の JVM オプションに「-Dcom.nec.webotx.multiUser=true」を追加します。

3. ディレクトリ権限変更

すべての関連するディレクトリとファイルのオーナー・グループ・権限を再設定します。/opt/WebOTX/bin/otxown.sh を用いて、WebOTX 配下のオーナーとグループを root に戻してください。

次に、各ドメインの以下のディレクトリ・ファイルのオーナーとグループを各運用ユーザにそれぞれ変更してください。

ディレクトリまたはファイル	ファイルオーナー	グループ
/opt/WebOTX/domains/domain1 とその配下	otxadm	otxadmin
/opt/WebOTX/domains/domain2 とその配下	otxadm2	otxadmin

WOAgentSvc による起動 (管理ドメインからの起動) を可能にするため、各ドメインの bin ディレクトリにある以下のファイル権限を 755 に設定してください。

ここの権限を変更しても、ユーザチェックがありますので他ユーザによる起動停止は実行できません。

```
/opt/WebOTX/domains/domain1/bin/startserv  
/opt/WebOTX/domains/domain1/bin/stopserv  
/opt/WebOTX/domains/domain2/bin/startserv  
/opt/WebOTX/domains/domain2/bin/stopserv
```

以下のディレクトリとファイルは、各運用ユーザすべてが読み込み・書き込みできる必要があります。それらに対して、運用ユーザのオーナー/グループで書き込みができるようにファイル権限を 775 に設定してください。

(/opt/WebOTX/Trnsv/multitpm に設定するグループは、各運用ユーザと同一にしておく必要があります。)

```
/opt/WebOTX/domains  
/opt/WebOTX/Trnsv/conf/CommAPI とその配下  
/opt/WebOTX/Trnsv/multitpm とその配下 (.'で始まるファイルも)  
/opt/WebOTX/Trnsv/logs  
/opt/WebOTX/Trnsv
```

また、権限返答の際、以下のディレクトリ・ファイルのグループを各運用ユーザと同一に設定してください。

ディレクトリまたはファイル	ファイルオーナー	グループ
/opt/WebOTX/Trnsv/multitpm	root	otxadmin
/opt/WebOTX/domains	root	otxadmin

設定についての補足

設定について、以下の補足事項があります。

1. システム ID の変更

/etc/services にサービス名とポート番号を指定するとき、「w-tpadm+システム ID」の値を指定する必要があります。もしその値が使用できない場合は、既存のシステム ID を変更して、新たな「w-tpadm+システム ID」を付与してください。

そのままの値を使用できるのであれば（例えば、w-tpadm が 8899、システム ID が 2 で、8901 が使用できるなら）、既存のシステムを変更せずに使用できます。変更する場合は、以下の手順で行ってください。

パッチモジュールの適用前にドメインを起動して、統合運用管理ツールから変更を行います。

ドメイン名

- TP システム --> [システム情報] タグ

[システム情報] 画面内の「システム ID」値を更新します。

運用管理コマンドによる設定も可能です。

```
% cd <INSTALLDIR>
% bin/otxadmin
otxadmin> login --user <ユーザ名> --password <パスワード> --host <ホスト名> --port <管理ポート番号>
otxadmin> set tpsystem.systemID=<システム ID>
```

2. Web サーバ設定の変更

WebOTX 運用管理ユーザを root 以外に設定した場合、OS の制約上 1024 番以下のポート番号は使用できません。もし、ポート番号 1024 番以下で内蔵型 Web サーバを使用中のドメインがあれば、そのドメインに対して、1025 番以降の現在のシステムで利用可能な番号に変更してください。

下に示した domain.xml ファイルを編集して、ポート番号を変更してください。

```
/opt/WebOTX/domains/domain1/config/domain.xml
/opt/WebOTX/domains/domain2/config/domain.xml
```

編集する箇所は、<http-service>要素の以下のサブ要素です。id が「http-listener-1」、「http-listener-2」となっている http リスナのポートを変更してください。

```
<http-listener accept-count="10" address="0.0.0.0" buffer-size="2048"
connection-timeout="60000" default-virtual-server="server"
enable-lookups="false" enabled="true" id="http-listener-1" limit-processors="15"
max-processors="20" min-processors="5" port="80" protocol="HTTP/1.1"
security-enabled="false" server-name="" xpowered-by="true">
</http-listener>
<http-listener accept-count="10" address="0.0.0.0" buffer-size="2048"
connection-timeout="60000" default-virtual-server="server"
enable-lookups="false" enabled="true" id="http-listener-2" limit-processors="15"
max-processors="20" min-processors="5" port="443" protocol="HTTP/1.1"
security-enabled="true" server-name="" xpowered-by="true">
</http-listener>
```

この設定は、パッチモジュールを適用する前にドメインが起動した状態から、統合運用管理ツールを用いて変更することもできます。

- ドメイン名
 - アプリケーションサーバ
 - HTTP サービス
 - http リスナ
 - http リスナ 1 --> [一般]タグ
 - http リスナ 2 --> [一般]タグ

[一般]画面内のポート番号を変更します。

運用管理コマンドによる設定も可能です。

```
% cd <INSTALLDIR>
% bin/otxadmin
otxadmin> start-domain <ドメイン名>
otxadmin> login --user <ユーザ名> --password <パスワード> --host <ホスト名> --port <管理ポート番号>
otxadmin> set server.http-service.http-listener.http-listener-1.port=<ポート番号>
otxadmin> set server.http-service.http-listener.http-listener-2.port=<ポート番号>
otxadmin> stop-domain <ドメイン名>
```

3.7. ホスト名の変更

WebOTX インストール後にマシンのホスト名を変更する手順を以下に示します。

新たに設定するホスト名は<hostname>としています。

運用管理コマンドより以下の設定を変更してください。

1. TPMonitorManager

- 接続サーバ名

```
otxadmin > set tpsystem.serverName=<hostname>
```

- 名前サービスホスト名

```
otxadmin > set tpsystem.nameSvHostName=<hostname>
```

- マルチサーバ サーバ名

マルチサーバ設定をしている場合は、サーバ名に変更があるものに対して実行してください。

```
otxadmin > set tpsystem.multiServers.<system group name>.<server set name>.serverName=<hostname>
<system group name>:システムグループ名
<server set name>:サーバセット名
```

2. JDBC データソース

- Oracle との接続文字列 (JDBC URL) の見直し

```
otxadmin > set server.resources.jdbc-datasource.hostname.dataSourceName=<hostname>
```

OCIドライバの場合、tnsnames.ora を変更してください。

thinドライバ場合、URL を変更してください。

3. Object Broker

- Object Broker サービスの名前サーバに関する設定変更

```
server.objectbrokerservice.namesv.NameServiceHostName(名前サーバが動作しているホスト名)
otxadmin> set server.objectbrokerservice.namesv.NameServiceHostName=<hostname>
server.objectbrokerservice.namesv.NameServiceExportHostName(名前サーバの IOR 中に含まれるホスト情報)
otxadmin> set server.objectbrokerservice.namesv.NameServiceExportHostName=<hostname>
server.objectbrokerconfig.NameServiceHostName(名前サービスが動作しているホスト名)
otxadmin> set server.objectbrokerservice.namesv.NameServiceHostName=<hostname>
```

- Object Broker サービスのインタフェースリポジトリに関する設定変更

```
server.objectbrokerservice.irsv.InterfaceRepositoryHost(インタフェースリポジトリが動作するホスト名)
otxadmin> set server.objectbrokerservice.irsv.InterfaceRepositoryHost=<hostname>
server.objectbrokerconfig.InterfaceRepositoryHost(インタフェースリポジトリが動作するホスト名)
otxadmin> set server.objectbrokerconfig.InterfaceRepositoryHost=<hostname>
```

- Object Broker サービスのインタフェースリポジトリに関する設定変更

```
server.objectbrokerservice.namesv.InterfaceRepositoryHost(インタフェースリポジトリが動作するホスト名)
otxadmin> set server.objectbrokerservice.namesv.InterfaceRepositoryHost=<hostname>
server.objectbrokerconfig.InterfaceRepositoryHost(インタフェースリポジトリが動作するホスト名)
otxadmin> set server.objectbrokerconfig.InterfaceRepositoryHost=<hostname>
```

- Object Broker サービスのキャッシュ名前サーバに関する設定変更

```
server.objectbrokerservice.cnamesv.nmsvlist(キャッシュ元となる名前サーバのリスト)
otxadmin> set server.objectbrokerservice.cnamesv.nmsvlist=host1, ..., <hostname>, ..., hostN
```

- IIOP 通信を行う、Object Broker サービス、CORBA アプリケーション、JNDI サービスなどに関する共通な設定変更

```
server.objectbrokerconfig.ExternalHostName(クライアント端末に公開するサーバのホスト名)
otxadmin> set server.objectbrokerconfig.ExternalHostName=<hostname>
server.objectbrokerconfig.AlternativeHostName(リファレンス生成時に格納するホスト名)
otxadmin> set server.objectbrokerconfig.AlternativeHostName=<hostname>
server.objectbrokerconfig.AlternativeHostNames(自ホストとみなすホスト名の一覧)
otxadmin> set server.objectbrokerconfig.AlternativeHostNames=host1:····:<hostname>:····:hostN
```

これらの設定で、明示的に指定しているホスト名を変更してください。

その他のホスト名は、server.objectbrokerconfig.ExternalHostName で一括指定することができます。

この他、ObjectBroker Java のプロパティで ORBInitialHost もしくは NameServiceHost(または NameServiceHostName)を指定していれば、その変更も必要です。クライアントとサーバが同一マシンであれば、既定値は自ホスト(localhost)なので、設定不要です。

「WebOTX Object Broker Java における ORB のプロパティ定義」を参照してください。

4. JMS

- JMS サーバのホスト名

```
otxadmin > set server.jms-service.jms-host.default_JMS_host.host=<hostname>
```

- JMS リソース(コネクションファクトリ)の接続先ホスト名

```
otxadmin> set server.resources.jms-resource.jms-connection-factory.
```

<CF name for JNDI>.wojmsBrokerHostName=<hostname>

- JDBC の接続先のホスト名

ドメインごとの config.properties に記述してある、JDBC の URL に含まれるホスト名を変更してください。

```
$(AS_INSTALL)/domains/$(DOMAIN_NAME)/wojms/instances/wojmsbroker/props/config.properties
```

```
wojms.persist.jdbc.opendburl
```

```
wojms.persist.jdbc.closedburl
```

```
wojms.persist.jdbc.createdburl
```

- JavaAP 用の JMS 管理対象オブジェクト(コネクションファクトリ)の接続先ホスト名

```
$(AS_INSTALL)/wojms/bin/wojmsobjmgr update -t qf -l <CF name for JNDI> -j wojmsBrokerHostName=<hostname>
```

5. Transaction Service

- モニタリングレベルの変更

```
otxadmin> set server.monitoring-service.module-monitoring-levels.transaction-service=LOW
```

- トランザクション一覧取得

```
otxadmin> get --monitor=true server.transaction-service.ActiveIds-Current
```

上記コマンドを実行しますと、残っているトランザクションが表示されます。トランザクションが残っている場合、以下のコマンドでトランザクションをすべて終了させてください。

```
強制ロールバック otxadmin > rollback-transaction <Transaction ID>
```

```
強制コミット otxadmin > commit-transaction <Transaction ID>
```

```
破棄 otxadmin > forget-transaction <Transaction ID>
```

```
強制削除 otxadmin > delete-transaction <Transaction ID>
```

- モニタリングレベルを戻す

```
otxadmin > set server.monitoring-service.module-monitoring-levels.transaction-service=OFF
```

6. Web コンテナ

- 仮想サーバ

```
otxadmin> set server.http-service.virtual-server.<仮想サーバID>.hosts=<hostname>
```

7. JNDI

クライアントで JNDI の URL としてホスト名を以下のように指定している場合は、変更が必要です。

Java オプションとして指定されている場合は以下のように変更してください。

```
-Djava.naming.provider.url=corbaname://<hostname>
```

上記の設定変更が終わりましたら、以下の手順を実行してください。

8. ドメインの停止

```
otxadmin > stop-domain $(DOMAIN_NAME)
```

9. namesv.ndf ファイルの削除

```
$(AS_INSTALL)/domains/$(DOMAIN_NAME)/config/ObjectBroker/namesv.ndf
```

10. ドメインの起動

```
otxadmin > start-domain $(DOMAIN_NAME)
```


3.8. Windowsサービスアカウントの変更

Windows 版 Standard Edition あるいは Enterprise Edition をインストールしたときに [ユーザアカウントの入力] 画面で指定したユーザアカウントは、Windows サービス「WebOTX TPBASEadm」のログオンアカウントとして設定されます。

ユーザアプリケーションプロセスは、このアカウントの環境変数を引き継ぐことになります。

このアカウント、パスワードを変更する場合は以下のように行ってください。

1. Windows のコントロールパネルから、[管理ツール]-[サービス]を実行する
2. WebOTX TPBASEadm を選択し、右クリックでプロパティを開き、[ログオン]タブを表示する
3. 指定されているアカウントとパスワードを指定する

指定するアカウントは Administrators 権限が必要です。また、“サービスとしてログオン”の権利を持っている必要があります。

4.ドメインの運用操作

システムを構築、運用するための手順について説明します。

4.1. ドメインの作成・削除

WebOTX のドメインを作成および削除する方法について説明します。作成・削除時は WebOTX サービスが起動している必要があります。

ドメインの作成

設定ファイル(properties)を asant コマンドの引数に渡すことで生成できます。asant コマンドは\${AS_INSTALL}/bin にあります。新規に domain2 を作成する場合の手順を以下に示します。

- 1.UNIX の場合**運用管理ユーザ**でログインします。
- 2.WebOTX インストールディレクトリに移動します。
- 3.新規ドメイン用の properties ファイルを作成します。作成する properties ファイルは WebOTX インストールディレクトリに配置してください。
(WebOTX インストールディレクトリにある domain1.properties をドメイン名.properties 等の名前で作成します。
例: "domain2"ドメインを作成する場合は、domain2.properties とします。)
- 4.domain2.properties をエディタで開き、各設定を domain2 用に変更します
なお、Web Edition の場合、JMS サービスに関する設定は無視されます。なお**ポート番号や識別名など既存のドメインと重複しないように定義してください。**
- 5.WebOTXサービスが起動していることを確認します
(起動していない場合は、「2.2サービスの起動・停止方法」の手順で起動します)
- 6.WebOTX インストールディレクトリで以下のコマンドを実行します
> ./bin/asant -buildfile setup.xml -Dfile=domain2.properties create-domain

作成したドメインを起動する場合は、

```
`${INSTALL_ROOT}`/bin/otxadmin start-domain domain2
```

コマンドで起動してください。なお **start-domain** コマンドの実行は必ず **WebOTX 運用ユーザ**で行なってください。

プロパティ一覧

ドメイン設定(domain.*)

プロパティ	説明	domain1 の既定値
hostname	ホスト名 ※ホスト名に使える文字については、RFC 1034(DNS)として標準化されている、A-z のアルファベットと"." "-"の組み合わせになります。"."(アンダースコアが含まれる場合や、数字から始まるホスト名は指定できません。	localhost
name	ドメイン名 ※必ずドメイン間で重複しないよう一意に設定してください	domain1

admin.user	管理ドメインに対して定義されている管理ユーザ	admin
admin.password	管理ユーザのパスワード	adminadmin
admin.port	エージェントが利用する JMXMP のポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	6212
instance.port	HTTP サーバインスタンスが利用する HTTP のポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	80
https.port	SSL で保護された HTTP ポートの番号 ※必ずドメイン間で重複しないよう一意に設定してください。	443
http.admin.port	Web コンテナが利用する管理コンソール用 HTTP ポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	4848
http.ajp.port	mod_jk2 を使用した HTTP サーバ-Web コンテナ連携ポートの番号 ※必ずドメイン間で重複しないよう一意に設定してください。	8009
jms.port	JMS 管理サーバが利用するポート番号 Web Edition の場合は無効です ※必ずドメイン間で重複しないよう一意に設定してください。	9700
ipv6-enable	IPv6 優先 ※ドメインの Java システムプロパティ java.net.preferIPv6Addresses の値に影響します	false

サーバライフサイクルモジュール設定(server.internal-lifecycle-module.*)

プロパティ	説明	domain1 の既定値
ObjectBrokerService.enabled	Object Broker サービス起動フラグ	true
JNDIService.enabled	JNDI サービス起動フラグ	true
J2EEService.enabled	J2EE サーバ起動フラグ	true
JMSProvider.enabled	JMS サービス起動フラグ Web Edition の場合は無視されます	true
WebContainerService.enabled	Web コンテナサービス起動フラグ	true
TransactionService.enabled	Transaction サービス起動フラグ	true
WebServerService.enabled	Web サーバ(Apache ベース)サービス起動フラグ	true
TPMonitorManagerService.enabled	TP モニタ・マネージャサービス起動フラグ	true

Java VM 設定(server.java-config.*)

プロパティ	説明	domain1 の既定値
debug-options	Java デバッグオプション	-Xdebug -Xrunjdw:transport=dt_socket,server=y,suspend=n, address=4004
debug-enabled	デバッグフラグ	false

Java VM 設定を有効にする場合は、コメントを外してください。

WebServer サービス設定(Web コンテナと Web サーバを連携するために必要な設定)

プロパティ	説明	domain1 の既定値
domain.path	ドメインのルートパス	\${INSTALL_ROOT} /domains/ドメイン名
domain.webserver.type	WebServer の種別 インストール時に指定した以下のサーバタイプのいずれかを指定します。 WebOTX WebServer : (指定なし) Apache HTTP Server 1.3: WebServer Apache HTTP Server 2.0: WebServer2	Web/Standard-J Edition (指定なし) Std/Enterprise Edtion WebServer2
domain.webserver.path	WebServer のインストールパス	Web/Standard-J Edition (指定なし) Std/Enterprise Edtion \${INSTALL_ROOT} /WebServer2
domain.webserver.name	WebServer 名	(指定なし)
domain.webserver.plugin.querymode	Web アプリケーションのコンテキスト名動的反映	on

Object Broker サービス設定(server.objectbrokerservice.*)

プロパティ	説明	domain1 の既定値
oadj.Port	oadj が使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	9826
corbaloc.CorbalocDefaultPort	URL でポート番号を指定しなかったときの値 ※必ずドメイン間で重複しないよう一意に設定してください。	2809

corbaloc.CorbalocServerPort	corbaloc サーバが使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	9827
namesv.NameServicePort	名前サーバが使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	2809
namesv.NameServiceRoundRobin	名前サーバのラウンドロビン拡張機能	true
irsv.InterfaceRepositoryPort	インタフェースリポジトリが使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	9830
oad.OadPort	接続先の oad が使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	9825
cnamesv.CacheNameServicePort	キャッシュ名前サーバが使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。 ※キャッシュ名前サーバは Enterprise Edition でのみインストールされます。	9829

Web コンテナ サービス設定(server.web-container.*)

プロパティ	説明	domain1 の既定値
processMultiplicity	Web コンテナ 動作モード マルチプロセスモード: true シングルプロセスモード: false	Web/Std-j Edition: false Std/Ent Edition: true

TP モニタ・マネージャサービス 設定 (tpsystem.*)

プロパティ	説明	domain1 の既定値
systemNameW	ドメインで動作する TP システム名 8 文字以内の英数字で指定 ※必ずドメイン間で重複しないよう一意に設定してください。	MySystem
systemID	TP システムを一意に識別する ID 0~255 の間で指定 ※必ずドメイン間で重複しないよう一意に設定してください。	0
admListener.portNumber	WebOTX Adm リスナが使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	5202
IIOPListener.listenerPortNumber	WebOTX IIOP リスナが使用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	5151
OLFTPLListener.listenerPortNumber	WebOTX OLF リスナが使用するポート番号	5251

※必ずドメイン間で重複しないよう一意に設定してください。

Transaction サービス 設定 (server.transactionservice.*)

プロパティ	説明	domain1 の既定値
rsc-cpp-port	C++版の RCS が使用するポート番号 (Web Edition、および Standard-J Edition では使用されません)。 ※必ずドメイン間で重複しないよう一意に設定してください。	5965

Reconfig 用 (server.*)

プロパティ	説明	domain1 の既定値
http-service.http-listener.http-listener-1.port	HTTP サーバインスタンスが利用する HTTP のポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	80
http-service.http-listener.http-listener-2.port	SSL で保護された HTTP ポートの番号 ※必ずドメイン間で重複しないよう一意に設定してください。	443
http-service.http-listener.admin-listener.port	Web コンテナが利用する管理コンソール用 HTTP ポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	4848
http-service.http-listener.ajp-listener-1.port	mod_jk2 を使用した HTTP サーバ-Web コンテナ連携ポートの番号 ※必ずドメイン間で重複しないよう一意に設定してください。	8009
jms-service.jms-host.default_JMS_host.port	JMS 管理サーバが利用するポート番号 ※必ずドメイン間で重複しないよう一意に設定してください。	9700

上記設定はドメイン新規作成時には利用しません。コメントとしてください。ドメイン作成後に Reconfig によりポート番号を変更するときにコメントを外して設定してください。

ドメインの削除

削除したいドメイン名を asant コマンドの引数に指定して削除します。asant コマンドは \${AS_INSTALL}/bin にあります。

domain2 を削除する場合の手順を以下に示します。

1. UNIX の場合 WebOTX 運用ユーザでログインします。

2. WebOTX インストールディレクトリに移動します。
3. WebOTXサービスが起動していることを確認します
(起動していない場合は、「2.2サービスの起動・停止方法」の手順で起動します)
4. Transaction サービスを利用したトランザクション管理を行っている場合は、Transaction サービスで提供する wotstx コマンドを使用して、全てのトランザクションが存在しないことを確認してください。トランザクションが残っている場合は全てのトランザクションを終了させてください。
5. ドメインが起動している場合は停止します
otxadmin> stop-domain domain2
6. 以下のコマンドを実行します
> ./bin/asant -buildfile setup.xml -Ddomain.name=domain2 delete-domain

4.2. ドメインの確認

ドメインの状態を確認する方法について説明します。

4.2.1. ドメインの一覧確認

ドメインの一覧を確認するには次のコマンド(list-domains)を実行してください。そのマシン上で動作するドメインの一覧が表示されます。

注意:このコマンドはローカルマシンからしか実行できません。

```
otxadmin> list-domains
```

(実行結果)

List of domains:

domain1 running

WebOTXAdmin running

補足 : WebOTXAdmin は管理ドメインです。

各ドメイン名の表示につづけて以下のいずれかの状態が表示されます。

domain 名 not running ...停止中

domain 名 starting ...起動処理中

domain 名 running ...稼動中

4.2.2. ドメインの起動完了確認

ドメインの起動が完了し、サービスの要求を受け付けられる状態になっているか確認する方法について説明します。

イベントログによる確認

以下のメッセージがドメインの起動・停止完了時にイベントログへ通知されます。

```
OTX01205135: アプリケーションサーバ [domain1:server] の起動に成功しました。(com.nec.webotx.enterprise.system.core)
```

ログファイルの出力による確認

以下のメッセージがドメインの起動・停止完了時に各ドメインのログファイル(webotx_agent.log)に出力されます。

```
9999-99-99 00:00:00,000 SLOGINFO com.nec.webotx.enterprise.system.core - OTX01205135: アプリケーションサーバ  
[domain1:server] の起動に成功しました。 [org.apache.commons.launcher.ChildMain]
```

コマンドの実行ステータスによるサービスの状態確認

otxadmin コマンドの get コマンドを利用し WebOTX の各サービスの状態を確認します。

get コマンドの --exitstatus オプションを利用することで otxadmin コマンドの終了コードに状態コードを反映させることができます。状態によってバッチ処理を行いたいような場合に利用することができます。

例) Web コンテナの起動状態を確認する

```
otxadmin> get --exitstatus=true --exiterrornum=-1 server.internal-lifecycle-module.WebContainerService.state  
if %errorlevel%==1 echo WebContainer running!
```

状態コードは以下のように対応しています。

状態	状態表示 (コード)
開始処理中(starting)	0
実行中(running)	1
停止処理中(stopping)	2
停止中(stopped)	3
起動・停止失敗(failed)	4
(APG)クライアント接続中(client connecting)	10
(PG)オペレーション実行中(process executing)	11
(PG)リカバリ処理中(recovery executing)	12

状態(state)をサポートするサービス

サービス	dottedmane
ドメイン管理サービス	server.internal-lifecycle-module.AdminService.state

JMSサービス	server.internal-lifecycle-module.JMSProvider.state
Object Broker サービス	server.internal-lifecycle-module.ObjectBrokerService.state
JNDIサービス	server.internal-lifecycle-module.JNDIService.state
トランザクションサービス	server.internal-lifecycle-module.TransactionService.state
Webコンテナサービス	server.internal-lifecycle-module.WebContainerService.state
J2EEサーバ	server.internal-lifecycle-module.J2EEServer.state
トランザクションモニタサービス	server.internal-lifecycle-module.TPMonitorManagerService.state
Webサーバ	server.internal-lifecycle-module.WebServerService.state
JMX リモートコネクタサービス	server.internal-lifecycle-module.RemoteJmxConnector.state

4.3. ドメインの起動・停止

ドメインの自動起動設定

WebOTX サービス起動時にドメインを自動起動させるかどうかの設定について説明します。なおデフォルトでは作成したドメインは自動起動します。

ドメインの自動起動を行わないよう設定するには管理ドメインに対して以下のコマンドを実行します。

domain1 の自動起動を無効に設定

```
>otxadmin
otxadmin>login --user admin -password **** --port 6202
otxadmin>invoke domain.disable domain1
otxadmin>exit
```

再度、自動起動を有効にするには管理ドメインに対して以下のコマンドを実行します。

domain1 の自動起動を有効に設定

```
>otxadmin
otxadmin>login --user admin -password **** --port 6202
otxadmin>invoke domain.enable domain1
otxadmin>exit
```

管理ドメインの起動/停止

管理ドメインは WebOTX サービスと連動して起動、停止を行います。管理ドメインを起動・停止するには WebOTX サービスを起動・停止させてください。なおサービスを停止すると全てのドメインは停止されます。

一般ドメインの起動/停止

特定のドメインを起動・停止させるには次のコマンドを実行してください。なおこのコマンドはローカルマシン上でのみ実行可能です。なおコマンドの実行は必ず WebOTX 運用ユーザで行なってください。

起動

```
`${INSTALL_ROOT}`/bin/otxadmin start-domain domainname
```

停止

```
`${INSTALL_ROOT}`/bin/otxadmin stop-domain domainname
```

マルチモードからのドメインの起動

UNIX 環境において、otxadmin コマンドを引き数なしのマルチモードでドメインを起動し、ターミナルウィンドウを“exit”を入力しないで強制終了させた場合にドメインが停止します。マルチモードでドメインを起動した場合は、必ず“exit”を入力して otxadmin コマンドを 終了するようにしてください。

マルチモードからドメインを起動した場合、OS に関わらず画面に以下のメッセージが表示されます。

```
Please enter "exit" after Domain domainname started. For more information about multimode subcommand, see operation guide on WebOTX online manual.
```

サービスでの運用とコマンドでの運用の違い

各ドメインが Windows 系 OS 上で起動される場合、start-domain、stop-domain コマンドによるドメイン運用はあくまで開発・評価時に対象となるドメインの起動・停止操作を速やかに行えることを第一の目的として提供しているコマンドです。

この運用方法の違いはプロセスを起動するユーザの違いに現れます。サービスから運用した場合、WebOTX の各種プロセスはシステムユーザとして起動されます。対して、運用管理コマンドから起動した場合は、運用ユーザのプロセスとして起動します。そのため、起動ユーザの環境設定に依存してしまい、運用時の思わぬトラブルの原因となってしまう場合があります。

そのため、本番環境での運用時に start-domain、stop-domain コマンドによるドメイン制御は行わないようご注意ください(本番時は WebOTX サービスとして各ドメインの起動・停止処理を行う運用となります)。

なお、ユーザドメインに対して次のコマンドを実行することで、管理ドメインと同一のユーザプロセスとして起動することが可能です。既に管理ドメインをサービスとして起動していて、システムユーザとしてユーザドメインを運用したい場合にご利用ください。

```
otxadmin> login --user admin --password xxxx --port 6202
otxadmin> invoke domain.startDomain domainname
```

併せて、ドメイン停止時は管理ドメインに対して次のコマンドを実行してください。

```
otxadmin> login --user admin --password xxxx --port 6202
otxadmin> invoke domain.stopDomain domainname
```

Windows系OSでドメインを動作させる場合の注意点

各ドメインが Windows 系 OS 上で起動される場合、start-domain コマンドによるドメイン起動時には、通常のアプリケーションを実行した時のようにタスクバー上に(ドメイン名がタイトル名となる)ウィンドウが表示される仕組みとなっています。また、その仕組みに合わせ、ウィンドウを閉じるとドメインが自動的に停止するようになっています。誤操作によってこのウィンドウを不意に閉じないよう注意してください。

なお、コマンドからのドメイン起動時にこのウィンドウを表示させなくするには、管理ドメインに対して上記の invoke コマンドを実行してください。

4.4. 運用ユーザの設定

WebOTX 運用ユーザの設定は、create-file-user コマンドを用いて行います。

指定するグループには必ず otxadmin を指定してください。また、設定するパスワードは8文字以上になるよう指定します。

作成したユーザはそのドメインのみ管理することが可能です。作成したドメインの再起動後有効になります。

以下に、サーバ host1 にある domain1 (ポート番号 6212) に対して、ユーザ名 admin1、パスワード adminadmin1 の運用ユーザを作成する例を示します。

```
otxadmin> create-file-user --userpassword adminadmin1 --groups otxadmin admin1
```

作成に成功したら、host1 で domain1 を再起動します。

```
otxadmin> stop-domain domain1
```

```
otxadmin> start-domain domain1
```

運用ユーザの削除は、delete-file-user コマンドで行います。

```
otxadmin> delete-file-user admin2
```

削除に成功したら、host1 で domain1 を再起動します。再起動されるまでは削除が有効になりません。

```
otxadmin> stop-domain domain1
```

```
otxadmin> start-domain domain1
```

4.5. 各サービスの起動・停止

ドメイン上で動作している各サービスを起動、停止する方法について説明します。

自動起動の設定

ドメインの起動時にサービスを自動起動させるかどうかの設定方法について説明します。

自動起動の確認

現在の設定を確認するには次のコマンドを実行します。

otxadmin> get server.internal-lifecycle-module.service-name.enabled
true の場合は自動起動設定、false の場合は自動起動しない設定です。
service-name には確認するサービス名が入ります。

service-name	説明
JMSProvider	JMS サービス
JNDIService	JNDI サービス
ObjectBrokerService	Object Broker サービス
TransactionService	Transaction サービス
WebContainerService	Web コンテナ
WebServerService	HTTP サーバ(Apache ベース WebOTX Web サーバ)
TPMonitorManagerService	TPモニタ・マネージャサービス

例) Object Broker サービスの自動起動設定を確認

```
otxadmin> get server.internal-lifecycle-module.ObjectBrokerService.enabled  
server.internal-lifecycle-module.ObjectBrokerService.enabled = true
```

自動起動の設定

自動起動の設定を変更するには次のコマンドを実行します。

```
otxadmin> set server.internal-lifecycle-module.service-name.enabled = { true | false }
```

service-name には設定を変更するサービス名が入ります。true の場合は自動起動設定、false の場合は自動起動しないに変更します。

注意: 上記表以外のサービスの自動起動設定は変更しないでください。正常にドメインが起動しなくなります。

例) Object Broker サービスの自動起動しないように設定する。

```
otxadmin> set server.internal-lifecycle-module.ObjectBrokerService.enabled=false
```

JMSサービス

JMS サービスの起動・停止方法について説明します。

起動

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

または

```
otxadmin> start-jms
```

停止

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

または

```
otxadmin> stop-jms
```

JNDIサービス

JNDI サービスの起動・停止方法について説明します。

起動

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

停止

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

Object Broker

Object Broker サービスの起動・停止方法について説明します。

起動

```
otxadmin> invoke server.objectbrokerservice.start
```

停止

```
otxadmin> invoke server.objectbrokerservice.stop
```

Transactionサービス

Transaction サービスの起動・停止方法について説明します。

起動

```
otxadmin> invoke server.transactionservice.start
```

または

```
otxadmin> start-transaction-service
```

停止

```
otxadmin> invoke server.transactionservice.stop
```

または

```
otxadmin> stop-transaction-service
```

Webコンテナ

Web コンテナの起動・停止方法について説明します。

起動

```
otxadmin> invoke server.internal-lifecycle-module.WebContainerService.start
```

停止

```
otxadmin> invoke server.internal-lifecycle-module.WebContainerService.stop
```

Webサーバ(WebOTX Webサーバ インストール時)

Web サーバの起動・停止方法について説明します。

起動

```
otxadmin> invoke server.internal-lifecycle-module.WebServerService.start
```

停止

```
otxadmin> invoke server.internal-lifecycle-module.WebServerService.stop
```

TPモニタ・マネージャサービス

TP モニタ・マネージャ自身の起動停止はできません。TP モニタ単位で行います。

TP モニタの起動・停止方法について説明します。

起動

```
otxadmin> start-system
```

停止

```
otxadmin> stop-system
```

4.6. ポリシーファイルの設定

WebOTX はセキュリティポリシーによりアクセス権が明示的に設定されています。

WebOTX の各サービスが動作するために必要なセキュリティポリシーはデフォルトで以下のファイルにより設定されています。

```
${ INSTANCE_ROOT }/config/server.policy
```

アプレット（またはセキュリティマネージャの下で動作しているアプリケーション）が、ファイルの読み書きなど、セキュリティ保護された操作を行うためには、その操作を行うためにアクセス権を付与する必要があります。必要に応じて server.policy ファイルに必要な権限を設定するようにしてください。

WebOTX では、各サービスの基本的動作に必要な権限に加え、統合運用管理ツールなどリモートからの運用操作に必要なアクセス権限をデフォルトで定義しています。リモート接続時に必要な権限は、プリンシパル `javax.management.remote.JMXPrincipal "admin"` によって定義されています。

例) ポリシーの追加

```
// Grant everyone the following permission:
grant {
    permission java.io.FilePermission "/tmp/*", "read,write";
};
```

値の意味は、末尾の文字によって変わります。末尾が「/」は、指定されたディレクトリ内のすべてのクラスファイル（JAR ファイルでない）に一致します。末尾が「/*」は、そのディレクトリ内にあるすべてのファイル（クラスファイルと JAR ファイルの両方）に一致します。末尾が「/-」は、ディレクトリのすべてのファイル（クラスファイルと JAR ファイルの両方）、および再帰的にそのディレクトリのサブディレクトリにあるすべてのファイルを表します。

セキュリティポリシー、ポリシーファイルの記述に関する詳細は、Java SDK の「セキュリティ」に関する情報を参照してください。

<http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/security/>

4.7. Java VM環境の設定

Java VM の環境設定方法について説明します。

4.7.1. クラスパスの設定

JDBCドライバや独自のライブラリをクラスパスに追加する方法について説明します。

コマンドによるclasspath-suffixの設定

運用管理コマンド(otxadmin)を利用することにより任意の場所にあるライブラリをクラスパスに追加できます。追加はCLASSPATHの最後に行われます。よってWebOTXのシステムライブラリが先に検索されます。

注意:set コマンドで指定するパスは追加ではありません。全て上書きします。

以下のコマンドを実行します。

確認

```
otxadmin> get server.java-config.classpath-suffix
```

設定

```
otxadmin> set server.java-config.classpath-suffix=class_path
```

例) クラスパスに d:\mylib.jar を追加

```
otxadmin> set server.java-config.classpath-suffix=d:\mylib.jar
```

\${INSTANCE_ROOT}/lib/extへの配置

ライブラリを\${INSTANCE_ROOT}/lib/ext ディレクトリにコピーすることにより自動的にそのライブラリをクラスパスに追加することができます。ここに jar, zip ファイルをコピーすると WebOTX のシステムライブラリより先に検索されます。このため、例えば J2EE の API クラスなどの WebOTX のシステムライブラリに含まれるクラスを含むライブラリをここにコピーすると、クラスロードの依存関係が変わり、エージェントが起動できなくなる可能性がありますので注意してください。

\${INSTANCE_ROOT}/libへの配置

ライブラリを\${INSTANCE_ROOT}/lib ディレクトリにコピーすることにより自動的にそのライブラリをクラスパスに追加することができます。ここに jar, zip ファイルをコピーすると WebOTX のシステムライブラリより後で検索されます。

共有コンポーネントとして登録 (Standard/Enterprise Editionのみ)

ライブラリを共有コンポーネントとして作成し、ドメインに配備することで CORBA Java アプリケーションまたは EJB アプリケーションからそのライブラリを参照することができます。この場合配備したアプリケーションの設定で参照する共有コンポーネントを指定する必要があります。

プロセスグループの環境変数CLASSPATHへの追加 (Standard/Enterprise Editionのみ)

CORBA Java アプリケーションまたは EJB アプリケーションのプロセスグループの環境変数 CLASSPATH を指定するとそのプロセスグループのクラスパスを指定することができます。

各コンテナでロードされるライブラリと検索の優先順位

EJB コンテナは Standard-J Edition と Standard/Enterprise Edition では、実行される Java VM のプロセスが異なります。Standard-J Edition の場合は、エージェント同じ Java VM 上で動作しますが、Standard/Enterprise Edition の場合は、エージェントとは別のプロセスグループの Java VM 上で動作します。

Edition 別のコンテナがロードするライブラリと、検索の優先順位を以下に示します。優先度の高いもの順から記述しています。“○”はロードされることを示し、“×”はロードされないことを示します。

設定方法	Web コンテナ	EJBコンテナ (Standard-J)	EJB コンテナ (Standard/Enterprise)
------	----------	----------------------	--------------------------------

`\${INSTANCE_ROOT}/lib/ext` への配置	○	○	○
WebOTXシステムライブラリ	○	○	○
classpath-suffix	○	○	×
プロセスグループの環境変数 CLASSPATH への追加	×	×	○
`\${INSTANCE_ROOT}/lib`	○	○	○
共有コンポーネントとして配備	×	×	○

4.7.2. VMオプションの設定

独自の VM オプションを設定する方法について説明します。

コマンドによる設定

運用管理コマンド(otxadmin)を利用することにより任意の場所にあるライブラリをクラスパスに追加できます。なお運用管理コマンド「create-jvm-options」、「delete-jvm-options」については「[運用管理コマンドリファレンスマニュアル](#)」を参照してください。

注意:VM オプション削除時はプロパティ名だけでなく現在設定されているプロパティの値も指定してください。

また、現在設定しているプロパティを変更するにはいったん削除して作成しなおしてください。

以下のコマンドを実行します。

確認

```
otxadmin> get server.java-config.jvm-options
```

設定

```
otxadmin> create-jvm-options jvm_option_name=jvm_option_value
```

削除

```
otxadmin> delete-jvm-options jvm_option_name=jvm_option_value
```

例) -Dmy-debug-option=true を追加

```
otxadmin> create-jvm-options -Dmy-debug-option=true
```

なおヒープサイズのような「name=value」の形式にならないオプションを設定、削除する場合は明示的にオプションの終了を示す「:」(コロン)を最後に付け加えてください。

例) -Xmx512m を追加

```
otxadmin> create-jvm-options -Xmx512m:
```

オペランドの前に -- を記述することでも実行可能です。

```
otxadmin> create-jvm-options -- -Xmx512m
```

4.8. 統計情報の取得

統計情報の採取および取得方法について説明します。なお詳細は「[WebOTX運用編\(モニタリング\)](#)」を参考にしてください。

4.8.1. 統計情報の出力レベル設定

まず統計方法の出力レベルを設定します。otxadmin の set コマンドを利用してレベルを設定します。なおモジュールにより設定できる出力レベルが異なります。

dotted name: server.monitoring-service.module-monitoring-levels.(モジュール)

指定する値: OFF,ON,HIGH,LOW

- OFF : 統計情報を出力しない
- ON : 統計情報を出力する(LOWと同様)
- LOW : 概要レベルの統計情報を出力する
- HIGH : 詳細レベルの統計情報を出力する

設定可能な統計情報

モジュール	既定値	説明
connector-connection-pool	OFF	コネクションプールに関する統計情報 OFF もしくは ON を設定します
ejb-container	OFF	EJB コンテナに関する統計情報 OFF もしくは LOW もしくは HIGH を設定します
http-service	OFF	Web コンテナの HTTP サービスに関する統計情報 OFF もしくは ON を設定します
jdbc-datasource	OFF	JDBC データソースに関する統計情報 OFF もしくは ON を設定します
thread-pool	OFF	スレッドプールに関する統計情報 OFF もしくは ON を設定します
transaction-service	OFF	トランザクションサービスに関する統計情報 OFF もしくは LOW もしくは HIGH を設定します
web-container	OFF	Web コンテナに関する統計情報 OFF もしくは ON を設定します Web アプリケーションの統計情報を採取するには、運用管理コンソールの「統計情報」-「Web Application の統計情報」画面で統計情報の収集を開始(収集開始ボタンを押下)しておく必要があります。
jms-service	OFF	JMS サービスに関する統計情報 OFF もしくは ON を設定します

例) ejb-container のレベルを LOW 設定します

```
otxadmin> set server.monitoring-service.module-monitoring-levels.ejb-container=LOW
```

4.8.2. 統計情報の出力レベル確認

現在の統計情報の出力レベルを確認するには otxadmin の set コマンドを利用します。

例) 全てのレベルを確認

```
otxadmin> get server.monitoring-service.module-monitoring-levels.*
server.monitoring-service.module-monitoring-levels.connector-connection-pool = OFF
server.monitoring-service.module-monitoring-levels.ejb-container = LOW
server.monitoring-service.module-monitoring-levels.http-service = OFF
server.monitoring-service.module-monitoring-levels.jdbc-datasource = OFF
server.monitoring-service.module-monitoring-levels.orb = OFF
server.monitoring-service.module-monitoring-levels.thread-pool = OFF
server.monitoring-service.module-monitoring-levels.transaction-service = OFF
server.monitoring-service.module-monitoring-levels.web-container = OFF
```

例) 特定のモジュール(ejb-container)のレベルを確認

```
otxadmin> get server.monitoring-service.module-monitoring-levels.ejb-container
server.monitoring-service.module-monitoring-levels.ejb-container = LOW
```

4.8.3. 統計情報の取得

統計情報の出力レベルを ON,LOW,HIGH にすると、統計情報採取用の MBean(StatsHolder MBean)が登録されますので、その属性を otxadmin コマンドで統計情報を取得することができます。

例) 現在登録されている統計情報一覧取得

```
otxadmin> get --monitor=true *
server.jvm.HeapSize-Current = 37822464
server.jvm.HeapSize-Description = Provides statistical information about the JVM's memory heap size.
server.jvm.HeapSize-HighWaterMark = 37822464
server.jvm.HeapSize-LastSampleTime = 1096597651824
server.jvm.HeapSize-LowWaterMark = 0
server.jvm.HeapSize-LowerBound = 0
server.jvm.HeapSize-Name = JvmHeapSize
server.jvm.HeapSize-StartTime = 1096597764867
server.jvm.HeapSize-Unit = bytes
server.jvm.HeapSize-UpperBound = 518979584
server.jvm.UpTime-Count = 1096597764877
server.jvm.UpTime-Description = Provides the amount of time the JVM has been running.
server.jvm.UpTime-LastSampleTime = 1096597764867
server.jvm.UpTime-Name = JvmUpTime
server.jvm.UpTime-StartTime = 1096597651814
server.jvm.UpTime-Unit = milliseconds
```

4.8.4. Transactionサービス(JTA)統計情報の取得

Transaction サービス(JTA)に関する統計情報の採取および取得方法について説明します。

パフォーマンスデータの取得方法

統合運用管理ツールを利用する場合

The screenshot shows the WebOTX統合運用管理ツール (WebOTX Unified Operation Management Tool) interface. The left sidebar displays a tree view of system components, with 'Transactionサービス' (Transaction Service) selected under 'WebOTX' and '統計情報' (Statistics). The main window displays the following information:

- ActiveIds:** A table showing active transactions.
- ActiveCount:** A summary of active transactions.
- CommittedCount:** A summary of committed transactions.
- Event Notification Table:** A table for monitoring service events.

ActiveIds	Transaction ID	Status	A/R	Start Time
トランザクション	1840303728815046	StatusActive	Active	2007/6/11 10:58:5
覧	1949615270862589	StatusActive	Active	2007/6/11 10:58:5

ActiveCount	Value
実行中トランザクション数	1

CommittedCount	Value
総コミット数	16,029,951

日付	時刻	ホスト名	ドメイン名	タイプ	シーケンス番号	メッセージ

Transactionサービス ユーザ: admin ドメイン: domain1

次の手順で実施します。

1. 統合運用管理ツールよりドメインと接続します。
2. ツリービューより「<ドメイン名>」-「統計情報」-[domain] -「アプリケーションサーバ」-「トランザクションサービス」を選択すると右側に表示されます。ただしこのツリーは、モニタリングレベルが OFF 以外に設定されている必要がありますのでご注意ください。設定方法については運用編(モニタリング)をご参照ください。

コマンドを利用する場合

次の手順で実施します。

1. otxadmin コマンドを起動し、ドメインにログインします。
otxadmin>login --user admin --password adminadmin --port 6212
2. 次のコマンドを実行します。ただし統合運用管理ツールの場合同様に、モニタリングレベルが OFF に設定されている場合失敗します。

```
otxadmin> get --monitor=true server.transaction-service.*
```

これらを実行すると次のような内容が表示されます。

```
server.transaction-service.ActiveCount-Count = 0
server.transaction-service.ActiveIds-Current = No Transactions exists.
server.transaction-service.AverageTAT-Count = 26975
server.transaction-service.CommittedCount-Count = 5
server.transaction-service.CompletedCount-Count = 7
server.transaction-service.DeletedCount-Count = 0
server.transaction-service.ForgottenCount-Count = 0
server.transaction-service.HeuristicCount-Count = 0
server.transaction-service.RolledbackCount-Count = 2
server.transaction-service.TimedOutCount-Count = 0
server.transaction-service.TransferredCount-Count = 1
```

トランザクション一覧の取得、および操作方法

モニタリングレベルを LOW、あるいは HIGH に設定した場合には、統合運用管理ツール、および otxadmin コマンドを使用して実行中のトランザクション情報も取得することができます。また表示したトランザクションに対する操作も可能です。

なお、モニタリングレベルが OFF の場合は otxadmin コマンドでは表示されません。

次に otxadmin コマンドを使用したトランザクション一覧の取得、および操作について説明します。

トランザクション一覧の表示

otxadmin コマンドを使用する場合、属性名を次のように指定します。

```
otxadmin> get --monitor=true server.transaction-service.ActiveIds-Current
```

これにより実行中トランザクションの一覧が確認できます。実行中のものが存在しない場合は

```
server.transaction-service.ActiveIds-Current = No Transaction Exists.
```

と表示されます。存在する場合は

```
server.transaction-service.ActiveIds-Current =
Transaction ID      Status              A/R      Start Time
-----
0277689751350000 StatusCommitting   Active    2004/10/8 14:39:53
0384489751290000 StatusCommitting   Active    2004/10/8 14:39:54
0255289751170000 StatusPrepared      Active    2004/10/8 14:39:57
0277689749150001 StatusUnknown       Recover   2004/10/8 14:40:49
0384489749070001 StatusUnknown       Active    2004/10/8 14:40:51
```

のように表示されます。統合運用管理ツールでは前述の方法で実施すると右側に上記のように表示されます。

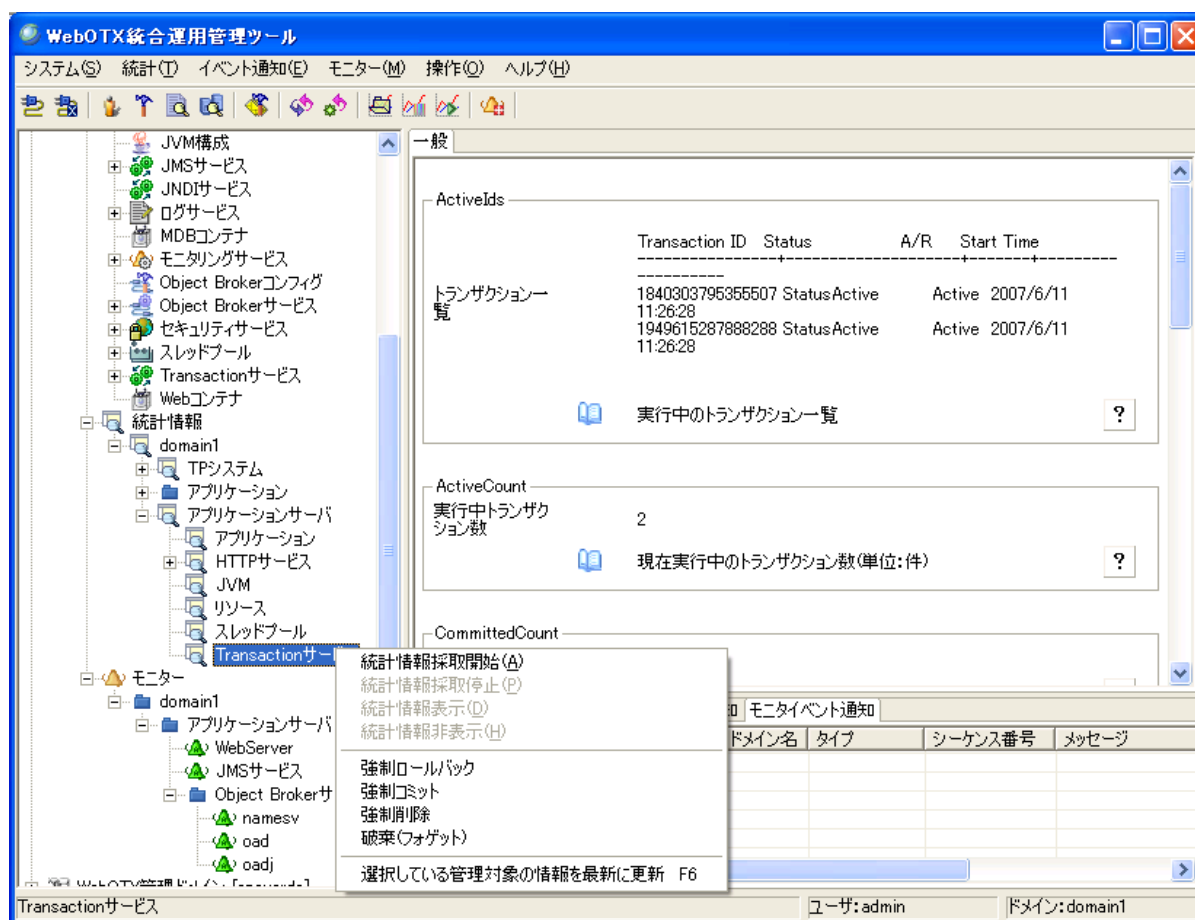
次にそれぞれの項目について説明します。詳細については運用編(コンフィグレーション)をご参照ください。

項目	説明
TransactionID	トランザクション識別子です。これを利用してトランザクションの操作を実施します。
Status	トランザクションの状態 です。

A/R	RCS にトランザクションが移管されているかどうかを表します。 「Active」の場合は移管されておらず、「Recover」の場合は移管されていることを示します。
Start Time	トランザクションが開始された日時です。

トランザクションの操作

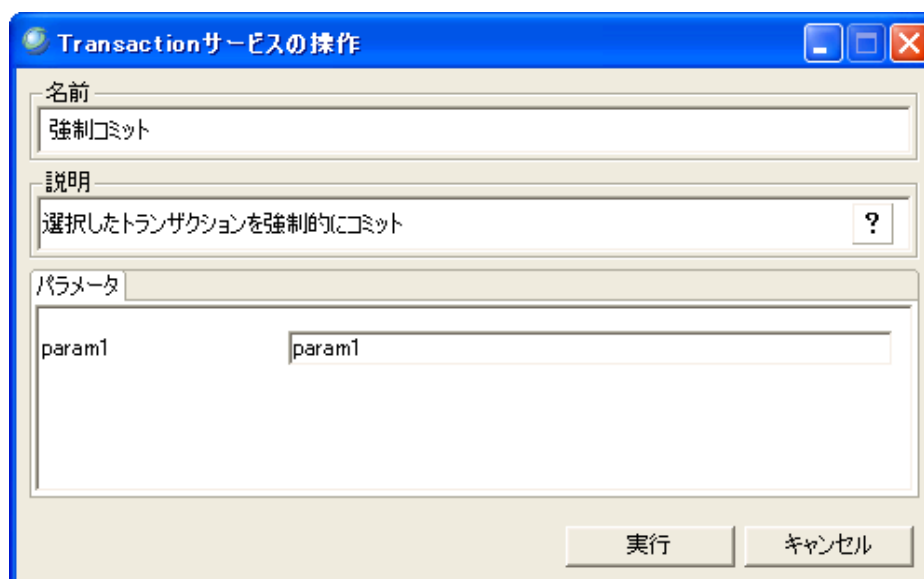
トランザクションの操作については次のように実施します。



統合運用管理ツールを利用する場合

1. 統合運用管理ツールよりドメインと接続します。
2. ツリービューより「<ドメイン名>」-「統計情報」-[domain]-「アプリケーションサーバ」-「トランザクションサービス」を選択します。
3. 右側に表示される Transaction ID をコピーし、「トランザクションサービス」ツリーを右クリックして上図のようなメニューを出します。
4. 実施するコマンドを指定してください。なお、トランザクションの状態によって実行可能なコマンドが制限されます。詳細は「[トランザクションの状態一覧](#)」をご参照ください。
5. 次のような画面が出ますので param1 のところに 3.でコピーした Transaction ID を貼り付けて「実行」を押して処理を開始

してください。



コマンドを利用する場合

1. otxadmin コマンドを起動し、ドメインにログインします。
otxadmin>login --user admin --password adminadmin --port 6212
2. それぞれ次のように、トランザクション識別子を指定してコマンドを実施します。ただしモニタリングレベルが OFF に設定されている場合失敗します。なお、この処理によってデータベースの状態が不安定になる可能性がありますので実行には十分注意が必要です。

トランザクションの強制コミット

```
otxadmin> commit-transaction 0277689751350000
```

トランザクションの強制ロールバック

```
otxadmin> rollback-transaction 0277689751350000
```

トランザクションの破棄(フォゲット)

```
otxadmin> forget-transaction 0277689751350000
```

トランザクションの強制削除

```
otxadmin> delete-transaction 0277689751350000
```

また、トランザクションの操作については、Transactionサービス(統合運用管理ツールの「<ドメイン名>」-「アプリケーションサーバ」-「Transactionサービス」ツリー)からも実施可能です。詳細については「[Transactionサービスの運用操作](#)」をご参照ください。

4.9. コード変換について

コード変換について説明します。

4.9.1. CORBA通信におけるコード変換

CORBA 通信でのコード変換について説明します。

C++アプリケーションと Java アプリケーション間の CORBA 通信で、日本語データを送受信する場合には、文字化けが発生しないように、適切なコードセットの設定を行う必要があります。C++アプリケーションと Java アプリケーション間ではそのコードセットの設定情報を交換して、どのようにコード変換を行うかを決定しています。個々の文字列の内容をチェックしている訳ではありません。そのため、実際に C++アプリケーションがどのような文字列を送受信するかによって、正しい Native コードセット

の設定を行っておく必要があります。例えば、C++アプリケーションでSJISの文字列をデータベースやファイルから読み込んで送受信するのであれば、NativeコードセットもSJISに設定する必要があります。

詳細については、次を参照してください。

Object Broker C++では、運用編「コンフィグレーション」の「11.3.WebOTX Object Broker C++における環境設定」の「コードセットに関するオプション設定」を参照してください

を参照してください。

Object Broker Java では、

WebOTX アプリケーション開発ガイド > 第4部 7.2.Object Broker > Object Broker Java の機能 > 文字コードセットを参照してください。

4.9.2. Webアプリケーションの文字エンコーディングの設定について

Webアプリケーションでは、リクエストやレスポンスの文字エンコーディングをさまざまところで指定することができます。以下に、場所に応じた文字エンコーディングの指定方法を説明いたします。

● Servlet の場合

リクエストパラメータ

Web ブラウザから POST メソッドで送付されるリクエストパラメータは、`javax.servlet.ServletRequest.getParameter()` を使って取得しますが、このパラメータを取得する際の文字エンコーディングは次のメソッドで指定します。

```
javax.servlet.ServletRequest.setCharacterEncoding()
```

このメソッドは、`getParameter()` の前に呼び出す必要があります。

レスポンス

Servlet からクライアントに出力するレスポンスの文字エンコーディングは次のメソッドで指定します。

```
javax.servlet.ServletResponse.setContentType()
```

引数として、“text/html; charset=Shift_JIS” のように指定します。このメソッドは、`PrintWriter` を取得する前に呼び出す必要があります。

● JSP の場合

リクエストパラメータ

Servlet の場合と同様です。

レスポンス

JSP からクライアントに出力するレスポンスの文字エンコーディングは次のように `page` ディレクティブで指定します。

```
<%@ page contentType="text/html; charset=Shift_JIS" %>
```

また、JSP ファイル自体がどの文字エンコーディングで保存されているかを指定する要素として次のものがあります。

```
<%@ page pageEncoding="Shift_JIS" %>
```

もし、上記の指定をしていない場合、デフォルトの “ISO-8859-1” となりますが、このデフォルト値を変更する事も可能です。変更するには、次のように `otxadmin` コマンドを使って `default-encoding` プロパティで指定します。ドメインを起動したあと、次のようにコマンドを実行してください。

```
otxadmin> set server.web-container.property.default-encoding=Shift_JIS
```

また、JSP を事前コンパイルする場合のデフォルト文字エンコーディングは `nec-web.xml` で次のように指定します。

```
<jsp-config>
```

```
<property name="precompileDefaultEncoding" value="Shift_JIS"/>
```

</jsp-config>

なお、デフォルト文字エンコーディングの指定は JSP をコンパイルする時に決定します。事前コンパイルした場合は nec-web.xml の指定が、そうでない場合は default-encoding プロパティ指定が有効になります。事前コンパイルでない場合、指定を変更するには Web アプリケーションを再配備し、JSP を再コンパイルする必要があります。

4.10. jconsoleを利用したJavaプロセスの監視・管理

JDK 5.0 および JDK6.0 に付属している jconsole^{*}を利用して、WebOTX 上で動作している JVM や、その上で動作するアプリケーションに関するパフォーマンス情報、およびリソース消費情報を参照する方法について説明します。

※jconsole : JMX(Java Management Extensions)仕様に準拠した監視ツール

なお、jconsole に関する詳しい説明は、使用するバージョンに応じて Sun の公式サイトを参考にしてください。

JDK5.0 向け (<http://java.sun.com/j2se/1.5.0/ja/docs/ja/guide/management/jconsole.html>)

JDK6.0 向け (<http://java.sun.com/javase/ja/6/docs/ja/technotes/guides/management/jconsole.html>)

(注 1)WebOTXを起動しているJavaのバージョンが 1.4 である場合は、本機能を利用することができません。

(注 2)以下のサイトには、jconsole使用時のFAQが情報公開されています。

<http://java.sun.com/javase/ja/6/docs/ja/technotes/guides/management/faq.html#win1>

FAQ「JConsole と JMX リモート管理」の項目 1. および 2. に記載されている項目に該当する場合、本機能は利用することができませんのでご了承下さい。

(注 3)JDK6.0 の環境でWebOTXおよびjconsoleを動作させる場合には、予め\${AS_INSTALL}/lib/endorsedディレクトリ配下に格納されているwojmx-ext.jarファイルをリネームする等して、実行時のパスから除外されるよう対処して下さい。これは、JDK6.0 にて提供される各種ライブラリと本ライブラリファイルのビルドバージョンの相違により、実行時に影響を及ぼすのを回避するためです。この作業に伴うWebOTXおよびjconsoleの機能自体に差異は生じません。再びJDK5.0 の環境でWebOTXおよびjconsoleを動作させる場合には、wojmx-ext.jarファイルを元の状態に戻して下さい。

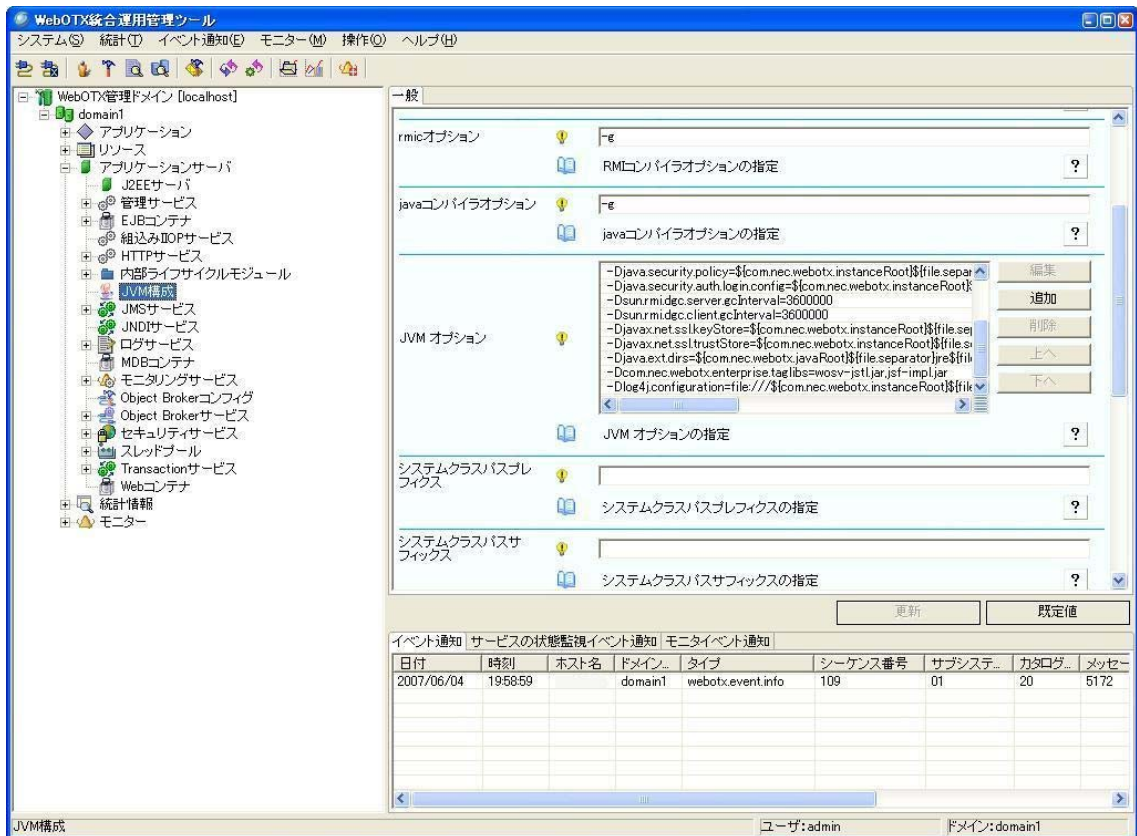
4.10.1. WebOTXに必要な設定

jconsole を WebOTX で利用可能にするための設定方法について説明します。

なお、jconsole は WebOTX 上の JMX エージェントの Java プロセスに対する監視を行う場合と、プロセスグループ上の Java プロセスに対する監視を行う場合^{*}とで設定箇所が異なります。また、jconsole を WebOTX と同じマシン上(ローカル)で動作させるか、異なるマシン(リモート)で動作させるかによっても設定値が異なりますので、それぞれの状況について説明します。

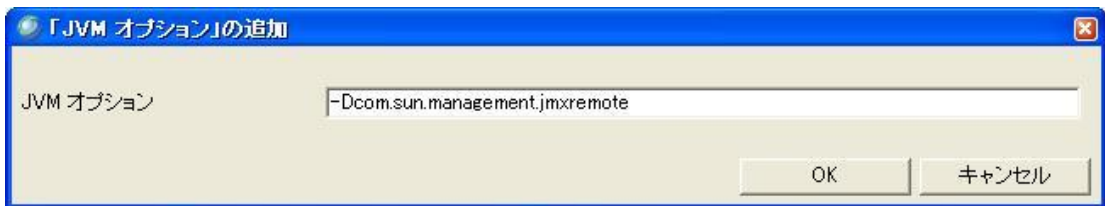
※本設定をプロセスグループに対して行う場合は、プロセスの多重度が 1(シングルプロセス)である場合のみ行って下さい。多重度が 2 以上となっている場合、各プロセスが同じポートを参照することになり、結果的にポートの設定に成功したプロセス以外のプロセスグループ内のプロセスは起動に失敗します。

統合運用管理ツールを使用して設定する場合



JMXエージェントのJavaプロセスを監視するための設定

1. 統合運用管理ツールよりドメインと接続します。
2. ツリービューより「<ドメイン名>」-「アプリケーションサーバ」-「JVM 構成」を選択します。
3. 右側に表示される「JVM オプション」属性の「追加」ボタンを押し、表示される以下の入力画面にて次の Java オプションを指定します(1 つずつ追加して下さい)。



<jconsole を WebOTX と同じマシン(ローカル)で動作させる場合>

-Dcom.sun.management.jmxremote

※Java プロセスが JDK6.0 の環境で動作する場合は、本設定は不要です。

<jconsole を WebOTX と異なるマシン(リモート)で動作させる場合>

-Dcom.sun.management.jmxremote.port=<任意の接続ポート番号>

-Dcom.sun.management.jmxremote.ssl=false

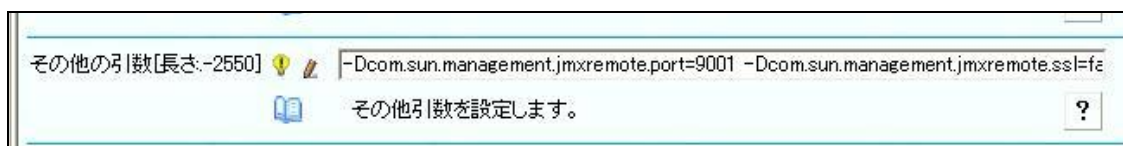
-Dcom.sun.management.jmxremote.authenticate=false

※本設定は接続時のセキュリティ機能を無効にするため、本番環境等では設定しないで下さい。

4. 設定後、「更新」ボタンを押します。設定が完了したら、ドメインを再起動します。

プロセスグループのJavaプロセスを監視するための設定

1. 統合運用管理ツールよりドメインと接続します。
2. ツリービューより「<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>」を選択します。
3. 右側に表示される「JVM オプション」タブ内の「その他の引数」属性に、次の Java オプションを指定します。



<jconsole を WebOTX と同じマシン(ローカル)で動作させる場合>

-Dcom.sun.management.jmxremote

※Java プロセスが JDK6.0 の環境で動作する場合は、本設定ではなく以下のリモート接続用の設定を行って下さい。

<jconsole を WebOTX と異なるマシン(リモート)で動作させる場合>

-Dcom.sun.management.jmxremote.port=<任意の接続ポート番号>

-Dcom.sun.management.jmxremote.ssl=false

-Dcom.sun.management.jmxremote.authenticate=false

※本設定は接続時のセキュリティ機能を無効にするため、本番環境では設定しないで下さい。

4. 設定後、「更新」ボタンを押します。設定が完了したら、プロセスグループを再起動します。

※本設定は、「Java システムプロパティ」タブ内の「Java システムプロパティ」属性にて指定することも可能です。その際は、“-D”は指定せず、また、“=”より前の文字列を「名前」に、“=”より後の文字列を「値」に指定して下さい。

運用管理(otxadmin)コマンドを使用して設定する場合

JMXエージェントのJavaプロセスを監視するための設定

1. 運用管理コマンドを起動し、ドメインにログインします。
(例) otxadmin > login --user admin --password adminadmin --port 6212

2. 次に示すように、Java オプションを設定するためのコマンドを実行します。

<jconsole を WebOTX と同じマシン(ローカル)で動作させる場合>

```
otxadmin > create-jvm-options -Dcom.sun.management.jmxremote
```

※Java プロセスが JDK6.0 の環境で動作する場合は、本設定は不要です。

<jconsole を WebOTX と異なるマシン(リモート)で動作させる場合>

```
otxadmin > create-jvm-options -Dcom.sun.management.jmxremote.port=<任意の接続ポート番号>
```

```
otxadmin > create-jvm-options -Dcom.sun.management.jmxremote.ssl=false
```

```
otxadmin > create-jvm-options -Dcom.sun.management.jmxremote.authenticate=false
```

※本設定は接続時のセキュリティ機構を無効にするため、本番環境では設定しないで下さい。

3. 設定が完了したら、ドメインを再起動します。

プロセスグループのJavaプロセスを監視するための設定

1. 運用管理コマンドを起動し、ドメインにログインします。
(例) otxadmin > login --user admin --password adminadmin --port 6212

2. 次に示すように、Java オプションを設定するためのコマンドを実行します。

<jconsole を WebOTX と同じマシン(ローカル)で動作させる場合>

```
otxadmin > set tpsystem.applicationGroups.<アプリケーショングループ名>.processGroups.<プロセスグループ名>
```

```
>.otherArguments=-Dcom.sun.management.jmxremote
```

※Java プロセスが JDK6.0 の環境で動作する場合は、本設定ではなく以下のリモート接続用の設定を行って下さい。

い。

<jconsole を WebOTX と異なるマシン(リモート)で動作させる場合>

```
otxadmin > set tpsystem.applicationGroups.<アプリケーショングループ名>.processGroups.<プロセスグループ名>.otherArguments="-Dcom.sun.management.jmxremote.port=<任意の接続ポート番号>
```

```
-Dcom.sun.management.jmxremote.ssl=false -Dcom.sun.management.jmxremote.authenticate=false"
```

※既に他のオプションが指定されている場合は、併せてそのオプションも再度指定します。

3. 設定が完了したら、プロセスグループを再起動します。

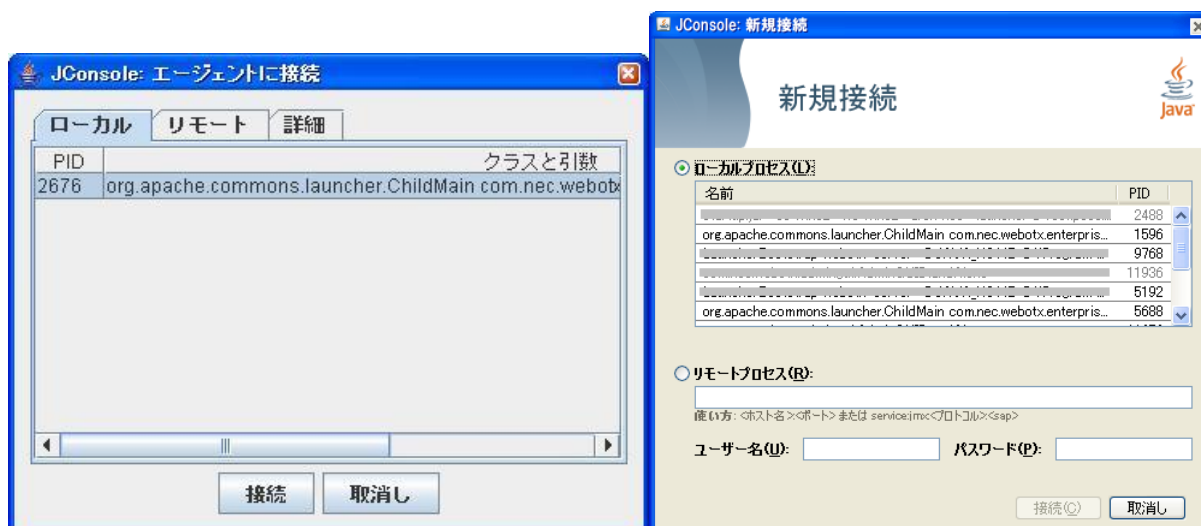
※本設定は、add-pg-javasytem-property コマンドにて指定することも可能です。その際は、“-D”は指定せず、また、“=”より前の文字列を-javasytemprop オプションに、“=”より後の文字列を-value オプションに指定して下さい。

4.10.2. WebOTXで動作するJavaプロセスへの接続

ドメインもしくはプロセスグループの再起動後、jconsole を起動します。上記設定が正しく行われている場合、起動後の画面で接続パラメータを指定することで WebOTX 内の JVM 情報を監視できるようになります。但し、jconsole を WebOTX と同じマシン(ローカル)で動作させるか、異なるマシン(リモート)で動作させるかによってパラメータ指定画面が異なります。以下に接続例を示します。

jconsole を WebOTX と同じマシン(ローカル)で動作させている場合

※JDK6.0 の環境でプロセスグループの各プロセスに接続する場合は、後述のリモート接続による作業を行って下さい。



JDK5.0 付属 jconsole 画面

JDK6.0 付属 jconsole 画面

「ローカル」タブ(JDK5.0 版) / 「ローカルプロセス」(JDK6.0 版)をクリックし、接続候補リストの中から「クラスと引数」(JDK5.0 版) / 「名前」(JDK6.0 版)列に“org.apache.commons.launcher.ChildMain ...”(JMX エージェントへの接続)もしくは空白(プロセスグループへの接続)が表示されている行を選択し、「接続」ボタンを押します。

jconsole を WebOTX と異なるマシン(リモート)で動作させている場合



JDK5.0 付属 jconsole 画面

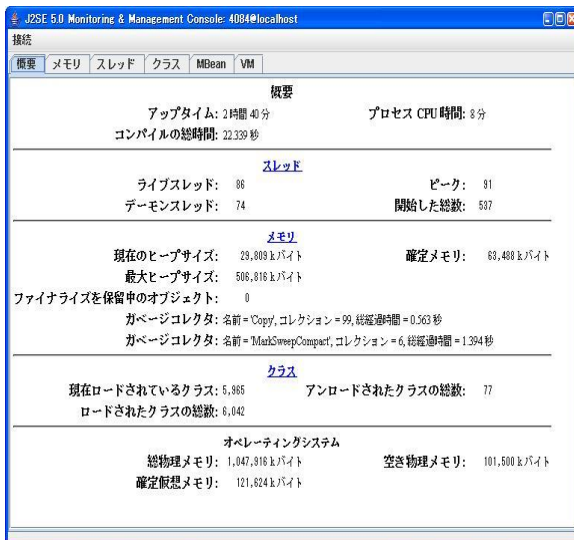


JDK6.0 付属 jconsole 画面

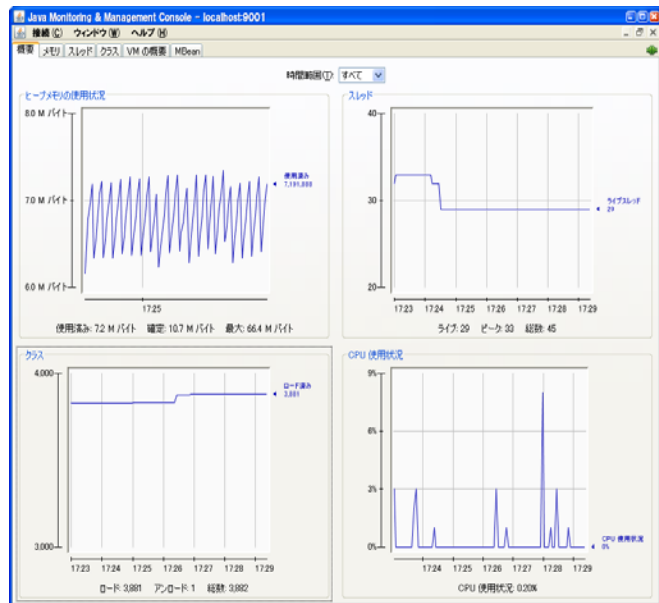
「リモート」タブ(JDK5.0 版) / 「リモートプロセス」(JDK6.0 版)をクリックし、「ホストまたは IP」列に接続先のホスト名または IP アドレスを、「ポート」には WebOTX 側で com.sun.management.jmxremote.port オプションプロパティで指定したポート番号を入力します(JDK5.0 版)。あるいは、入力フィールドに<ホスト名>:<ポート>の形式で入力します(JDK6.0 版)。本接続例ではクライアント認証機能を無効に設定しているため、「ユーザ名」と「パスワード」には何も入力しません。

パラメータ指定後、「接続」ボタンを押します。

接続に成功すると、以下のような画面が表示されます。



JDK5.0 付属 jconsole 画面



JDK6.0 付属 jconsole 画面

初期表示画面では「概要」タブが選択されており、監視対象の JVM に関するパフォーマンス情報とリソース消費情報、および OS に関するリソース情報が表示されます。

この他の画面では、カテゴリ毎に情報が細分化され、それらの監視項目がグラフィカルに表示されます。以下に、各表示項目について簡潔に説明します。

「メモリ」：JVM 内でのメモリの消費状況や、メモリプールに関する情報が表示されます。

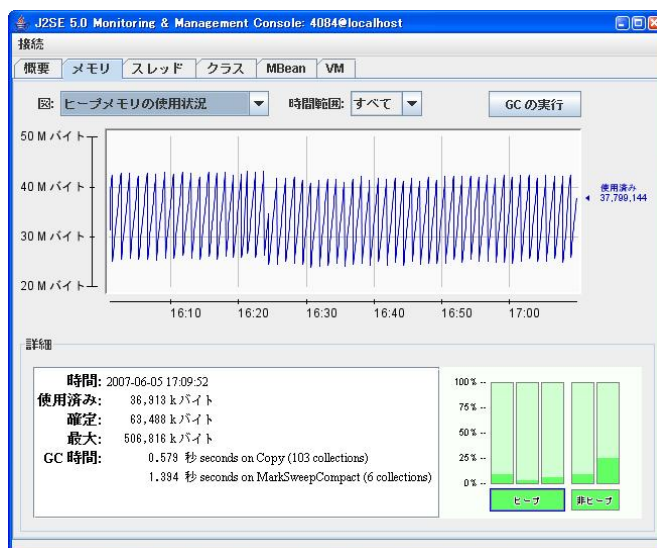
「スレッド」：JVM が生成するスレッドに関する情報が表示されます。WebOTX が生成するスレッドや、ユーザが配備したアプリケーションなどのスレッドに関する情報も併せて表示されます。

「クラス」：JVM 内で使用されるクラスのロードに関する情報が表示されます。

「MBean」：JVM 内に存在する MBean (Management Bean) に関する情報が表示されます。ユーザが作成した MBean に関する情報も併せて表示されます。なお、WebOTX が生成する MBean の情報を参照することはできません。*

「VM (の概要)」：JVM に関する監視項目に加え、クラスパスなどの設定情報が表示されます。WebOTX やユーザが追加設定したクラスパスなどの情報も併せて表示されます。

(※)WebOTX では、高機能化の実現にあたり、独自に JMX 仕様を拡張させた MBean を使用しています。そのため、その MBean を示すツリーノードは画面左側に表示されますが、ノードをクリックした場合、jconsole では対応しきれず以下のエラーメッセージが表示されます。



これらの MBean は通常システム内部でのみ使用するため、ユーザが参照する必要はありません。これらの MBean は、システム内部および「統合運用管理ツール」や「運用管理コマンド」などから参照されます。

4.10.3. 接続できない場合の対処

ご使用の環境によっては、「4.10.1. WebOTX に必要な設定」での作業を行っても、jconsole で WebOTX の Java プロセスがうまく監視できない場合があります。その時は、以下の対処法によって接続可能となるかを確認して下さい。

● コマンドからのドメイン起動

Windows 環境で、サービスから WebOTX Agent Service によってドメインを立ち上げている場合、監視できない場合があります。その際は、一旦 WebOTX Agent Service を停止し、以下の運用管理コマンドによってドメインを立ち上げてから監視を試みて下さい。

```
otxadmin > start-domain <ドメイン名>
```

● JVM オプションの追加

上記コマンドからのドメイン起動でも jconsole によって監視できない場合、さらに次の JVM オプションを追加して下さい(ドメインを起動した状態で、かつ login コマンドでログイン操作を行ってから実施します)。

```
otxadmin > create-jvm-options -Djavax.management.builder.initial=com.nec.webotx.jmx.mbeanserver.JmxMBeanServerBuilder
```

オプション追加後、ドメインを再起動し、jconsole から接続可能となるかを確認して下さい。

なお、jconsole による監視終了後は、上記 JVM オプション、および「4.10.1. WebOTX に必要な設定」で行った設定は全て解除して下さい。

5. リソースの追加・削除

リソースの追加・削除は統合運用管理ツールもしくは運用管理コマンド(otxadmin)で行うことができます。

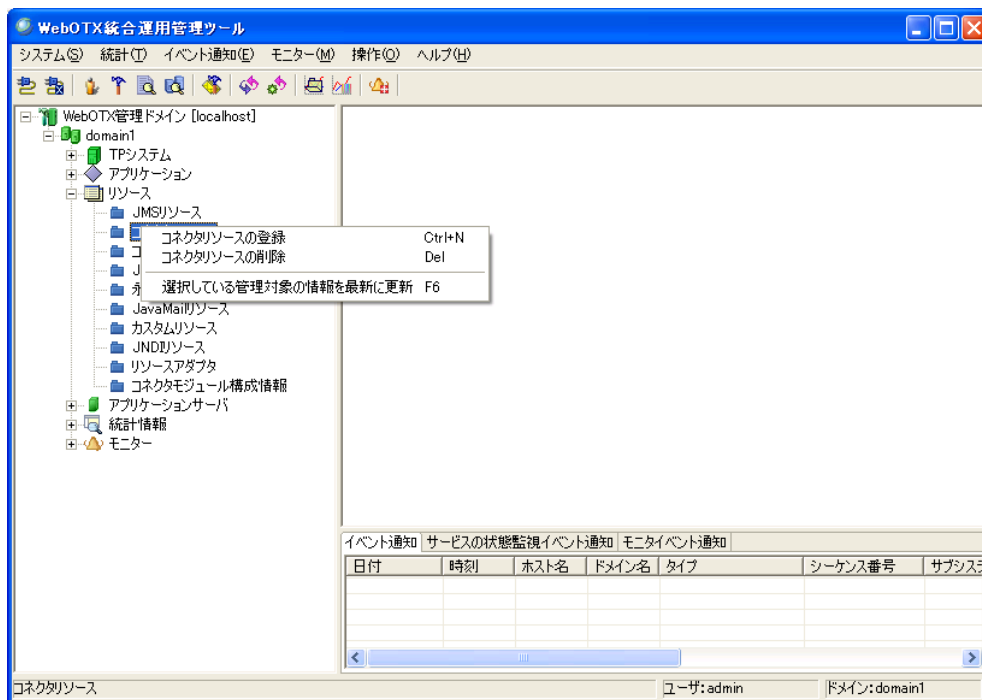
提供リソース一覧

WebOTX で提供するリソースの一覧を以下に示します。なお個々のリソースについての詳細は運用編「MO 定義リファレンス」を参照してください。

リソース	説明
admin-object-resource	Adminオブジェクトリソース server.resources.admin-object-resource.resource-name
connector-connection-pool	コネクションプール server.resources.connector-connection-pool.resource-name
connector-resource	コネクタリソース server.resources.connector-resource.resource-name
custom-resource	カスタムリソース server.resources.custom-resource.resource-name
external-jndi-resource	JNDI リソース server.resources.external-jndi-resource.resource-name
jdbc-datasource	JDBC データソース server.resources.jdbc-datasource.resource-name
jms-connection-factory	JMS コネクションファクトリリソース server.resources.jms-resource.jms-connection-factory.resource-name
jms-logical-destination	JMS 送信先リソース server.resources.jms-resource.jms-logical-destination.resource-name
jms-resource	JMS リソース server.resources.jms-resource
persistence-manager-factory-resource	永続化リソース server.resources.persistence-manager-factory-resource.resource-name
resource-adapter-config	リソースアダプタ server.resources.resource-adapter-config.resource-name
resources	リソースの親 MO server.resources

統合運用管理ツールから設定

1. 統合運用管理ツールよりドメインと接続します。
2. ツリービューより「リソース」を選択します。
3. さらに追加もしくは削除したいリソースの種別を選択します。
4. 右クリックを行い追加する場合「XXXX の登録」を、削除する場合「XXXX の削除」を実行してください。
5. 登録の場合、表示された画面に登録するのに必要な情報を設定して「実行」ボタンをクリックしてください。削除の場合削除するリソースを選択し「実行」ボタンをクリックしてください。



運用管理コマンドから設定

コマンドで以下のリソースを追加したり、削除したりすることが出来ます。なお各コマンドの詳細については運用編「運用管理コマンドリファレンスマニュアル」を参照してください。

コマンド	説明
add-resources	指定された XML ファイルへリソースを登録
create-connector-resource	指定された JNDI 名でのリソースの登録
create-custom-resource	カスタムリソースの登録
create-javamail-resource	JavaMail リソースの登録
create-jdbc-datasource	JDBC データソースの登録
create-jms-resource	JMS リソースの登録
create-jndi-resource	JNDI リソースの登録
create-persistence-resource	永続化リソースの登録
delete-connector-resource	指定のコネクタリソースの削除
delete-custom-resource	カスタムリソースの削除
delete-javamail-resource	JavaMail リソースの削除
delete-jdbc-datasource	JDBC データソースの削除
delete-jms-resource	JMS リソースの削除
delete-jndi-resource	JNDI リソースの削除
delete-persistence-resource	永続化リソースの削除

6. バックアップ/リストア

作成したドメインの環境をバックアップ、リストアする手順について説明します。

6.1. ドメイン環境の自動バックアップ

WebOTX では、ファイル書き込み中の電源障害による設定ファイルの破損や、設定変更のミスによりドメインが正常に起動できなくなった場合に備え、前回正常起動した時の状態を自動でバックアップしています。バックアップからファイルを復元することで前回正常起動した状態まで環境を元に戻すことができます。

注意

- ・前回起動後に配備されたアプリケーションは復元されません。
- ・前回起動後に変更された設定は復元されません。
- ・ログファイルは自動バックアップの対象外のため復元されません。

ドメイン環境の自動バックアップでは、各ドメイン毎の`$(INSTANCE_ROOT)`配下のファイルを以下のディレクトリに自動でバックアップしています。(デフォルトでは、前回3回分の正常起動時のファイルが `files1` ~ `files3` のディレクトリ名で保存されています。)

`$(INSTANCE_ROOT)/backup/files1 ~ files3`

バックアップ世代数の変更を変更する場合は、以下の手順を行ってください。世代数は 0~3 の間で設定できます。以下の例は世代数を 1 に変更する場合です。1 の部分を設定する世代数にあわせて変更してください。

例: `otxadmin > set domain.num-config-backup-generations=1`

V6.50.02 より、ドメイン環境の自動バックアップされるディレクトリを指定することができます。 `domain.backup-option` に値を設定することで、バックアップ対象を指定することができます。設定できる値は `all`、または `configs` となります。(デフォルトでは `all` が指定されています。)

以下のように設定を行います。

`otxadmin > set domain.backup-option=all`

または、

`otxadmin > set domain.backup-option=configs`

バックアップされるディレクトリの一覧は以下のようになります。

バックアップされるディレクトリ	<code>domain.backup-option=all</code> の場合	<code>domain.backup-option=configs</code> の場合
applications	○	×
autodeploy	○	×
backup	×	×
bin	○	×
config	○	○

docroot	○	×
generated	○	×
jbi(Enterprise Service Bus をご利用の場合)	○	○
lib	○	×
logs	×	×
session-store	○	×
stats	○	×
wojms	○	×

以下では、前回正常起動時の状態にドメイン環境を戻す方法について説明します。
(作業の前には、次章のバックアップ手順を参照してバックアップをとってください。)

①ドメインを停止します。

```

${AS_INSTALL}/bin/otxadmin stop-domain WebOTXAdmin

```

②前回正常起動時のドメイン環境に復元します。

```

${AS_INSTALL}/bin/otxrecovery <ドメイン名>

```

この操作により、上記で指定したドメインの環境(=\${INSTANCE_ROOT}配下)が前回正常起動時の状態に復元されます。復元前の(ドメインの起動に失敗した環境の)各種ファイルは、デフォルトでは\${INSTANCE_ROOT}配下に作成される一時保管領域である failed ディレクトリ内に移動されます。

なお、クラスタ環境構築などにより、ドメイン環境が<WebOTX インストールディレクトリ>/domains 配下に存在しない場合や、ディスク容量不足などの理由で、バックアップ領域や一時保管領域を別のディスク領域に変更する場合は、以下のように指定してください。

```

${AS_INSTALL}/bin/otxrecovery <${INSTANCE_ROOT}の値> </バックアップ格納場所> <一時保管場所>

```

(例) `${AS_INSTALL}\bin\otxrecovery D:\domains\domain1 D:\domains\domain1\backup D:\domains\domain1\failed`

復元処理にあたって次の点に注意をしてください。

ドメイン環境内のファイルをエディタなどで開いている場合は、あらかじめ閉じておいてください。

万一復元処理の途中で処理が失敗した場合は、\${INSTANCE_ROOT}/backup、\${INSTANCE_ROOT}/logs 以外のディレクトリとその中身を全て削除し、\${INSTANCE_ROOT}/backup/files1 配下にあるバックアップファイルを\${INSTANCE_ROOT}配下へ上書きでコピーしてください。

ファイルの削除:

- Windows 版


```
cd ${INSTANCE_ROOT}
del /S /Q .\applications\j2ee-apps\*
del /S /Q .\applications\j2ee-modules\*
del /S /Q .\autodeploy\*
```

:

- HP-UX 版、Solaris 版、Linux 版


```
cd ${INSTANCE_ROOT}
rm -Rf ./applications/j2ee-apps/*
rm -Rf ./applications/j2ee-modules/*
rm -Rf ./autodeploy/*
```

:

ファイルのコピー:

- Windows 版


```
cd ${INSTANCE_ROOT}
xcopy /S /E /Y .\backup\files1\* .\
```

- HP-UX 版、Solaris 版、Linux 版


```
cd ${INSTANCE_ROOT}
cp -pR ./backup/files1/* ./
```

前回のドメイン正常起動時におけるバックアップ処理に失敗している場合、復元処理を行っても、前回より前のドメイン環境に復元される可能性があります。ドメイン正常起動時におけるバックアップ処理に成功している場合は、\${INSTANCE_ROOT}/backup/files1～files3 配下に recovery_success というファイルが作成されています。そのファイルが存在しないバックアップ環境を \${INSTANCE_ROOT} 配下へコピーしても、ドメイン起動の保障は致しかねます。

6.2. バックアップ手順

環境の退避は、バージョンアップ前の環境を保存したり、環境の移行の際の移行前の環境を保存するといった際に利用します。バージョンアップの際には、旧バージョンの WebOTX をアンインストールする前に行ってください。

以下では、各々の OS についての環境の退避方法について説明します。

6.2.1. Windows

WebOTX Web/Standard-J Edition

Windows 版での環境の退避は次のファイルをバックアップしてください。

- <WebOTX インストールディレクトリ>\%config 配下の全ファイル
- <WebOTX インストールディレクトリ>\%domains 配下の全ファイル
- <WebOTX インストールディレクトリ> 配下のドメイン設定ファイル(*.properties)

6.2.2. HP-UX版

HP-UX 版での環境の退避は次のファイルをバックアップしてください。

- /opt/WebOTX/config 配下の全ファイル
- /opt/WebOTX/domains 配下の全ファイル

- /opt/WebOTX 配下のドメイン設定ファイル(*.properties)

6.2.3. Solaris

Solaris 版での環境の退避は次のファイルをバックアップしてください。

- /opt/WebOTX/config 配下の全ファイル
- /opt/WebOTX/domains 配下の全ファイル
- /opt/WebOTX/ 配下のドメイン設定ファイル(*.properties)

6.2.4. Linux

Linux 版での環境の退避は次のファイルをバックアップしてください。

- /opt/WebOTX/config 配下の全ファイル
- /opt/WebOTX/domains 配下の全ファイル
- /opt/WebOTX/ 配下のドメイン設定ファイル(*.properties)

6.3. 同一環境にリストアする手順

バックアップした環境から WebOTX の環境をリストア(復元)します。

以下では、各々の OS についての環境のリストアする手順について説明します。

ただし、同一環境にリストアする場合でも、ドメインの構成に変更がある環境やアンインストール後のリストアの場合は、6.4異なる環境にリストアする手順 で示した手順を用いてOSのサービス登録情報などを復元する必要があります。

6.3.1. Windows

バックアップした次のファイルを元の場所にコピーしてください。

- <WebOTX インストールディレクトリ>%config 配下の全ファイル
- <WebOTX インストールディレクトリ>%domains 配下の全ファイル
- <WebOTX インストールディレクトリ> 配下のドメイン設定ファイル(*.properties)

WebOTX 開発環境

バックアップした次のファイルを元の場所にコピーしてください。

- <WebOTX のインストールディレクトリ>%Studio%workspace

6.3.2. HP-UX版

バックアップした次のファイルを元の場所にコピーしてください。

- /opt/WebOTX/config 配下の全ファイル

- /opt/WebOTX/domains 配下の全ファイル
- /opt/WebOTX/ 配下のドメイン設定ファイル(*.properties)

6.3.3. Solaris

バックアップした次のファイルを元の場所にコピーしてください。

- /opt/WebOTX/config 配下の全ファイル
- /opt/WebOTX/domains 配下の全ファイル
- /opt/WebOTX/ 配下のドメイン設定ファイル(*.properties)

6.3.4. Linux

バックアップした次のファイルを元の場所にコピーしてください。

- /opt/WebOTX/config 配下の全ファイル
- /opt/WebOTX/domains 配下の全ファイル
- /opt/WebOTX/ 配下のドメイン設定ファイル(*.properties)

6.4. 異なる環境にリストアする手順

WebOTX の環境を別のマシンに移行する場合の手順は以下のとおりです。ただし、移行前のマシンと移行後のマシンで別々の WebOTX のライセンスを利用する場合は、移行前のマシンの WebOTX を削除する必要はありません。

①移行前のマシンの環境の退避

バックアップ手順 を参照してください。

②移行前のマシンの環境の削除

移行前のマシンの WebOTX をアンインストールしてください。UNIX 版の場合は、設定したライセンスも削除してください。

③移行後のマシンへのWebOTXのインストール

移行するマシンに WebOTX をインストールしてください。

④設定ファイルの復元

移行前にバックアップした以下の設定ファイルを移行後のマシンに復元します。(domains 配下のファイルについては、ここではリストアしないでください。)

- 1.Windows 版
<WebOTX インストールディレクトリ>%config
<WebOTX インストールディレクトリ>%*.properties

2.HP-UX 版、Solaris 版、Linux 版
/opt/WebOTX/config
/opt/WebOTX/*.properties

⑤移行後の環境設定

移行後のマシンの WebOTX インストールフォルダが異なる場合は以下のファイルを修正します。

1.Windows 版

<WebOTX インストールディレクトリ>%config%otxenv.bat に記述されている各ディレクトリ設定を移行後の設定に合わせ修正します。

```
set AS_ANT=C:%WebOTX%lib%ant
set AS_ANT_LIB=C:%WebOTX%lib%ant%lib
set AS_WEBSERVICES_LIB= C:%WebOTX%lib
set AS_IMQ_LIB= C:%WebOTX%wojms%lib
set AS_IMQ_BIN= C:%WebOTX%wojms%bin
set AS_CONFIG= C:%WebOTX%config
set AS_INSTALL= C:%WebOTX
set AS_JAVA=C:%j2sdk1.4.2_14
set AS_ACC_CONFIG= C:%WebOTX%lib%appclient%nec-acc.xml
set AS_ICU_LIB= C:%WebOTX%lib
set AS_DEF_DOMAINS_PATH= C:%WebOTX%domains
set AS_OSPI_LIB= C:%WebOTX%ObjectBroker%lib
set AS_BOOT_DOMAIN= WebOTXAdmin
```

2.HP-UX 版、Solaris 版、Linux 版

/opt/WebOTX/config/otxenv.conf に記述されている各ディレクトリ設定を移行後の設定に合わせ修正します。

```
AS_ANT="/opt/WebOTX/lib/ant"
AS_ANT_LIB="/opt/WebOTX/lib/ant/lib"
AS_WEBSERVICES_LIB="/opt/WebOTX/lib"
AS_IMQ_LIB="/opt/WebOTX/wojms/lib"
AS_IMQ_BIN="/opt/WebOTX/wojms/bin"
AS_CONFIG="/opt/WebOTX/config"
AS_INSTALL="/opt/WebOTX"
AS_JAVA="/usr/java/j2sdk1.4.2_14"
AS_ACC_CONFIG="/opt/WebOTX/lib/appclient/nec-acc.xml"
AS_ICU_LIB="/opt/WebOTX/lib"
```

```
AS_DEF_DOMAINS_PATH="/opt/WebOTX/domains"  
AS_OSPI_LIB="/opt/ObjectSpinner/lib"  
AS_BOOT_DOMAIN="WebOTXAdmin"
```

⑥ドメインの作成

ドメインを作成します。ドメインを作成することでOSのサービス登録情報などを復元します。

WebOTXインストールディレクトリ配下に先ほど復元した、ドメイン設定ファイル(`xxx.properties`)ファイルを使ってドメインを作成します。以下のコマンドを実行してください。ドメインが複数ある場合はすべてのドメインについて `create-domain` コマンドを実行します。また、インストール時にデフォルトで作成される `domain1` を利用しない場合は `delete-domain` コマンドにて削除します。

```
${AS_INSTALL}/bin/otxadmin start-domain WebOTXAdmin  
${AS_INSTALL}/bin/asant -buildfile setup.xml -Ddomain.name=domain1 delete-domain      (※domain1 未使用時のみ)  
${AS_INSTALL}/bin/asant -buildfile setup.xml -Dfile=xxx.properties create-domain  
...  
${AS_INSTALL}/bin/otxadmin stop-domain WebOTXAdmin
```

⑦ドメイン環境を復元

移行前にバックアップした `domains` 配下のファイルを移行後のマシンに上書きで復元します。

- 1.Windows 版
 <WebOTX インストールディレクトリ>\domains
- 2.HP-UX 版、Solaris 版、Linux 版
 /opt/WebOTX/domains

6.5. ドメイン構成ファイル(`domain.xml`と`domains-config.xml`)の自動復旧機能

WebOTX では、これまでに紹介したバックアップ機能を更に強化し、特にドメインの起動を妨げるような状況に陥るのをできるだけ回避するよう、ドメインの構成情報を格納する 2 つの重要なファイル(`domain.xml`、および `domains-config.xml`)については、更新時にリアルタイムにバックアップを取るようになっています。そして、ファイル破損等の万一の障害の際には自動でバックアップファイルから復旧するよう強化しています。これにより、障害の影響でドメインが正常に起動しなくなったといったような問題が大幅に解決されます。

自動復旧機能の対象となるファイルは以下の場所に格納されています。

```
${INSTANCE_ROOT}/config/domain.xml  
${INSTANCE_ROOT}/config/domains-config.xml (管理ドメインのみ)
```

これらのファイルに対して本機能が正常に実行された場合は、`webotx_agent.log` に以下のようなメッセージが出力されます。

(例) ドメイン構成ファイル `domain.xml` が不正なため読み込みに失敗しましたが、バックアップファイルによってリカバリされ、正常に読み込まれました。

なお、これらのバックアップファイルは同じディレクトリ内に `“backup”` という拡張子で必要時に作成されますので、以下の事項に

ご注意願います。

注意

- ・何らかの理由でバックアップファイルの作成に失敗していた場合は、自動復旧機能の対象となりません。
- ・指定の位置にあるバックアップファイルを他のディレクトリに置いた場合等は、自動復旧機能の対象となりません。
- ・対象となるファイルのディレクトリ内には、“.backup”拡張子を付けてオリジナルのバックアップファイルを格納しないで下さい。