



WebOTX と外部 Web サーバの連携設定

WebOTX V6,
WebOTX V7,
WebOTX V8 編

2009.01.16

第 2.0 版

前書き

【本書の位置付け】

本書は、WebOTX Web コンテナと外部 Web サーバの連携を行う際に必要となる作業手順を解説したガイドブックです。
対象とする WebOTX のバージョンとエディションは以下の通りです。

WebOTX Web Edition V6.x
WebOTX Standard-J Edition V6.x
WebOTX Standard Edition V6.x
WebOTX Enterprise Edition V6.x
WebOTX Application Server Web Edition V7.x
WebOTX Application Server Standard-J Edition V7.x
WebOTX Application Server Standard Edition V7.x
WebOTX Application Server Enterprise Edition V7.x
WebOTX Application Server Web Edition V8.1
WebOTX Application Server Standard-J Edition V8.1

【対象読者】

本書は、既に WebOTX をインストール済みの環境で、外部 Web サーバとの連携設定を行う方を対象にしています。
以下のいずれかに該当する場合は、外部 Web サーバとの連携設定が必要です。

- WebOTX Web サーバ以外の外部 Web サーバを使用する場合
- WebOTX インストールマシンと使用する外部 Web サーバのマシンが別マシンである場合
- WebOTX V6.x を使用する場合

※V7 以降の WebOTX をインストールしたマシンで WebOTX Web サーバを使用する場合、インストール時に自動で連携設定が行われるため、連携設定は不要です。

【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 を限定しない箇所では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

目次

1. 全体概要	1
2. プラグインの種類と各バージョンでの提供機能	3
3. WebOTX Web サーバとの連携設定	5
3.1. Windows での連携設定	5
3.1.1. WebOTX Web サーバへの連携情報の設定	5
3.1.2. WebOTXドメインへの外部 Web サーバ連携の設定	7
3.1.3. Web コンテナとは別マシン上の Web サーバを使う場合	8
3.2. Linux での連携設定	10
3.2.1. WebOTX Web サーバへの連携情報の設定	10
3.2.2. WebOTXドメインへの外部 Web サーバ連携の設定	12
3.2.3. Web コンテナとは別マシン上の Web サーバを使う場合	13
3.3. 設定の確認	14
4. IIS との連携設定	17
4.1. WebOTXドメインへの外部 Web サーバ連携の設定	17
4.2. 外部 Web サーバへの WebOTX 連携情報の設定前の作業	18
4.3. 外部 Web サーバへの WebOTX 連携情報の設定	21
4.4. Web コンテナとは別マシン上の IIS を使う場合	22
4.5. IIS の設定	23
4.5.1. 基本認証の設定	23
4.5.2. Web サービス拡張の設定	23
4.6. 設定の確認	24
4.7. 連携設定の解除	26
5. 詳細設定	27
5.1. 動的反映	27
5.1.1. 動的反映オフの設定	28
5.1.2. 動的反映オプション query once の設定	29
5.1.3. 設定ファイルの更新	29
5.2. ロードバランス設定	31
5.2.1. IOP プラグインでのロードバランス設定	31
5.2.2. Web サーバプラグインでのロードバランス設定	32
5.3. ロードバランス利用時のセッションの維持	34
5.3.1. セッションレプリケーションの設定	34
5.3.2. JNDI サーバの多重化	34
5.4. エラードキュメントの概要	35
5.4.1. WebOTX Web サーバ(Apache)のエラードキュメント	35
5.4.2. プラグインのエラードキュメント	36
5.4.3. Web コンテナのエラードキュメント	36

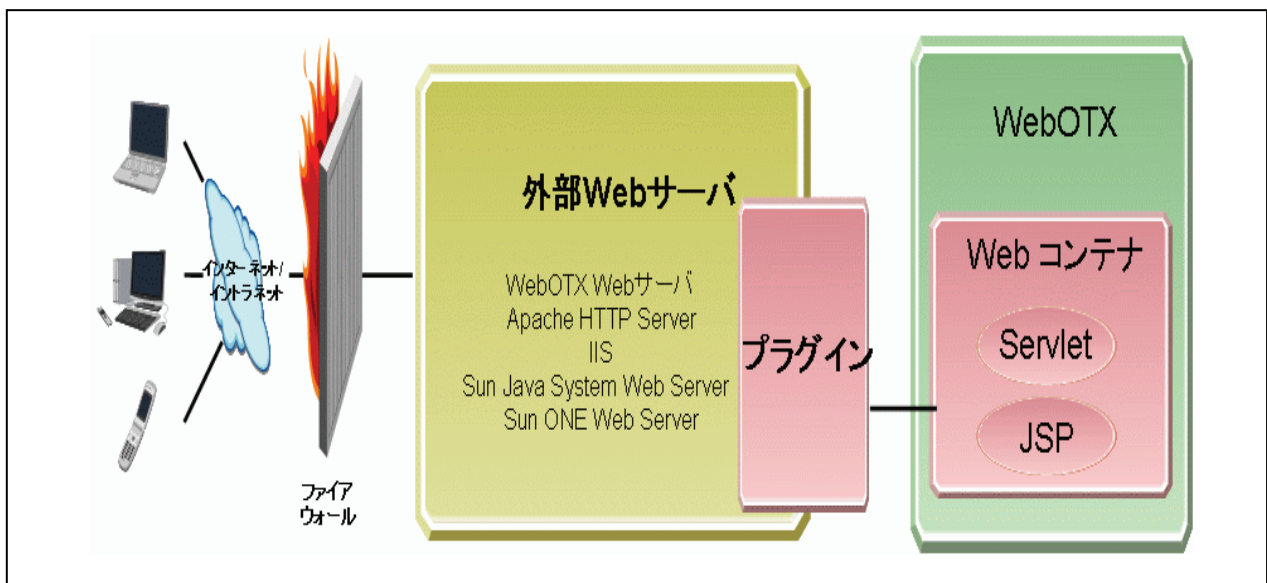
5.4.4.	カスタマイズの併用.....	36
5.5.	エラードキュメントのカスタマイズ	39
5.5.1.	WebOTX Web サーバ(Apache)でのカスタマイズ	39
5.5.2.	IOP プラグインでの HTTP エラー画面カスタマイズ.....	41
5.5.3.	IOP プラグインでの CORBA エラー画面カスタマイズ.....	42
5.5.4.	Web コンテナでのカスタマイズ.....	44
5.6.	連携設定の自動化	45
6.	ライセンスについて.....	48
7.	注意事項	49

1. 全体概要

本章では、連携設定の概要と、この資料の構成について説明します。

WebOTX では、クライアントからの HTTP/HTTPS リクエストを受け付けるサーバとして、Web コンテナ内蔵の Web サーバを提供しています。また、サーブレットや JSP が動作する機能層の Web コンテナと、WebOTX の外部で動作する Web サーバとを連携させることも可能です。

Webコンテナと外部Webサーバを連携させることにより、HTTP/HTTPS サービスは外部 Web サーバとして公開し、HTTP リクエストがサーブレットのコンテキストであった場合は、外部 Web サーバがプラグインを通じて WebOTX 内部の Web コンテナへリクエストを転送して、Web コンテナで処理するようになります。



本資料では、外部 Web サーバと連携する際に必要となる作業手順を解説しています。外部 Web サーバとの連携設定の詳細な手順については、主に利用されている WebOTX Web サーバと Internet Information Services(IIS)を取り上げて解説しています。

WebOTX では、WebOTX にバンドルされている Apache HTTP Server ベースの WebOTX Web サーバだけでなく、IIS や、Apache HTTP Server 1.3.x、2.0.x、Sun Java System Web Server 6.1、Sun ONE Web Server 6.0 以降と連携させての動作もサポートしています。

構成

この資料は「1. 全体概要」、「2. プラグインの種類と各バージョンでの提供機能」、「3. WebOTX Web サーバとの連携設定」、「4. IIS との連携設定」、「5. 詳細設定」、「6. ライセンスについて」、「7. 注意事項」から成り立っています。以下に各項目の概要を説明します。

「1. 全体概要」

ここでは連携設定の概要とこの資料の構成について説明します。

「2.プラグインの種類と各バージョンでの提供機能」

プラグインの種類と各プラグインのバージョンごとの提供機能について説明します。使用している WebOTX のプラグインのバージョンを確認する場合や、プラグインで利用できる機能を確認する場合はこの項目を参照してください。

「3. WebOTX Web サーバとの連携設定」

WebOTX Web サーバと連携設定を行う手順について説明します。Web コンテナと連携する外部 Web サーバに WebOTX にバンドルされている WebOTX Web サーバを使用する場合は、この項目を参照してください。

「4. IIS との連携設定」

Internet Information Service (IIS) と連携設定を行う手順について説明します。Web コンテナと連携する外部 Web サーバに IIS を使用する場合は、この項目を参照してください。

「5. 詳細設定」

「5.1 動的反映」、「5.2 ロードバランス設定」、「5.3 エラードキュメントの概要」、「5.4 エラードキュメントのカスタマイズ」、「5.5 連携設定の自動化」、「5.6 セッションレプリケーションの設定」について説明します。

「5.1 動的反映」

動的反映の概要、動的反映オフの設定、動的反映 query once オプションについて説明します。動的反映について知りたい場合や、動的反映をオフにする場合、動的反映 query once オプションを使用する場合は、この項目を参照してください。

「5.2 ロードバランス設定」

ロードバランス設定を行う手順について説明します。ロードバランス設定を行う場合は、この項目を参照してください。

「5.3 ロードバランス利用時のセッションの維持」

ロードバランス時にセッションを維持するための、セッションレプリケーションの設定方法について説明します。セッションレプリケーションを行う場合や、セッション情報の多重化を行う場合は、この項目を参照してください。

「5.4 エラードキュメントの概要」

Apache のエラードキュメント・IIOP プラグインのエラードキュメント・Web コンテナのエラードキュメントの概要について説明します。

「5.5 エラードキュメントのカスタマイズ」

エラードキュメントのカスタマイズ方法について説明します。エラードキュメントをカスタマイズする場合はこの項目を参照してください。

「5.6 連携設定の自動化」

連携設定の自動化を行う場合に必要となる、連携設定ツールのコマンドラインからの使用方法について説明します。

「6.ライセンスについて」

Web コンテナと Web サーバが別マシン構成である場合のライセンスの考え方について説明します。

「7. 注意事項」

連携設定を行う際の注意事項について説明します。

2. プラグインの種類と各バージョンでの提供機能

プラグインの種類とプラグインの各バージョンでの提供機能について説明します。

提供している機能はプラグインの種類・バージョンによって異なります。WebOTX が提供しているプラグインのバージョンと各バージョンの備考は下表の通りです。

WebOTX のバージョン	プラグインのバージョン	備考
WebOTX AS V8.1	jk 1.2.26	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2)
WebOTX AS V7.1, 7.11 標準修正	jk 1.2.26 (シングルプロセスモード時に使用)	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2)
	IIOP 1.1.0 (マルチプロセスモード時に使用)	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2) CORBA エラー画面カスタマイズ機能(※3) HTTP エラー画面カスタマイズ機能(※3)
WebOTX AS V7.11	jk 1.2.21	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2)
	IIOP 1.1.0	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2) CORBA エラー画面カスタマイズ機能(※3) HTTP エラー画面カスタマイズ機能(※3)
WebOTX AS V7.1	jk 1.2.21 (シングルプロセスモード時に使用)	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2)
	IIOP 1.1.0 (マルチプロセスモード時に使用)	動的反映の OFF 指定機能(※1) query once 機能(※1)(※2) CORBA エラー画面カスタマイズ機能(※3) HTTP エラー画面カスタマイズ機能(※3)

	IIOP 1.0.0 (マルチプロセスモード時に使用)	動的反映の OFF 指定機能(※1) HTTP エラー画面カスタマイズ機能(※3)
WebOTX V6.4 WebOTX V6.31 WebOTX V6.3 標準修正 (V6.31.00 以降) WebOTX V6.22 標準修正 (V6.22.07 以降) WebOTX V6.21 標準修正 (V6.22.07 以降) WebOTX V6.2 標準修正 (V6.22.07 以降)	jk 1.2.15	
WebOTX V6.3	jk 1.2.14	
WebOTX V6.22 WebOTX V6.21 WebOTX V6.2	jk 1.2.8	
WebOTX V6.10.08	jk 2.0.4 (WebOTX Web サーバ2 使用時)	動的反映機能なし(※1)(※4)
	jk 1.2.15	
WebOTX V6.1	jk 2.0.4 (WebOTX Web サーバ2 使用時)	動的反映機能なし(※1)(※4)
	jk 1.2.1-beta-1(上記以外)	

(※1)動的反映とは、Web アプリケーションが配備された事を自動的に外部 Web サーバに反映させる機能で、デフォルトでは有効になっています。詳細は「5.1 動的反映」を参照してください。

(※2)query once 機能を使用すると、動的反映を Web サーバの起動時に1度だけ行います。詳細は「5.1 動的反映」を参照してください。

(※3)CORBA エラー画面カスタマイズ機能、HTTP エラー画面カスタマイズ機能については「5.3 エラードキュメントの概要」「5.4 エラードキュメントのカスタマイズ」を参照してください。

(※4)WebOTX V6.1 で WebOTX Web サーバ2(Apache2)を使用する場合、動的反映は使えません。新規配備後はドメインと Web サーバを再起動してください。

3. WebOTX Web サーバとの連携設定

WebOTX に付属する WebOTX Web サーバとの連携設定について説明します。

Web コンテナ動作マシンと WebOTX Web サーバ動作マシンが別マシンである場合、WebOTX Web サーバのインストールには WebOTX Web Edition のライセンスが必要です。作業に先立って、WebOTX Web サーバをインストールしてください。

※WebOTX Application Server V7.1 以降では、インストール時に WebOTX Web サーバとの連携設定は自動的に行われます。そのため、インストール直後の WebOTX Web サーバと domain1 との連携設定は不要です。

3.1. Windows での連携設定

3.1.1. WebOTX Web サーバへの連携情報の設定

この作業の前に連携設定を行う外部 Web サーバとドメインが停止していることを確認してください。

ここでの作業は全て、Administrator 権限を持つユーザで行ってください。

[スタート] - [プログラム] - [WebOTX] - [環境設定ツール]メニューを選択し、環境設定ツールを起動します。起動された環境設定ツール画面に情報を入力して、Web コンテナと Web サーバの連携設定を行っていきます。

WebOTX Webコンテナ と Webサーバ を連携するために必要な情報を設定してください。

Webコンテナの情報

ドメインディレクトリ(D) C:\WebOTX\domains\domain1 参照(R)

Webサーバの情報

種別

Internet Information Services(IIS)

WebOTX Web Server 1.3(Apache1.3) WebOTX Web Server 2.0(Apache2.0)

Sun Java System Web Server(Sun ONE) 6.x

インストールディレクトリ(W) C:\WebOTX\WebServer2 参照(R)

連携ポート番号(P) 8009

名称(N)

Standard/Enterprise Edition TPモータ上で動作するWebコンテナ(マルチプロセスモード)を利用(I)

Webアプリケーションのコンテキスト動的反映オプション

動的反映 有り(従来どおり(O))

動的反映 1回のみ(Webサーバ起動時に1回のみ動的反映を行う(Q))

注意事項: 「動的反映 1回のみ」のオプションでWebアプリケーションを新規に配備した場合、外部Webサーバを再起動する必要があります。動的反映の詳細については、ヘルプをご覧ください。

OK
キャンセル
ヘルプ(H)

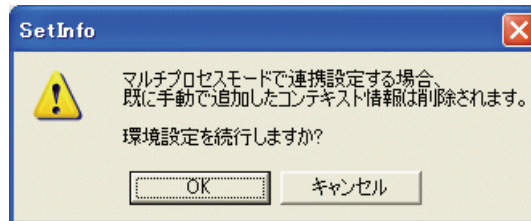
設定項目		説明
Web コンテナ の情報	ドメインディレクトリ	Web サーバと連携する Web コンテナのドメインを指定します。 ドメインの指定例を次に示します。 C:¥WebOTX¥domains¥domain1
Web サーバの 情報	種別	Web コンテナと連携する Web サーバの種別を選択します。 WebOTX Web サーバ 1.3 を使用する場合には「WebOTX WebServer1.3 (Apache 1.3)」を選択してください。 WebOTX Web サーバ 2.0 を使用する場合には「WebOTX WebServer2.0 (Apache2.0)」を選択してください。
	インストールディレクトリ	Web コンテナと連携する Web サーバのインストールディレクトリを指定します。 「WebOTX WebServer1.3 (Apache 1.3)」を選択した場合は、 \${INSTALL}¥WebServer を指定してください。 「WebOTX WebServer2.0 (Apache 2.0)」を選択した場合は、 \${INSTALL}¥WebServer2 を指定してください。 WebOTX Web サーバ 2.0 を指定する場合の指定例を次に示します。 C:¥WebOTX¥WebServer2
	連携ポート番号	Web コンテナと Web サーバとの連携に使用するポート番号を指定します。 domain1 の既定値は「8009」です。 ※下記、「Standard/Enterprise Edition TP モニタ上で動作する Web コンテナを利用」にチェックが入っている場合は、この項目は指定できません。 ※ v8 での既定値は「8099」です。
	Standard/Enterprise Edition TP モニタ上で動作する Web コンテナを利用	Standard/Enterprise Edition の、TP モニタ上で動作する Web コンテナを利用する場合にチェックします。
	Web アプリケーションの コンテキスト名 動的反映オプション	Web アプリケーションのコンテキスト名動的反映オプションを選択します。 「動的反映あり(従来どおり)」を選択した場合、動的反映が常に有効になります。 「動的反映1回のみ(Web サーバ起動時に1回のみ動的反映を行う)」を選択した場合、Web サーバ起動時に1回のみ動的反映が行われます。動的反映の詳細は「5.1 動的反映」を参照してください。 ※このオプションは WebOTX V6.5 以前のバージョンでは利用できません。

環境設定ツールを使用して環境設定を行うと、\${INSTANCE_ROOT}¥config¥WebCont 配下にある workers.properties (シングルプロセスモード時使用)、ior_workers.properties (マルチプロセスモード時使用) が初期化されます。

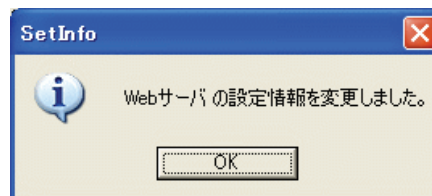
またマルチプロセスモード時は手動で加えたコンテキスト情報も削除されます。

これらのファイルを手動で編集している場合はバックアップを取っておいてください。

入力が完了すれば「OK」ボタンを押下してください。「キャンセル」ボタンを押下すると設定情報を更新せずにツールは終了します。



マルチプロセスモード時は上記ダイアログが表示されるので、よければ「OK」ボタンを押下してください。
「キャンセル」ボタンを押下すると環境設定画面に戻ります。



上記ダイアログが表示されれば、Web コンテナの各種定義ファイル、および Web サーバの設定ファイルに設定情報が反映されま
す。

3.1.2. WebOTX ドメインへの外部 Web サーバ連携の設定

ここでは、WebOTX 内蔵 Web サーバを使用する設定から、内蔵 Web サーバを無効にして外部 Web サーバを使用する設定に変更する方法を説明します。インストール時に WebOTX Web サーバまたは WebOTX Web サーバ 2 を選択した場合、デフォルトでは外部 Web サーバを使用する設定になっているため、設定を変更しなくてもこの手順は不要です。内蔵 Web サーバを使用する設定に変更していた場合、下記手順にしたがって外部 Web サーバを使用する設定に変更してください。

外部 Web サーバを使用する設定になっているか、WebOTX 内蔵 Web サーバを使用する設定になっているかを確認するには運用管理コマンドを起動し、下記のコマンドを実行してください。

```
otxadmin> get server.http-service.virtual-server.server.http-listeners
```

上記コマンドの実行結果が下記のようにになっていた場合、内蔵 Web サーバを使用する設定になっているため、下記手順に従い設定を行ってください。

```
server.http-service.virtual-server.server.http-listeners = http-listener-1,http-listener-2,ajp-listener-1
```

実行結果が下記のようにになっていた場合、既に外部 Web サーバを使用する設定になっているため、ここでの作業は不要です。

```
server.http-service.virtual-server.server.http-listeners = ajp-listener-1
```

連携設定を行うドメインが複数存在する場合は、ドメインごとに「WebOTX ドメインへの外部 Web サーバ連携の設定」作業を行ってください。

運用管理コマンドの起動

最初に、[スタート] - [プログラム] - [WebOTX] - [運用管理コマンド]を起動し、設定を変更するドメインが起動していることを確認します。

```
otxadmin> list-domains
```

ドメインの設定変更

Java ベースの内蔵 Web サーバを利用する設定から、外部 Web サーバを利用する設定に変更します。

管理ユーザとしてログインします。

```
otxadmin> login --user <管理ユーザ名> --password <管理ユーザパスワード>
--port <管理ポート番号>
```

※ 実際には1行で実行してください。

外部 Web サーバを利用する設定に変更します。

```
otxadmin> set server.http-service.virtual-server.server.http-listeners=
"ajp-listener-1"
```

※ 実際には1行で実行してください。

ドメインの停止

ドメインを停止し、ログアウトします。

```
otxadmin> stop-domain <ドメイン名>
otxadmin> logout
otxadmin> exit
```

3.1.3. Web コンテナとは別マシン上の Web サーバを使う場合

Web コンテナが動作するマシンと、Web サーバが動作するマシンが別マシンである場合、Web サーバが読み込む設定ファイルの変更が必要になります。下記手順に従い、変更してください。

マルチプロセスモード時

コンテナの情報は\${INSTANCE_ROOT}¥config¥httpgateway.ior から読み込まれます。下記例を参考に設定を変更してください。

- ① Web サーバが動作するマシンに\${INSTANCE_ROOT}¥config¥iors ディレクトリを作成します。
- ② ①で作成したディレクトリに、Web コンテナが動作するマシン上にある httpgateway.ior をコピーします。httpgateway.ior は Web コンテナ起動時に自動的に作成されます。
- ③ Web サーバが動作するマシンの\${INSTANCE_ROOT}¥config¥WebCont¥ior_workers.properties を開き、下記例を参考に、②でコピーした httpgateway.ior を参照するように ior_def 定義を修正します。(赤字が修正箇所)

```
# TM_PLUGIN_IOR-start
worker.list=otxiiop
otx_domain=C:/WebOTX/domains/domain1
worker.otxiiop.type=otxiiop
worker.otxiiop.default_ior=localdomain
worker.otxiiop.ior_def.localdomain=$(otx_domain)/config/iors/httpgateway.ior
worker.otxiiop.cachesize=150
# V8 では cachesize の代わりに connection_pool_size を使用してください。
# TM_PLUGIN_IOR-end
```

シングルプロセスモード時

`${INSTANCE_ROOT}¥config¥WebCont¥workers.properties` の下記項目を必要に応じて変更してください。

```
worker.ajp13.port=<連携ポート番号>
worker.ajp13.host=<連携するドメインのホストのアドレス>
```

また、Web コンテナと Web サーバが別マシンである構成で動的反映をオフにした場合、新規配備後は設定ファイルの手動更新が必要になります。動的反映・設定ファイルの更新については「5.1 動的反映」を参照してください。

3.2. Linux での連携設定

3.2.1. WebOTX Web サーバへの連携情報の設定

この作業の前に連携設定を行う外部 Web サーバとドメインが停止していることを確認してください。

ここでの作業は全て、root ユーザで行います。root ユーザでログインしていることを確認してください。

環境設定ツールを起動します。

```
root> cd /opt/WebOTX/bin
root> ./setconf.sh
```

起動されたスクリプトの指示にしたがって、Web コンテナと Web サーバの連携設定を行っていきます。

```
Please enter a domain directory.
[/opt/WebOTX/domains/domain1]
```

外部 Web サーバと連携する Web コンテナが動作するドメインのディレクトリを指定してください。

```
Please select Web server.[2]
  1. WebOTX WebServer 1.3 (Apache 1.3)
  2. WebOTX WebServer 2.0 (Apache 2.0)
  3. Sun Java System Web Server (Sun ONE) 6.x
```

連携する外部 Web サーバの種別を選択してください。

使用する WebOTX Web サーバのバージョンに合わせて、「WebOTX WebServer 1.3 (Apache 1.3)」または「WebOTX WebServer 2.0 (Apache 2.0)」を選択してください。

```
Please enter Web server installation directory.
[/opt/WebOTX/WebServer2]
```

連携する外部 Web サーバのインストールディレクトリを指定してください。

「WebOTX WebServer 1.3 (Apache 1.3)」を選択した場合は「/opt/WebOTX/WebServer」を指定してください。

「WebOTX WebServer 2.0 (Apache 2.0)」を選択した場合は「/opt/WebOTX/WebServer2」を指定してください。

```
Would you like to select Multi Process Mode? (y/n) [default y]
You can select Multi Process Mode in Standard/Enterprise Edition.
```

Web コンテナの動作モードを選択してください。

Standard/Enterprise Edition のインストールオプションの Web コンテナ動作モードで、マルチプロセスモードを選択した場合は「y」を選択してください。

```
Please enter port for communication with WebOTX Web Container.[8009]
```

外部 Web サーバと Web コンテナの連携に使用するポート番号を指定してください。

Web サーバがクライアントからのリクエストを受け付けるポート番号ではありません。

※ この項目は、前項の Web コンテナの動作モードの選択で「n」を選択した場合にのみ表示されます。

※ V8 ではポート番号の既定値は 8099 です。

```
The context information added by manually will be deleted.  
(Please Enter Key)
```

手動で加えたコンテキスト情報は削除されます。Enter キーを押下してください。

※ この項目は、前項の Web コンテナの動作モードの選択で「y」を選択した場合にのみ表示されます。

```
Please select context name dynamic reflection option of Web application.[1]  
1. Dynamic reflection option is on.  
2. Dynamic reflection option is once.  
(Note: When a Web server starts, you do dynamic reflection once.  
When you deployed a Web application, you have to reboot a Web server.)
```

Web アプリケーションのコンテキスト名動的反映オプションを選択してください。

「1. Dynamic reflection option is on.」を選択した場合、動的反映が常に有効になります。「2. Dynamic reflection option is once」を選択した場合、Web サーバ起動時に1回のみ動的反映が行われます。動的反映の詳細は「5.1 動的反映」を参照してください。

※ このオプションは WebOTX V6.5 以前のバージョンでは利用できません。

必要な情報の入力が完了すると、確認画面を表示します。

```
The following information are set.  
WebOTX Web Container installation directory:/opt/WebOTX  
Domain directory:/opt/WebOTX/domains/domain1  
Web server:WebOTX WebServer 2.0(Apache 2.0)  
Web server installation directory:/opt/WebOTX/WebServer2  
Use the Web Container executed on the TP monitor:n  
Port for communication with WebOTX Web Container:8009  
Web server plugin dynamic reflection option:on  
Would you like to continue?(y/n)
```

情報に誤りがなければ「y」を入力して環境設定し、誤りがある場合は「n」を入力して処理を中断してください。

中断した場合は、再度 setconf.sh を起動して環境設定を行ってください。

※ Web コンテナの動作モードで「y」を指定した場合は、「Port for communication with WebOTX Web Container」項目は表示されません。


```
Web server setting is completed.
```

上記メッセージが出力されれば環境設定が完了します。

途中でエラーになった場合は、再度 `setconf.sh` を実行してください。

3.2.2. WebOTX ドメインへの外部 Web サーバ連携の設定

ここでは、WebOTX 内蔵 Web サーバを使用する設定から、内蔵 Web サーバを無効にして外部 Web サーバを使用する設定に変更する方法を説明します。インストール時に WebOTX Web サーバまたは WebOTX Web サーバ 2 を選択した場合、デフォルトでは外部 Web サーバを使用する設定になっているため、設定を変更していなければここでの手順は不要です。内蔵 Web サーバを使用する設定に変更していた場合、下記手順にしたがって外部 Web サーバを使用する設定に変更してください。

外部 Web サーバを使用する設定になっているか、WebOTX 内蔵 Web サーバを使用する設定になっているかを確認するには運用管理コマンドを起動し、下記のコマンドを実行してください。

```
otxadmin> get server.http-service.virtual-server.server.http-listeners
```

上記コマンドの実行結果が下記のようにになっていた場合、内蔵 Web サーバを使用する設定になっているため、下記手順に従い設定を行ってください。

```
server.http-service.virtual-server.server.http-listeners = http-listener-1,http-listener-2,ajp-listener-1
```

実行結果が下記のようにになっていた場合、既に外部 Web サーバを使用する設定になっているため、ここでの作業は不要です。

```
server.http-service.virtual-server.server.http-listeners = ajp-listener-1
```

連携設定を行うドメインが複数存在する場合は、ドメインごとに「WebOTX ドメインへの外部 Web サーバ連携の設定」作業を行ってください。

運用管理コマンドの起動

最初に、運用管理コマンドを起動し、設定を変更するドメインが起動していることを確認します。

運用管理コマンドを起動します。

```
root> /opt/WebOTX/bin/otxadmin
```

設定を変更するドメインが起動していることを確認します。

```
otxadmin> list-domains
```

ドメインの設定変更

Java ベースの内蔵 Web サーバを利用する設定から、外部 Web サーバを利用する設定に変更します。

管理ユーザとしてログインします。

```
otxadmin> login --user <管理ユーザ名> --password <管理ユーザパスワード>
--port <管理ポート番号>
```

※ 実際には1行で実行してください。

外部 Web サーバを利用する設定に変更します。

```
otxadmin> set server.http-service.virtual-server.server.http-listeners=
"ajp-listener-1"
```

※ 実際には1行で実行してください。

ドメインの停止

ドメインを停止し、ログアウトします。

```
otxadmin> stop-domain <ドメイン名>
otxadmin> logout
otxadmin> exit
```

3.2.3. Web コンテナとは別マシン上の Web サーバを使う場合

Web コンテナが動作するマシンと、Web サーバが動作するマシンが別マシンである場合、Web サーバが読み込む設定ファイルの変更が必要になります。下記手順に従い、変更してください。

マルチプロセスモード時

コンテナの情報は\${INSTANCE_ROOT}/config/httpgateway.ior から読み込まれます。下記例を参考に設定を変更してください。

- ① Web サーバが動作するマシンに\${INSTANCE_ROOT}/config/iors ディレクトリを作成します。
- ② ①で作成したディレクトリに、Web コンテナが動作するマシン上にある httpgateway.ior をコピーします。httpgateway.ior は Web コンテナ起動時に自動的に作成されます。
- ③ Web サーバが動作するマシンの\${INSTANCE_ROOT}/config/WebCont/ior_workers.properties を開き、下記例を参考に②でコピーした httpgateway.ior を参照するように ior_def 定義を修正します。(赤字が修正箇所)

```
# TM_PLUGIN_IOR-start
worker.list=otxiop
otx_domain=/opt/WebOTX/domains/domain1
worker.otxiop.type=otxiop
worker.otxiop.default_ior=localdomain
worker.otxiop.ior_def.localdomain=$(otx_domain)/config/iors/httpgateway.ior
worker.otxiop.cachesize=150
# V8 では cachesize の代わりに、connection_pool_size を使用してください。
# TM_PLUGIN_IOR-end
```

シングルプロセスモード時

`${INSTANCE_ROOT}/config/WebCont/workers.properties` の下記項目を必要に応じて変更してください。

```
worker.ajp13.port=<連携ポート番号>
worker.ajp13.host=<連携するドメインのホストのアドレス>
```

また、Web コンテナと Web サーバが別マシンである構成で動的反映をオフにした場合、新規配備後は設定ファイルの手動更新が必要になります。

動的反映・設定ファイルの更新については「5.1 動的反映」を参照してください。

3.3. 設定の確認

ここでは、上記連携設定によって設定ファイルが正常に更新されたことを確認します。

通常はこの作業は必要ありません。トラブルシューティングなどの際に参考にしてください。

コンテナの動作確認

V6.x, V7.x の場合

`http://<コンテナのホスト>:<ポート番号>/admin/` にアクセスし、ログインできることを確認します。

※ポート番号の既定値は「4848」です。

V8.x の場合

`http://<コンテナのホスト>:<ポート番号>/manager/` にアクセスし、ログインできることを確認します。

※ポート番号の既定値は「5858」です。

適当な Web アプリケーションを配備します。

連携設定の確認

`${INSTANCE_ROOT}/config/WebServer/httpd.conf` を開き、下記のようなコメントに囲まれた記述があることを確認します。

```
# TM_WS_PLUGIN-start
include "C:/WebOTX/domains/domain1/config/WebCont/mod_jk_om-20.conf-auto"
# TM_WS_PLUGIN-end
```

`include` の部分で指定するファイル名は、使用する Web サーバの種類等により変わります。下表を参照してください。

使用する Web サーバ	プロセスモード	ファイル名
WebOTX Web サーバ 1.3	シングル	\${INSTANCE_ROOT}/config/WebCont/mod_jk.conf-auto
	マルチ	\${INSTANCE_ROOT}/config/WebCont/mod_jk_om.conf-auto
WebOTX Web サーバ 2.0	シングル	\${INSTANCE_ROOT}/config/WebCont/mod_jk-20.conf-auto
	シングル (WebOTX V6.1 の場合)	\${INSTANCE_ROOT}/config/WebCont/mod_jk2.conf-auto
	マルチ	\${INSTANCE_ROOT}/config/WebCont/mod_jk_om-20.conf- auto

設定ファイルの確認

・マルチプロセスモードの場合

下記例を参考に\${INSTANCE_ROOT}/config/WebCont/ior_workers.properties が正しく設定されていることを確認します。

```
# TM_PLUGIN_IOR-start
worker.list=otxiiop
otx_domain=C:/WebOTX/domains/domain1
worker.otxiiop.type=otxiiop
worker.otxiiop.default_ior=localdomain
worker.otxiiop.ior_def.localdomain=${otx_domain}/config/iors/httpgateway.ior
worker.otxiiop.cachesize=150
# V8 では cachesize から conection_pool_size に変更されました。
# TM_PLUGIN_IOR-end
```

・シングルプロセスモードの場合

下記例を参考に\${INSTANCE_ROOT}/config/WebCont/workers.properties が正しく設定されていることを確認します。

```
worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
worker.ajp13.cachesize=150
# V8 では cachesize から connection_pool_size に変更されました。
```

連携動作の確認

http://<Web サーバのホスト>:<Web サーバのポート番号>/<Web アプリケーションのコンテキスト名>/にアクセスし、正常に動作することを確認します。

ログレベルの変更

プラグインのログレベルを「debug」にしてログを採取することにより、問題解決の手助けとなることがあります。ログレベルには、「trace」、「debug」、「info」、「warn」、「error」、「emerg」が指定可能です。

ログレベルを変更する場合は、上記「連携設定の確認」項目で指定した設定ファイル(xxx.conf-auto)を下記のように手動で更新してください。(赤字が更新箇所)

```
JkLogLevel debug
```

更新後は Web サーバを再起動してください。

※WebOTX V6.1 で WebOTX Web サーバ 2.0 を使用する場合は、\${INSTANCE_ROOT}/config/WebCont/workers2.properties を下記のように変更してください。ログレベルには、「DEBUG」、「INFO」、「ERROR」、「EMERG」が指定可能です。(赤字が変更箇所)

```
[logger.apache2]  
level=DEBUG
```

4. IIS との連携設定

IIS との連携設定について説明します。

※IIS7 と連携する場合、Web サーバー(IIS)の役割サービス「ISAPI 拡張」および「ISAPI フィルタ」がインストールされている必要があります。また、環境設定ツールを使用して連携設定するためには、役割サービス「IIS6 メタベース互換」がインストールされている必要があります。インストールされていない場合、下記の手順でインストールを行ってください。

- ①「スタートメニュー」-「全てのプログラム」-「管理ツール」から「サーバーマネージャ」を起動します。
- ②左画面で「サーバーマネージャ」-「役割」-「Web サーバー(IIS)」を選択します。
- ③右画面で「概要」-「役割サービスの」の「役割サービスの追加」から上記の役割サービスをインストールしてください。

4.1. WebOTX ドメインへの外部 Web サーバ連携の設定

インストール時に WebOTX Web サーバをインストールしなかった場合、デフォルトでは内蔵 Web サーバを使用する設定になっています。

ここでは、WebOTX 内蔵 Web サーバを使用する設定から、内蔵 Web サーバを無効にして外部 Web サーバを使用する設定に変更する方法を説明します。外部 Web サーバを使用する設定になっているか、WebOTX 内蔵 Web サーバを使用する設定になっているかを確認するには、運用管理コマンドを起動し下記のコマンドを実行してください。

```
otxadmin> get server.http-service.virtual-server.server.http-listeners
```

上記コマンドの実行結果が下記のようにになっていた場合、内蔵 Web サーバを使用する設定になっているため、下記手順に従い設定を行ってください。

```
server.http-service.virtual-server.server.http-listeners = http-listener-1,http-listener-2,ajp-listener-1
```

実行結果が下記のようにになっていた場合、既に外部 Web サーバを使用する設定になっているため、ドメインを停止し「4.2 外部 Web サーバへの WebOTX 連携情報の設定前の作業」に進んでください。

```
server.http-service.virtual-server.server.http-listeners =ajp-listener-1
```

連携設定を行うドメインが複数存在する場合は、ドメインごとに「WebOTX ドメインへの外部 Web サーバ連携の設定」作業を行ってください。

運用管理コマンドの起動

最初に、[スタート] - [プログラム] - [WebOTX] - [運用管理コマンド]を起動し、設定を変更するドメインが起動していることを確認します。

```
otxadmin> list-domains
```

ドメインの設定変更

Java ベースの内蔵 Web サーバを利用する設定から、外部 Web サーバを利用する設定に変更します。

管理ユーザとしてログインします。

```
otxadmin> login --user <管理ユーザ名> --password <管理ユーザパスワード>
--port <管理ポート番号>
```

※ 実際には1行で実行してください。

外部 Web サーバを利用する設定に変更します。

```
otxadmin> set server.http-service.virtual-server.server.http-listeners=
"ajp-listener-1"
```

※ 実際には1行で実行してください。

ドメインの停止

ドメインを停止し、ログアウトします。

```
otxadmin> stop-domain <ドメイン名>

otxadmin> logout

otxadmin> exit
```

この作業が終われば、「4.2 外部 Web サーバへの WebOTX 連携情報の設定前の作業」に進んでください

4.2. 外部 Web サーバへの WebOTX 連携情報の設定前の作業

環境設定ツールを使用して環境設定を行うと、`$(INSTANCE_ROOT)\config\WebCont` 配下にある `workers.properties` が初期化されます。これらのファイルを編集している場合はバックアップを取っておいてください。

次の作業に先立って、IIS を停止してください。

※ IIS の複数サイトに対して、複数のドメインと連携するように設定を行った場合、ただしく連携することができません。

Web コンテナと IIS が同一マシン上で動作する場合、ここでの作業はありません。「4.3 外部 Web サーバへの WebOTX 連携情報の設定」に進んでください。

Web コンテナとは別マシン上の IIS を使用する場合

Web コンテナ動作マシンと IIS 動作マシンの OS が異なる場合、IIS 用のプラグインを入手するために、IIS 動作マシンに WebOTX のインストールが必要です。IIS 動作マシンに WebOTX のインストールを行ってください。また、IIS 動作マシンに WebOTX をインストールした場合ここで必要な作業はありません。「4.3 外部 Web サーバへの WebOTX 連携情報の設定」に進んでください。

Web コンテナ動作マシンと IIS 動作マシンの OS が同じ場合、下記「IIS 動作マシンに WebOTX をインストールしない場合」の手順で必要なファイルを Web コンテナ動作マシンからコピーし、手動でレジストリに設定を行うことにより、Web コンテナとの連携動作は可能となりますが、この場合外部 Web サーバとの連携についてはサポートの対象外となります。完全なサポートを受けるためにも、IIS の動作するマシンに WebOTX Web Edition をインストールすることを推奨します。詳細は「6. ライセンスについて」を参照してください。

IIS 動作マシンに WebOTX をインストールした場合、ここでの作業はありません。「4.3 外部 Web サーバへの WebOTX 連携情報の設定」に進んでください。

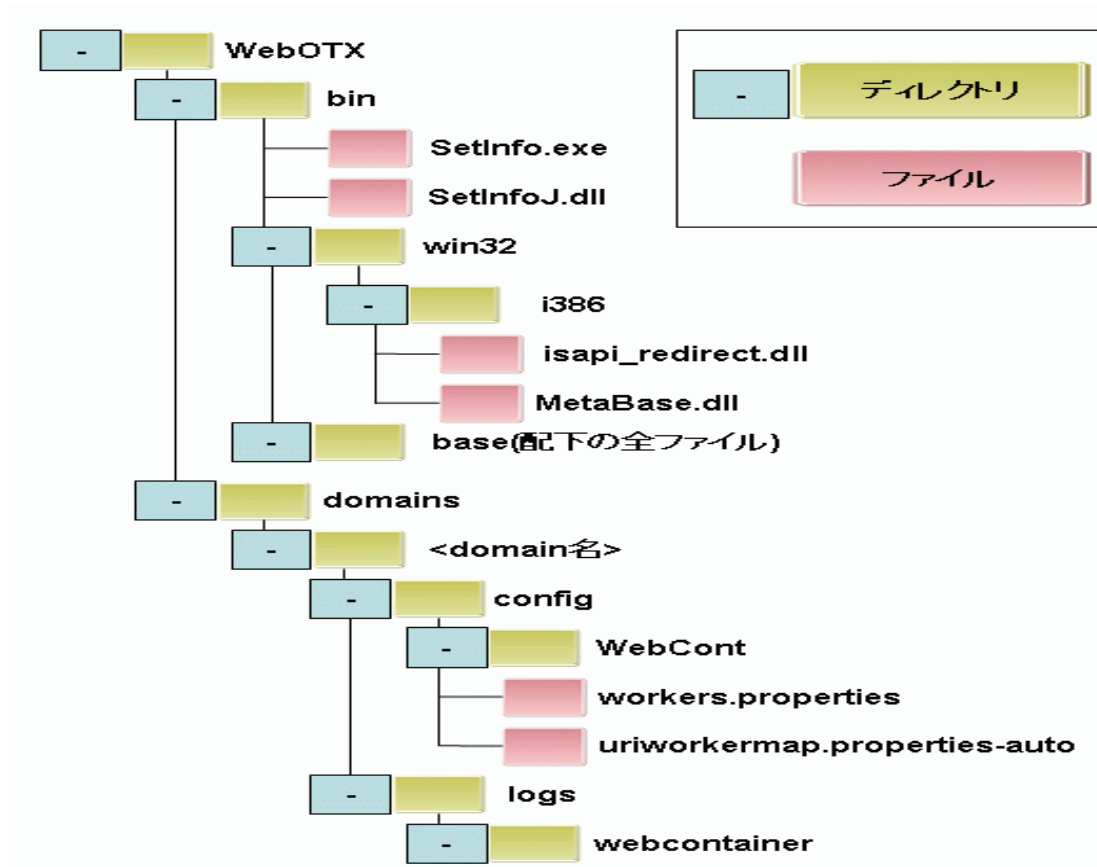
・IIS 動作マシンに WebOTX をインストールしない場合

手動でレジストリの操作およびファイルのコピーを行い、環境設定ツールをインストールします。

※レジストリの編集を誤ると深刻な問題が発生することがあります。最悪の場合、オペレーティングシステムの再インストールが必要です。レジストリの編集はバックアップをとった上、各自の責任で注意して行ってください。

①ファイルのコピー

IIS の動作するマシンに WebOTX をインストールしない場合は、下図のような構成を保って Web コンテナ動作マシン上のファイルを IIS 動作マシンにコピーしてください。



②レジストリ設定用ファイル(is_redirect.reg)の作成

iisredirect.regを新規に作成し、テキストエディタ等で下記の内容を記述します。”C:¥¥WebOTX¥¥”の箇所は①で作成したディレクトリを指定してください。

WebOTX V6.x, V7.x を使用している場合

```
REGEDIT4
[HKEY_LOCAL_MACHINE¥SOFTWARE¥NEC¥WebOTX WebContainerT50]
"PathName"="C:¥¥WebOTX¥¥"
```

WebOTX V8 を使用している場合

```
REGEDIT4
[HKEY_LOCAL_MACHINE¥SOFTWARE¥NEC¥WebOTX WebContainer]
"PathName"="C:¥¥WebOTX¥¥"
```

③レジストリの設定

②で作成した iis_redirect.reg をダブルクリックして実行し、レジストリを設定します。ダブルクリック後、レジストリエディタの確認画面が表示されるので、「はい」を押して設定を行ってください。

4.3. 外部 Web サーバへの WebOTX 連携情報の設定

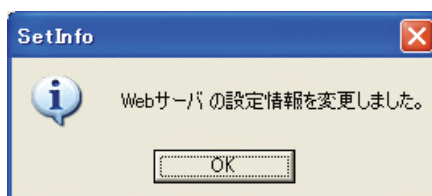
この作業の前に連携設定を行う外部 Web サーバが停止していることを確認してください。

[スタート] - [プログラム] - [WebOTX] - [環境設定ツール]メニューを選択し、環境設定ツールを起動します。起動された環境設定ツール画面に情報を入力して、Web コンテナと Web サーバの連携設定を行っていきます。

設定項目		説明
Web コンテナ の情報	ドメインディレクトリ	Web サーバと連携する Web コンテナのドメインを指定します。 ドメインの指定例を次に示します。 C:\WebOTX\domains\domain1
Web サーバの 情報	種別	Web コンテナと連携する Web サーバの種別を選択します。 「Internet Information Services(IIS)」を選択してください。
	連携ポート番号	Web コンテナと Web サーバとの連携に使用するポート番号を指定します。 domain1 の既定値は「8009」です。 ※ v8 での既定値は「8099」です。

Web アプリケーションのコンテキスト名動的反映オプション	<p>Web アプリケーションのコンテキスト名動的反映オプションを選択します。</p> <p>「動的反映あり(従来どおり)」を選択した場合、動的反映が常に有効になります。</p> <p>「動的反映1回のみ(Web サーバ起動時に1回のみ動的反映を行う)」を選択した場合、Web サーバ起動時に1回のみ動的反映が行われます。動的反映の詳細は「5.1 動的反映」を参照してください。</p> <p>※このオプションは WebOTX V6.5 以前のバージョンでは利用できません。</p>
-------------------------------	---

入力が完了すれば「OK」ボタンを押下してください。「キャンセル」ボタンを押下すると設定情報を更新せずにツールは終了します。



上記ダイアログが表示されれば、Web コンテナの各種定義ファイル、および Web サーバの設定ファイルに設定情報が反映されます。

4.4. Web コンテナとは別マシン上の IIS を使う場合

Web コンテナが動作するマシンと Web サーバが動作するマシンが別マシンである場合、Web サーバマシン上の `$(INSTANCE_ROOT)\%config%\WebCont` 配下、`workers.properties` ファイルの変更が必要になります。このファイルは環境設定ツールの実行により初期化されるため、環境設定ツール実行後に下記設定変更を行ってください。

下記項目を必要に応じて変更してください。

```
worker.ajp13.port=<連携ポート番号>
worker.ajp13.host=<連携するドメインのホストのアドレス>
```

また、この構成で動的反映をオフにした場合、新規配備後は設定ファイルの手動更新が必要になります。

動的反映・設定ファイルの更新については「5.1 動的反映」を参照してください。

4.5. IIS の設定

4.5.1. 基本認証の設定

IIS+Web コンテナの環境で、IIS の基本認証の設定が出来ていないと、認証を行う Web アプリケーションにログインできません。以下の作業を行ってください。

基本認証の有効化

IIS マネージャ (IIS5.0 ではインターネットサービスマネージャ) を起動し、連携設定対象の Web サイトのプロパティを開きます。「ディレクトリ セキュリティ」タブから「認証とアクセス制御」(IIS5.0 では「匿名アクセスおよび認証コントロール」) の「編集」ボタンを押下し、「認証方法」画面を表示します。「認証方法」画面で「統合 Windows 認証」を無効にし、「基本認証」を有効にします。

※「匿名アクセスを有効にする」のチェックをはずしている場合、IIS へのアクセスの際に認証が必要になるため、一般の Web アプリケーションへのアクセス時にも認証が必要となります。認証画面では Windows システムに存在するいずれかのユーザでログインしてください。

認証ユーザの登録

IIS で基本認証を行う場合、Windows システムに認証を行うユーザが登録されている必要があります。以下のいずれかの方法で対応してください。

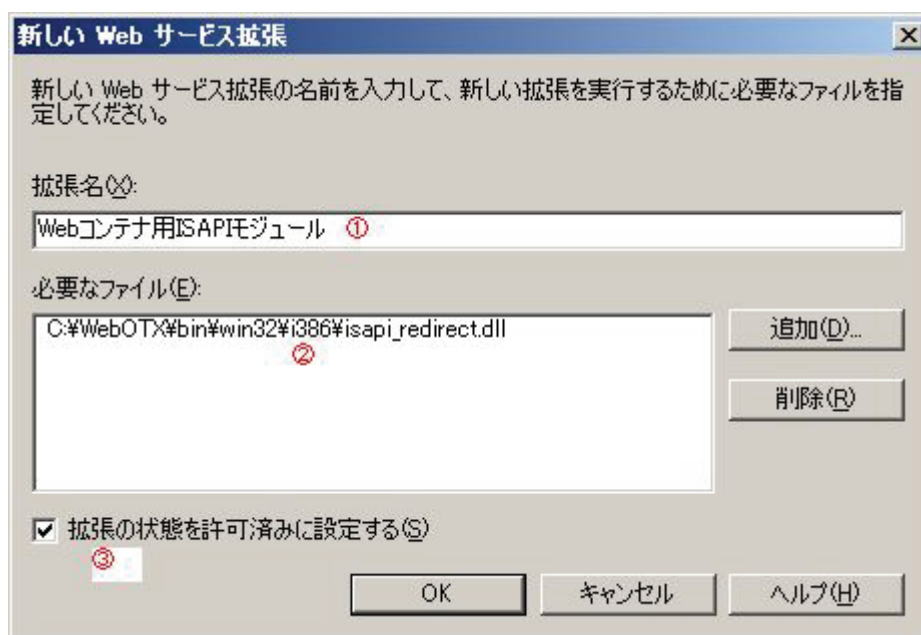
- Windows システムに Web コンテナの認証に使うユーザを登録する
- ドメインの管理ユーザに Windows システムのログインユーザを設定する。

4.5.2. Web サービス拡張の設定

IIS6.0、IIS7.0 と連携する場合、下記の設定が必要です。

IIS6.0 の場合

IIS マネージャで左のツリーから「Web サービス拡張」を選択します。右画面に Web サービスの拡張状態が表示されますので、タスクとして「新しい Web サービス拡張を追加…」を押下し、「新しい Web サービス拡張」画面を開きます。

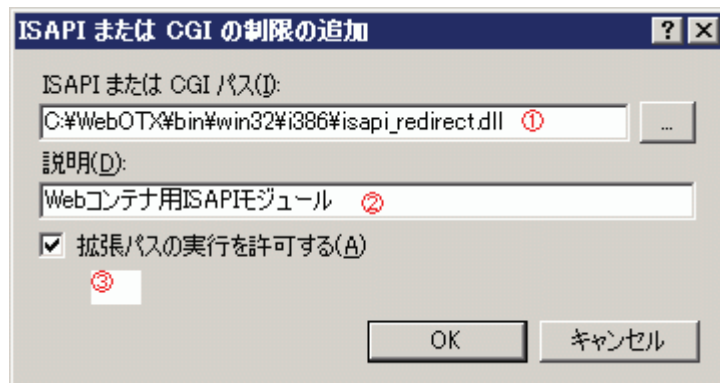


開いた「新しい Web サービス拡張」画面で、「拡張名」に「Web コンテナ用 ISAPI モジュール」(①)を指定し、追加ボタンを押下して「必要なファイル」に「%INSTALL%\bin\win32\i386\isapi_redirect.dll」を指定(②)します。

そして「拡張の状態を許可済みに設定する」をチェック(③)し、「OK」を押下してください。

IIS7.0 の場合

IIS マネージャを開き、サーバーの階層に移動し、「機能」ビューの「ISAPI および CGI の制限」をダブルクリックします。操作ウィンドウの「追加」をクリックし、「ISAPI または CGI の制限の追加」ダイアログを開きます。



開いたダイアログの「ISAPI または CGI パス」に「%INSTALL%\bin\win32\i386\isapi_redirect.dll」(①)を指定し、「説明」に「Web コンテナ用 ISAPI モジュール」(②)を指定します。

「拡張パスの実行を許可する」をチェック(③)し、「OK」を押下してください。

4.6. 設定の確認

ここでは、上記連携設定によって設定ファイルが正常に更新されたことを確認します。

通常はこの作業は必要ありません。トラブルシューティングなどの際に参考にしてください。

コンテナの動作確認

v6.x,v7.x の場合

http://<コンテナのホスト>:<ポート番号>/admin/ にアクセスし、ログインできることを確認します。

※ポート番号の既定値は「4848」です。

v8 の場合

http://<コンテナのホスト>:<ポート番号>/manager/ にアクセスし、ログインできることを確認します。

※ポート番号の既定値は「5858」です。

適当な Web アプリケーションを配備します。

連携設定の確認

レジストリエディタ等を使用し、レジストリキー HKEY_LOCAL_MACHINE\SOFTWARE\NEC\WebOTX\WebContainerT50\Isapi Redirector\1.0 配下に以下の項目があることを確認します。

※V8 では HKEY_LOCAL_MACHINE\SOFTWARE\NEC\WebOTX\WebContainer\Isapi Redirector\1.0 配下

名前	データ例
extension_uri	/webcont/isapi_redirect.dll
log_file	\${INSTANCE_ROOT}/logs/webcontainer/iis_redirect.log
log_level	error
worker_file	\${INSTANCE_ROOT}/config/WebCont/workers.properties
worker_mount_file	\${INSTANCE_ROOT}/config/WebCont/uriworkermap.properties-auto

設定ファイルの確認

下記例を参考に、上記 worker_file 項目で設定した workers.properties ファイルが正しく設定されていることを確認します。

```
worker.list=ajp13
worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
worker.ajp13.cachesize=150
```

連携動作の確認

http://<Web サーバのホスト>:<Web サーバのポート番号>/<Web アプリケーションのコンテキスト名>/にアクセスし、正常に動作することを確認します。

ログレベルの変更

プラグインのログレベルを「debug」にしてログを採取することにより、問題解決の手助けとなることがあります。ログレベルには、「trace」、「debug」、「info」、「warn」、「error」、「emerg」が指定可能です。下記手順を行うことによりログレベルを変更できます。

IIS を停止してください。

次に、レジストリエディタを起動し、上記「連携設定の確認」項目で確認した log_level のデータを下記のように手動で更新してください。(赤字が追加箇所)

```
"log_level"="debug"
```

更新後は Web サーバを再起動してください。

指定しているディレクトリにログが出力されない場合

IIS 連携時に、プラグインのログが指定しているディレクトリに出力されない場合があります。この場合、次の処理を行ってください。

- ① \${INSTANCE_ROOT}\logs の「プロパティ」を開き、「セキュリティ」タブで「追加」ボタンを押下します。
- ② 「ユーザーまたはグループの選択」ダイアログで自マシンの NETWORK SERVICE を選択し、「OK」を押下します。
- ③ 「詳細設定」ボタンを押下します。
- ④ 「セキュリティの詳細設定」ダイアログの「アクセス許可」タブで「追加」を押下します。

- ⑤「ユーザーまたはグループの選択」ダイアログで自マシンの NETWORK SERVICE を選択し、「OK」を押下します。
- ⑥「アクセス許可エントリ」ダイアログの「適用先」で「このフォルダとサブフォルダ」を選択し、「アクセス許可」で「ファイルの作成／データの書き込み」と「フォルダの作成／データの追加」をチェックして「OK」を押下します。
- ⑦「セキュリティの詳細設定」ダイアログ、「プロパティ」を「OK」を押下して閉じます。

4.7. 連携設定の解除

WebOTXをアンインストールした場合でも、IISなどの外部 Web サーバには連携設定の内容が残っているため、連携設定を解除せずそのまま Web サーバを使い続けた場合、システムによっては Web サーバが正常に起動しなくなる可能性があります。

ここでは IIS での連携設定の解除の手順について説明します。

IIS マネージャ(IIS5.0 ではインターネットサービスマネージャ)を起動します。

仮想ディレクトリの削除

連携していた「Web サイト」を展開し、仮想ディレクトリ「webcont」を削除します。

ISAPI フィルタの削除

連携していた Web サイトのプロパティを開き、「ISAPI フィルタ」タブから「webcont」を削除します。

認証設定の変更

IIS の設定時に変更した基本認証の設定を必要に応じて変更してください。また、Web コンテナの認証ユーザを Windows システムに登録した場合、不要ならば Windows システムのユーザを削除してください。

IIS6.0 の場合

IIS6.0 を使用している場合、下記の設定変更も必要です。

IIS マネージャで左のツリーから「Web サービス拡張」を選択します。右画面に Web サービスの拡張状態が表示されますので、「Web コンテナ用 ISAPI モジュール」を指定し、右クリックメニューから削除を選び削除します。

IIS7.0 の場合

IIS7.0 を使用している場合、下記の設定変更も必要です。

IIS マネージャでサーバーの階層で機能ビューの「ISAPI および CGI の制限」をダブルクリックします。「Web コンテナ用 ISAPI モジュール」を選択し、右クリックメニューから削除を選び削除します。

5. 詳細設定

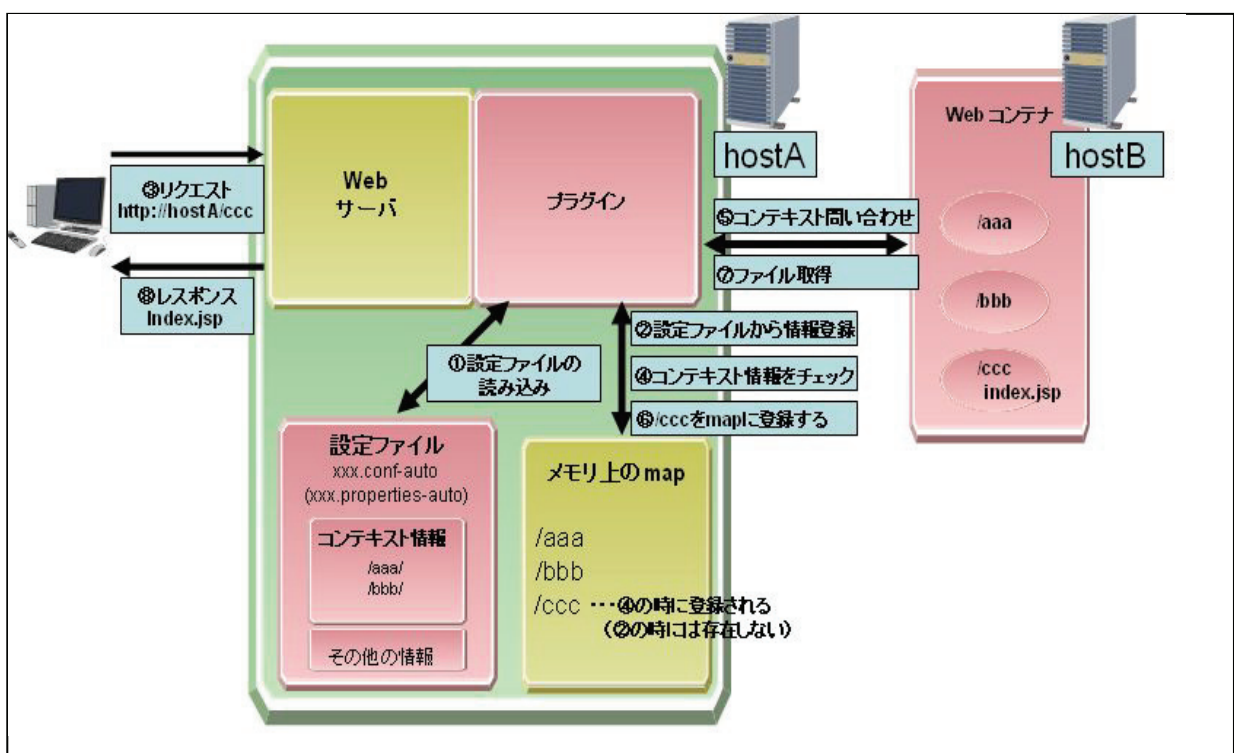
動的反映、ロードバランス設定、エラードキュメントのカスタマイズなど、詳細な設定項目について説明します。

5.1. 動的反映

Web コンテナ上のコンテキストの情報は、設定ファイルに記述され、プラグインから参照されます。

動的反映とは、新規に配備された Web アプリケーションのコンテキストなど設定ファイルに記述されていないコンテキストに対するリクエストがきた場合に、Web コンテナに対してコンテキスト問い合わせの query を送信することにより、新規コンテキストを外部 Web サーバに動的に反映する機能です。

下図は動的反映のイメージ図です。



- ① Web サーバの起動時に設定ファイルからコンテキストの情報を読み込む(/aaa、/bbb の情報が読み込まれる)
- ② 設定ファイルから読み込んだコンテキスト情報をメモリ上の map に登録する (map に /aaa、/bbb が登録される)
ここまでは Web サーバ起動時に行われる
- ③ ブラウザから新規配備された Web アプリケーションへのリクエストがくる (<http://hostA/ccc/>)
- ④ メモリ上の map を参照し、コンテキストが登録されているかを確認する
(/ccc は新規に配備されたので、登録されていない)
- ⑤ Web コンテナに対して (hostB:8009)、コンテキスト問い合わせの query を送信し、配備されているかどうかを確認する
- ⑥ コンテキスト名をメモリ上の map に登録する (/ccc が登録される)
- ⑦ リクエストされたファイルを取得する (index.jsp を取得)
- ⑧ レスポンスを返す (index.jsp を返す)

この機能はデフォルトでは有効になっており、この機能によってドメインや Web サーバ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。

5.1.1. 動的反映オフの設定

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となります。そこで、この動的反映をしない設定を行うことによって、負荷を低減することが出来ます。

この設定を行った場合、Web サーバは Web コンテナに対してコンテキスト名の問い合わせを行いません。そのため、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、プラグインが参照する設定ファイルを更新し、外部 Web サーバを再起動する必要があります。

※コンテキスト情報が記述された設定ファイルは、シングルプロセスモードの場合、ドメインの再起動で自動的に更新されます。ただし、Web コンテナが動作するマシンと、Web サーバが動作するマシンが異なる場合は、Web サーバ側の設定ファイルは更新されないため、手動で更新する必要があります。

動的反映オフの設定は次の手順で行います。

外部 Web サーバの停止

設定ファイルを変更するため、外部 Web サーバを停止してください。

workers.properties の設定

``${INSTANCE_ROOT}/config/WebCont/workers.properties`` に次の名称でワーカ名を `worker.list` プロパティの最後に追加してください。

```
ajp13_original
```

workers.properties の例 (赤字が追加箇所)

```
#----- DEFAULT woret 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
```

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。

5.1.2. 動的反映オプション query once の設定

動的反映オプションを「query once」にした場合、Web サーバは起動時に1度だけコンテナに対してコンテキスト問い合わせの query を送信します。Web サーバ起動後にアプリケーションを新規配備した場合は、Web サーバの再起動により、コンテキスト名を Web サーバに反映することができます。

「query once」を使用すれば、コンテナへの余計な通信を減らすことができ、Web サーバの再起動により設定ファイルの手動での更新も不要になります。

query once の設定は環境設定ツールの「Web アプリケーションのコンテキスト名動的反映オプション」で行います。詳細は各 Web サーバの連携設定の手順を参照してください。

5.1.3. 設定ファイルの更新

WebOTX の構成によっては、手動での設定ファイルの更新が必須になることがあります。下記表で「手動でのファイルの更新」が「要」になっている構成・運用方法を取り、新規コンテキストに配備を行った場合、コンテキスト情報を設定ファイルに手動で記述してください。手動でのファイル更新後は Web サーバの再起動が必要です。

動的反映	マシン構成	プロセスモード	手動でのファイル更新
ON	-	-	不要
query once	-	-	不要(※1)
OFF	コンテナと Web サーバが 同一マシン上にある	シングル	不要(※2)(※3)
		マルチ	要
	コンテナと Web サーバが 別マシン上にある	-	要

(※1) 配備後、Web サーバを再起動してください。

(※2) 配備後、ドメインを再起動してください。再起動しない場合は、手動でのファイル更新が必要になります。

(※3) Apache HTTP Server1.3.x を使用する場合は、手動でのファイル更新が必要です。

コンテキスト名を記述する設定ファイルは使用する Web サーバの種類によって変わります。

コンテキスト情報を手動で記述する必要がある場合は、下記 Web サーバごとの記述例と、Web コンテナ側の設定ファイルを参考に追加してください。

Web Server の種類	設定ファイル
WebOTX Web サーバ 1.3(シングルプロセスモード)	mod_jk.conf-auto
WebOTX Web サーバ 1.3(マルチプロセスモード)	mod_jk_om.conf-auto
WebOTX Web サーバ 2.0(シングルプロセスモード)	mod_jk-20.conf-auto
WebOTX Web サーバ 2.0(マルチプロセスモード)	mod_jk_om-20.conf-auto
IIS	uriworkermap.properties-auto
Sun Java Web Server (Sun ONE) 6.x	uriworkermapNS.properties-auto

WebOTX Web サーバ 1.3(シングルプロセスモード) / WebOTX Web サーバ 2.0(シングルプロセスモード)の記述例

```
Alias "/context 名" "C:/WebOTX/domains/domain1/applications/j2ee-modules/app
名"

JkMount "/context 名/*" ajp13
JkMount "/context 名" ajp13
```

WebOTX Web サーバ 1.3(マルチプロセスモード) / WebOTX Web サーバ 2.0(マルチプロセスモード)の記述例

```
Alias "/context 名" "C:/WebOTX/domains/domain1/applications/j2ee-modules/app
名"

JkMount "/context 名/*" otxiop
JkMount "/context 名" otxiop
```

IIS / Sun Java Web Server の記述例

```
/context 名/*=$(default.worker)
/context 名=$(default.worker)
```

5.2. ロードバランス設定

ここではプラグインのロードバランス機能を使用するための手順を説明します。

プラグインでのロードバランス機能の振り分け方法は単純なラウンドロビンのみです。新規コネクション生成を単純なラウンドロビン方式で行い、コネクション生成に失敗した場合は、別のドメインへ接続をリトライします。ドメインを閉塞したり、復旧したりする機能はありません。

5.2.1. IIOP プラグインでのロードバランス設定

IIOP プラグインのロードバランス機能を使用するための手順を説明します。

下記の例を参考にしてロードバランス設定を行ってください。

以下は、hostA の domainA と hostB の domainB に負荷を分散する場合の手順例です。

- ① hostA の domainA と hostB の domainB にテスト用アプリケーションを配備する。
- ② 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 domainA のプロセスグループの JavaVM オプション

```
"-DjvmRoute=otxiiop1"
```

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

```
"-DjvmRoute=otxiiop2"
```

- ③ domainA と domainB を再起動する。
- ④ $\${INSTANCE_ROOT}/\text{config}/\text{iors}$ ディレクトリを作成する。
- ⑤ $\${INSTANCE_ROOT}/\text{config}/\text{iors}/\text{hostA_domainA}$ ディレクトリを作成する。
- ⑥ $\${INSTANCE_ROOT}/\text{config}/\text{iors}/\text{hostB_domainB}$ ディレクトリを作成する。
- ⑦ hostA の $\${INSTANCE_ROOT}/\text{config}/\text{httpgateway.ior}$ を⑤にコピーする。
- ⑧ hostB の $\${INSTANCE_ROOT}/\text{config}/\text{httpgateway.ior}$ を⑥にコピーする。
- ⑨ $\${INSTANCE_ROOT}/\text{config}/\text{WebCont}/\text{ior_workers.properties}$ を以下の例を参考に修正する。(赤字が修正箇所)

```
# TM_PLUGIN_IOR-start
#worker.list=otxiiop
otx_domain=C:/WebOTX/domains/domain1
#デフォルトの設定をコメントアウト
#worker.otxiiop.type=otxiiop
#worker.otxiiop.default_ior=localdomain
#worker.otxiiop.ior_def.localdomain=$(otx_domain)/config/httpgateway.ior
# TM_PLUGIN_IOR-end

#ワーカーリストに domainA 用のワーカー otxiiop1 と domainB 用のワーカー otxiiop2 を追加
```

```

#ロードバランス用ワーカー otxiop は末尾に記述
worker.list=otxiop1,otxiop2,otxiop

#domainA 用のワーカー otxiop1 を設定
worker.otxiop1.type=otxiop
worker.otxiop1.default_ior=hostA_domainA
worker.otxiop1.ior_def.hostA_domainA=$(otx_domain)/config/iors/hostA_domainA/httpgateway.ior

#domainB 用のワーカー otxiop2 を設定
worker.otxiop2.type=otxiop
worker.otxiop2.default_ior=hostB_domainB
worker.otxiop2.ior_def.hostB_domainB=$(otx_domain)/config/iors/hostB_domainB/httpgateway.ior

#ロードバランスを行うワーカー otxiop を設定
worker.otxiop.type=lb
worker.otxiop.balanced_workers=otxiop1,otxiop2

```

⑩ Web サーバを再起動する

以上でロードバランスの設定は完了です。

5.2.2. Web サーバプラグインでのロードバランス設定

Web サーバプラグイン(JK コネクタ)のロードバランス機能を使用するための手順を説明します。Web サーバプラグインはシングルプロセスモード時に使用されます。IIS やシングルプロセスモード時に WebOTX Web サーバを使用する場合はこの項目を参照してください。

下記の例を参考にしてロードバランス設定を行ってください。

以下は、hostA の domainA と hostB の domainB に負荷を分散する場合の手順例です。

- ① hostA の domainA と hostB の domainB にテスト用アプリケーションを配備する。
- ② `${INSTANCE_ROOT}/config/WebCont/workers.properties` を以下の例を参考に修正する。(赤字が修正箇所)

```

#
#----- DEFAULT worket list -----
#-----
ps=/
#
# The workers that your plugins should create and work with
#

```

```
#ワーカリストに domainA 用のワーカ ajp13-1 と domainB 用のワーカ ajp13-2 を追加
#ロードバランス用ワーカ ajp13 は末尾に記述

worker.list=ajp13-1,ajp13-2,ajp13

#
#----- 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
#
#domainA 用のワーカ ajp13-1 を設定
worker.ajp13-1.port=8009
worker.ajp13-1.host=<hostA>
worker.ajp13-1.type=ajp13
worker.ajp13-1.cachesize=150

#domainB 用のワーカ ajp13-2 を設定
worker.ajp13-2.port=8009
worker.ajp13-2.host=<hostB>
worker.ajp13-2.type=ajp13
worker.ajp13-2.cachesize=150

#ロードバランスを行うワーカ ajp13 を設定
worker.ajp13.type=lb
worker.ajp13.balanced_workers=ajp13-1,ajp13-2
```

③ Web サーバを再起動する

以上でロードバランスの設定は完了です。

5.3. ロードバランス利用時のセッションの維持

WebOTXではセッションレプリケーションの機能が利用できます。レプリケーションの機能を利用することにより、複数のWebコンテナでHTTPセッションを共有することができます。さらにJNDIサーバを多重化すれば、一方のJNDIサーバがダウンした場合でも、もう一方のJNDIサーバからセッション情報を取得することができます。

WebOTX V6.xでは、JNDIサーバの多重化を行うとHTTPセッションだけでなく、JDBCデータソース、JMSリソースも同じ内容となってしまいますが、V7.1では、HTTPセッションの情報のみを多重化することが可能です。HTTPセッションの情報のみの多重化を行えば、ドメインごとに異なる設定内容を使用することができます。

ここではセッションレプリケーションの設定方法について説明します。

5.3.1. セッションレプリケーションの設定

Webアプリケーションのweb.xmlに<distributable/>を記述することにより、セッション情報がJNDIサーバにレプリケートされるようになります。<distributable/>を追加するWebアプリケーションがHttpSessionオブジェクトに格納するオブジェクトはSerializableである必要があります。

下記、記述例を参考に設定を行ってください。

web.xml の記述例

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!DOCTYPE web-app
  PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//EN"
  http://java.sun.com/dtd/web-app_2_3.dtd>
<web-app>
  <distributable/>
  <context-param>
    <param-name>paramName1</param-name>
    <param-value>paramValue1</param-value>
  </context-param>
  ...
  ...
</web-app>
```

5.3.2. JNDI サーバの多重化

HTTPセッションの情報のみの多重化:

WebOTX V7以降のWebOTXではHTTPセッションの情報だけを多重化することができます。

運用管理コマンドから下記のように設定してください。

```
otxadmin> set server.web-container.session-replication-jndi-url=rmiioop://<host1>:<port1>,<host2>:<port2>,...
```

※ここで指定するポートは WebOTX ドメインの組み込み IIOP リスナのポート番号で、WebOTX インストール時に作成される domain1 のポート番号の既定値は 7780 です。

V6.x での多重化:

この方法では HTTP セッションの情報だけを多重化することはできません。多重化の設定を行った場合、HTTP セッションの情報だけでなく、JDBC データソース、JMS リソースも同じ内容となります。

運用管理コマンドから下記のように設定してください。

```
otxadmin> set server.jndi-service.url=corbaname://<host1>:<port2>,corbaname://<host2>:<port2>,...
```

※ここで指定するポートは WebOTX ドメインの名前サーバのポート番号で、WebOTX インストール時に作成される domain1 のポート番号の既定値は 2809 です。

設定後はドメインの再起動が必要です。

5.4. エラードキュメントの概要

ホームページの閲覧時に表示される「ファイルが見つからない」などのエラー表示画面を、エラードキュメントと呼びます。

このエラードキュメントをシステムで決められた内容ではなく、わかりやすい内容に変更して表示させることにより、ユーザにより多くの情報を提供し、正常なページと統一感を持たせることなどが可能になります。

エラードキュメントには、Web サーバが返すもの、Web コンテナが返すものなど複数のパターンがあり、発生するエラーに応じて表示される内容が変わってきます。

ここでは、エラードキュメントをカスタマイズする際に参考となるエラードキュメントの概要について説明します。

詳細な設定方法については、「5.4 エラードキュメントのカスタマイズ」を参照してください。

5.4.1. WebOTX Web サーバ(Apache)のエラードキュメント

問題やエラーが発生した時の動作として、WebOTX Web サーバ(Apache)には以下の4つのうちの1つの動作を設定することができます。WebOTX Web サーバ(Apache)ではこの設定を行うことによりエラードキュメントをカスタマイズすることができます。

- Apache 標準の簡単なエラーメッセージを表示
- 自分で指定したメッセージを表示
- 問題やエラーの処理をするために、自サーバ内の URL-path へリダイレクト
- 問題やエラーの処理をするために、外部の URL へリダイレクト

最初のものがデフォルトの動作で、2番目から4番目は、ErrorDocument ディレクティブにより、HTTP のレスポンスコードとメッセージか URL を指定することで設定します。

詳細は Apache のドキュメントを参照してください。

5.4.2. プラグインのエラードキュメント

マルチプロセス時に使用される IIOP プラグインには、以下の2パターンのエラーコードに対応したエラードキュメントを返す機能があります。デフォルトではプラグインはエラードキュメントを変更しません。

カスタマイズする時は手動で`$(INSTANCE_ROOT)/config/WebCont/ior_workers.properties` を変更する必要があります。

HTTP エラーコード

HTTP レスポンスのエラーコードに応じたエラー画面を返します。

404 や 500 といったエラーコードごとにエラードキュメントを指定することが出来ます。

CORBA エラーコード

CORBA エラーコードに応じたエラー画面を返します。

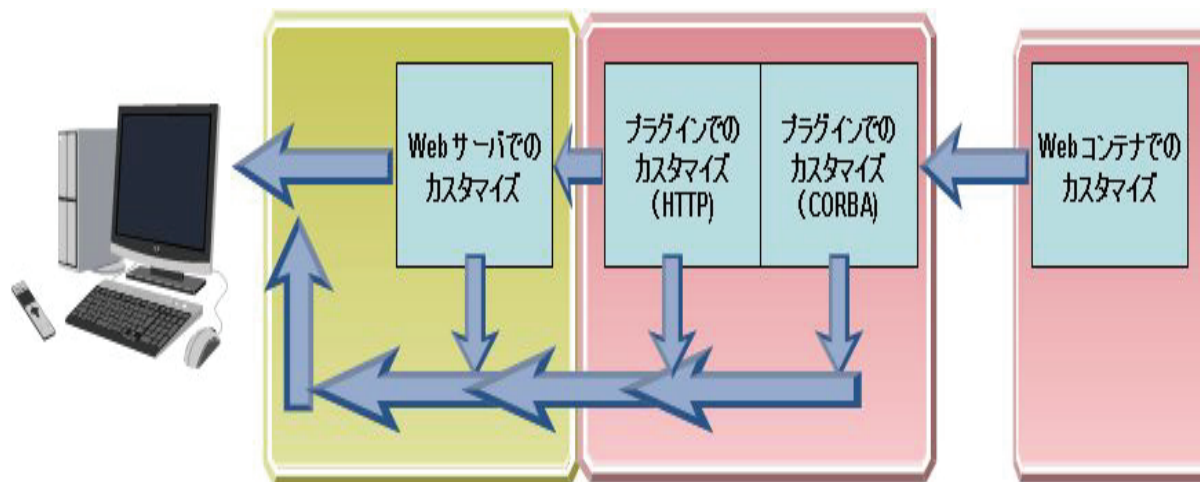
CORBA::NO_RESPONSE や CORBA::INV_OBFRREF といった CORBA のエラーコードごとにドキュメントを指定することができます。またエラーコードはマイナーコードごとに指定することも可能です。マイナーコードの指定方法には、マイナーコードを列挙する方法やマイナーコードの範囲を指定する方法などが利用できます。

5.4.3. Web コンテナのエラードキュメント

Web コンテナ側ではアプリケーションごとに HTTP エラーコードに応じたエラー画面を返すことができます。

5.4.4. カスタマイズの併用

下図はマルチプロセスモード時のエラードキュメントカスタマイズのイメージ図です。矢印がデータの流れを表します。



上図のように、エラードキュメントのカスタマイズは複数の箇所を設定することが出来ますが、「プラグインでの CORBA エラーカスタマイズ」が最も優先されます。優先順位は優先順位が高い方から順に、「プラグインでの CORBA エラーカスタマイズ」、「プラグインでの HTTP エラーカスタマイズ」、「Web サーバでのカスタマイズ」、「Web コンテナでのカスタマイズ」となります。また、発生するエラーの内容、発生場所などによって、有効なカスタマイズの箇所が変わってきます。

例として、障害によりプラグインと Web コンテナの通信ができなかった場合を考えます。この場合、IIOP プラグインでは「HTTP の 500 エラー」と「CORBA の NO_RESPONSE エラー」が発生します。

この時、Web コンテナでエラーが発生していても、通信が出来ないため Web コンテナでのカスタマイズは無効となります。また Apache で 500 エラー画面のカスタマイズを行っていた場合、このカスタマイズは有効となりますが、同様にプラグインでも 500 エラーのカスタマイズを行っていた場合、ブラウザにはプラグインでカスタマイズされた画面が返ります。また、プラグインで CORBA エラーのカスタマイズと HTTP エラーのカスタマイズを同時に行っていた場合、CORBA エラーのカスタマイズの画面が返されます。

さらに、動的反映のオン・オフや、設定ファイルへのコンテキスト名の記述の有無なども有効なカスタマイズ箇所に影響を与えます。下表は設定ファイルへのコンテキスト名の記述の有無と動的反映の設定によるタイプ分けの表です。

コンテキスト名の記述	動的反映	タイプ
あり	-	A
なし	ON	B
	OFF	C
	query once	A(初回リクエスト時)/C(2回目以降) (※)

(※)動的反映オプションに query once を指定した場合、Web サーバ起動後最初のリクエスト時に、Web コンテナに対してコンテキスト問い合わせの query が送信され、その後は問い合わせが行われません。そのため、初回リクエスト時に存在していたコンテキストに対するリクエストと、その後新規に Web アプリケーションが配備されたコンテキストに対するリクエストの動作が異なります。

下表は WebOTX Application Server V7.1 で WebOTX Web サーバ 2.0 使用時の一般的なエラー要因、デフォルトで表示されるエラー画面とエラーコード、カスタマイズ箇所ごとで検出できるエラーコードの例です(別表タイプは上表を参照)。カスタマイズ時の参考にしてください。

例として、別表タイプが A でプラグインにエラー要因(リスナと通信できない)がある場合、デフォルトでは Web サーバ (Apache)の 404 の画面が表示されます。この時プラグインでは CORBA::NO_RESPONSE(マイナーコード 1141)が検出されます。また、WebOTX Web サーバでも 404 エラーが検出されます。そのため、プラグインでの CORBA エラー画面カスタマイズと WebOTX Web サーバでのエラー画面カスタマイズが有効になります。両方を設定した場合は優先順位により CORBA エラーカスタマイズが有効になります。この時プラグインでの HTTP エラー画面カスタマイズとコンテナでのエラー画面カスタマイズは無効であるため、カスタマイズの設定を行っていても効果はありません。

マルチプロセスモード時											
別表タイプ	プラグイン	配備	エラー要因				デフォルトのエラー画面	カスタマイズできるエラーコード			
			アプリケーショングループ	プロセスグループ	アプリケーション	ファイル		プラグイン		WebOTX Webサーバ	コンテナ
								CORBA エラー	HTTP エラー		
C	-	-	-	-	-	-	Webサーバ 404	無効	無効	404	無効

A	リスナと通信できない	-	-	-	-	-	Webサーバ 404	NO_RESPONSE (1141)	無効	404	無効
B	リスナと通信できない	-	-	-	-	-	Webサーバ 500	NO_RESPONSE (1141)	500	500	無効
A	正常	配備なし	-	-	-	-	Webサーバ 404	INV_OBJREF (3960)	無効	404	-
B	正常	配備なし	-	-	-	-	Webサーバ 404	INV_OBFREF (3960)	404	404	-
A	正常	正常	停止	-	-	-	Webサーバ 404	INV_OBJREF (3960)	無効	404	無効
B	正常	正常	停止	-	-	-	Webサーバ 404	INV_OBFREF (3960)	404	404	無効
A	正常	正常	正常	停止	-	-	Webサーバ 404	INV_OBJREF (3960)	無効	404	無効
B	正常	正常	正常	停止	-	-	Webサーバ 500	INV_OBJREF (3880)	500	500	無効
A/B	正常	正常	正常	正常	停止	-	コンテナ 404	無効	404	無効	404
A/B	正常	正常	正常	正常	正常	存在しない	コンテナ 404	無効	404	無効	404

シングルプロセスモード時							
別表タイプ	エラー要因				デフォルトの エラー画面	カスタマイズできる エラーコード	
	プラグインの 状態	配備	アプリケーション	ファイル		WebOTX Web サーバ	コンテナ
C	-	-	-	-	Web サーバ 404	404	無効
A	リスナと通信 できない	-	-	-	Web サーバ 503	503	無効
B	リスナと通信 できない	-	-	-	Web サーバ 404	404	無効
A	正常	配備なし	-	-	コンテナ 404	無効	-
B	正常	配備なし	-	-	Web サーバ 404	404	-
A/B	正常	正常	停止	-	コンテナ 503	無効	503
A/B	正常	正常	正常	存在しない	コンテナ 404	無効	404

5.5. エラードキュメントのカスタマイズ

5.5.1. WebOTX Web サーバ(Apache)でのカスタマイズ

WebOTX Web サーバ(Apache)でエラードキュメントをカスタマイズする場合、手動で
`#{INSTANCE_ROOT}/config/WebServer/httpd.conf` に `ErrorDocument` ディレクティブを追加してください。

ErrorDocument ディレクティブ

構文: `ErrorDocument error-code document`

パラメータ説明:

error-code : HTTP レスポンスのエラーコードを指定します。

document : 表示するドキュメントの URL かブラウザに表示するメッセージを指定します。URL にはスラッシュ(/)で始まるローカルの web-path か、クライアントが解決できる完全な URL を指定します。完全な URL を指定した場合、Apache は同じサーバ上であってもリダイレクトをクライアントに送出するため、クライアントは元々のエラーコードではなく、リダイレクトのエラーコードを受け取ります。

※WebOTX Web サーバ 1.3 でメッセージを使用する場合、メッセージの先頭に、対になっていない二重引用符を付けます(例 “message”)。WebOTX Web サーバ 2 でメッセージを使用する場合は、メッセージを二重引用符で囲みます(例 “message”)。

設定例

```
ErrorDocument 500 http://foo.example.com/cgi-bin/tester
ErrorDocument 404 /cgi-bin/bad_urls.pl
ErrorDocument 401 /subscription_info.html
ErrorDocument 403 "Sorry can't allow you access today"
```

※ErrorDocument 401 にリモートの URL を指定した場合、クライアントは 401 のステータスコードを受け取らないため、パスワードをユーザに入力要求しなければならないことがわかりません。従って、ErrorDocument 401 というディレクティブを使う場合は、必ずローカルな文書を参照しなければなりません。

詳細は Apache のドキュメントを参照してください。

5.5.2. IIOP プラグインでの HTTP エラー画面カスタマイズ

IIOP プラグインで HTTP レスポンスに応じたエラー画面をカスタマイズする場合、手動で `#{INSTANCE_ROOT}/config/WebCont/ior_workers.properties` に下記プロパティを追加してください。

プロパティ	説明	書式・既定値
<code>worker.otxiop.http_error.list</code>	<p>エラードキュメントを変更する HTTP レスポンスの Status-Code の一覧を“, ”区切りで列挙します。</p> <p>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>このプロパティは複数定義できません。</p>	<p>書式:</p> <pre>worker.otxiop.http_error.list=<Status-Code>[,<Status-Code>]...</pre> <p>パラメータ説明:</p> <p><Status-Code> : Web コンテナから返ってくる HTTP レスポンスの Status-Code</p> <p>既定値:なし</p>
<code>worker.otxiop.http_error.<Status-Code>.doc</code>	<p>worker.otxiop.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.otxiop.http_error.<Status-Code>.doc=<File URL></pre> <p>パラメータ説明:</p> <p><Status-Code> : 前述を参照</p> <p><File URL> : file://で始まる URL</p> <p>既定値:なし</p>
<code>worker.otxiop.http_error.charset</code>	<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ーセットを指定します。</p>	<p>書式:</p> <pre>worker.otxiop.http_error.charset=<charset></pre> <p>パラメータ説明:</p> <p><charset> : 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ーセット</p> <p>既定値:ISO-8859-1</p>

<code>worker.otxiiop.http_error.<Status-Code>.charset</code>	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。	書式: <code>worker.otxiiop.http_error.<Status-Code>charset=<charset></code> パラメータ説明: <charset> : Status-Code のエラードキュメントのファイルキャラクタセット 既定値: <code>worker.otxiiop.http_error.charset</code> の値
--	---	---

設定例

```

worker.otxiiop.http_error.list=404,500
worker.otxiiop.http_error.charset="Shift-JIS"
worker.otxiiop.http_error.404.doc=file:///error/404error.html
worker.otxiiop.http_error.500.doc=file:///error/500error.html
worker.otxiiop.http_error.500.charset="ISO-8859-1"

```

5.5.3. IIOP プラグインでの CORBA エラー画面カスタマイズ

IIOP プラグインで CORBA エラーコードに応じたエラー画面をカスタマイズする場合、手動で `$(INSTANCE_ROOT)/config/WebCont/ior_workers.properties` に下記プロパティを追加してください。

プロパティ	説明	書式・既定値
<code>worker.otxiiop.corba_error.list</code>	エラードキュメントを変更する CORBA の Error-Name の一覧を", "区切りで列挙します。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。またこのプロパティは複数定義できません。	書式: <code>worker.otxiiop.corba_error.list=<Error-Name>[, <Error-Name>]...</code> パラメータ説明: <Error-Name> : Web コンテナから返ってくる CORBA エラーの名称 既定値: なし ※Error-Name に使用できる文字はアルファベットの小文字のみです。"_", "-", "="等は使用できません。

worker.otxiiop.corba_error.charset	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>書式:</p> <p>worker.otxiiop.corba_error.charset=<charset></p> <p>パラメータ説明:</p> <p><charset> : エラードキュメントファイルのデフォルトのファイルキャラクタセット</p> <p>既定値:ISO-8859-1</p>
worker.otxiiop.corba_error.<Error-Name>.err	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>書式:</p> <p>worker.otxiiop.corba_error.<Error-Name>.err=<エラー定義>[,<エラー定義>]...</p> <p>パラメータ説明:</p> <p><Error-Name> : worker.otxiiop.corba_error.list で指定した CORBA エラーの名称</p> <p><エラー定義> : <CORBA-ErrorName> [(<MinorCode-列挙> <MinorCode-範囲指定>)]</p> <p><CORBA-ErrorName> : Web コンテナから返ってくる CORBA のエラー名</p> <p><MinorCode-列挙> : <CORBA-MinorCode>[,<CORBA-MinorCode>]...</p> <p><MinorCode-範囲指定> : <CORBA-MinorCode>-<CORBA-MinorCode></p> <p><CORBA-MinorCode> : Web コンテナから帰ってくる CORBA のマイナーコード</p> <p>既定値:なし</p>
worker.otxiiop.corba_error.<Error-name>.doc	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>書式:</p> <p>worker.otxiiop.corba_error.<Error-Name>.doc=<FileURL></p> <p>パラメータ説明:</p> <p><Error-Name> : 前述を参照</p> <p><FileURL> : file://で始まるURL(フルパスで指定)</p> <p>Windows での URL 例 : file:///C:/err/err.html</p> <p>LinuxでのURL例 : file:///err/err.html</p> <p>既定値:なし</p>

<code>worker.otxiiop.corba_error.<Error-Name>.charset</code>	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。	書式: <code>worker.otxiiop.corba_error.<Error-Name>.charset=<charset></code> パラメータ説明: <charset> : 指定した 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 既定値: <code>worker.otxiiop.corba_error.charset</code> の値
--	---	--

5.5.4. Web コンテナでのカスタマイズ

Web コンテナでアプリケーションごとにエラードキュメントをカスタマイズする場合、web.xml に手動で下記要素を追加してください。要素の追加場所は<web-app>タグの中である必要があります。追加場所は Servlet 仕様で決まっているため、詳細は Java Servlet specification を参考にしてください。

追加要素:

```
<error-page>
<error-code>err-code</error-code>
<location>document</location>
</error-page>
```

パラメータ説明:

err-code : Web コンテナが返す HTTP レスポンスのエラーコードを指定します。
document : 表示させるドキュメントを指定します。

記述例(赤字は追加箇所)

```
<?xml version="1.0" encoding="UTF-8"?><web-app xmlns="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="2.4"
xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <display-name xml:lang="ja">WebAP</display-name>
  <servlet>
    <servlet-name>index</servlet-name>
    <jsp-file>/welcome.jsp</jsp-file>
  </servlet>
  <servlet-mapping>
    <servlet-name>index</servlet-name>
    <url-pattern>/welcome.jsp</url-pattern>
  </servlet-mapping>
```

```

<error-page>
  <error-code>404</error-code>
  <location>/error/404document.html</location>
</error-page>
<error-page>
  <error-code>500</error-code>
  <location>/error/500document.html</location>
</error-page>
</web-app>

```

5.6. 連携設定の自動化

WebOTX Application Server V7.1 では、環境設定ツールをコマンドラインから引数を指定して呼び出すことにより、各 Web サーバの「外部 Web サーバへの WebOTX 連携情報の設定」の設定をコマンドラインから行うことができます。(V6.x ではこの機能は利用できません)

ここではコマンドラインから環境設定ツールを使用する手順について説明します。

連携設定の自動化を行う際は、各 Web サーバとの連携設定の項目も参照してください。

ドメインプロパティファイルの作成

環境設定ツールが読み込むドメインプロパティファイルを作成します。\${INSTALL}/domain1.properties を参考に作成してください。

ドメインプロパティファイルは\${INSTALL}配下に保存しておく必要があります。

ドメインプロパティファイルに記述するプロパティは下表を参照してください。

プロパティ	説明
domain.path	連携設定を行うドメインのパスを指定します。 記述が必要な Web サーバ: 全ての Web サーバ 記述例: domain.path=C:¥WebOTX¥domains¥domain1

<p>domain.webserver.type</p>	<p>ドメインと連携する Web サーバの種類を指定します。</p> <p>WebOTX Web サーバ 1.3 を使用する場合は「WebServer」、WebOTX Web サーバ 2.0 を使用する場合は「WebServer2」、Sun Java System Web Server (Sun ONE) 6.x を使用する場合は「sun」、IIS を使用する場合は「iis」を指定してください。大文字・小文字は区別されます。</p> <p>記述が必要な Web サーバ: 全ての Web サーバ</p> <p>記述例: domain.webserver.type=WebServer2</p>
<p>domain.webserver.path</p>	<p>ドメインと連携する Web サーバのパスを指定します。</p> <p>記述が必要な Web サーバ: WebOTX Web サーバ 1.3 / WebOTX Web サーバ 2.0 Sun Java System Web Server (Sun ONE) 6.x</p> <p>記述例: domain.webserver.path=C:\¥WebOTX¥WebServer2</p>
<p>domain.webserver.name</p>	<p>Sun Java System Web Server (Sun ONE) 6.x を使用する場合は、複数の内部 Web サーバを構築した場合に指定する「ServerName」を指定します。</p> <p>IIS を使用する場合は、「Web サイトの識別名」を指定します。</p> <p>記述が必要な Web サーバ: Sun Java System Web Server (Sun ONE) 6.x IIS</p> <p>記述例: domain.webserver.name="既定の Web サイト"</p>
<p>server.web-container.processMultiplicity</p>	<p>インストール時にマルチプロセスモードを選択した場合には「true」を、その他の場合は「false」を指定します。※マルチプロセスモードを選択できるのは Standard/Enterprise Edition のみです。</p> <p>記述が必要な Web サーバ: 全ての Web サーバ</p> <p>記述例: domain.webcontainer.processMultiplicity=true</p>
<p>domain.webserver.plugin.querymode</p>	<p>query once 機能を使用する場合には「once」を指定してください。</p> <p>※動的反映オフの設定は環境設定ツールからは行えません。動的反映オフの設定を行う場合は、上記設定を行った後、「5.1.1 動的反映オフの設定」を参照して、設定ファイルを変更してください。</p> <p>記述が必要な Web サーバ: 全ての Web サーバ (query once 機能を利用しない場合は不要)</p> <p>記述例: domain.webserver.plugin.querymode=once</p>

ドメインプロパティファイル記述例(#はコメントとして認識されます)

```
domain.path=C:¥WebOTX¥domains¥domain1
domain.webserver.type=iis
#domain.webserver.path=
domain.webserver.name="既定の Web サイト"
server.web-container.processMultiplicity=false
#domain.webserver.plugin.querymode=once
```

環境設定ツールの実行

Windows の場合

```
C:¥>cd C:¥WebOTX
C:¥WebOTX> bin¥SetInfo.exe -s filename
```

Linux の場合

```
root> cd /opt/WebOTX
root> ./bin/SetInfo.sh -s filename
```

6. ライセンスについて

ここでは、外部 Web サーバと Web コンテナが別マシン上にある場合に必要となるライセンスについて説明します。外部 Web サーバと Web コンテナが同一マシン上にある場合は、追加で必要なライセンスはありません。

WebOTX Web サーバを使用する場合

WebOTX Web サーバのインストールには WebOTX Web/Standard-J/Standard/Enterprise Edition のいずれかのライセンスが必要です。そのため、Web コンテナとは別マシンで WebOTX Web サーバを利用する場合、Web サーバを動作させるマシン用に上記のライセンスを購入し、WebOTX をインストールしてください。

その他の外部 Web サーバを使用する場合

Web コンテナ動作マシンと外部 Web サーバ動作マシンの OS が異なる場合、連携に必要なプラグインを入手するために、WebOTX Web/Standard-J/Standard/Enterprise Edition のいずれかのライセンスを購入し WebOTX をインストールすることが必須です。

Web コンテナ動作マシンと外部 Web サーバ動作マシンの OS が同じ場合でも、WebOTX Web/Standard-J/Standard/Enterprise Edition のライセンスを購入し、Web サーバ動作マシンに WebOTX をインストールすることを推奨します。

Web コンテナ動作マシンと外部 Web サーバ動作マシンの OS が同じ場合、Web サーバプラグインやその他の必要なファイルを Web サーバマシンにコピーし、連携動作させることは可能ですが、この場合、Web サーバとの連携動作についてはサポートの対象外となります。設定については各自の責任で行ってください。

完全なサポートを受けるためには WebOTX Web/Standard-J/Standard/Enterprise Edition いずれかのライセンスの購入が必要です。

7. 注意事項

注意事項について説明します。

- ドメインをコピーした場合の注意事項

WebOTX V6.5 でマルチプロセスモード時に、設定ファイルが既に存在するドメインに対して環境設定ツールを使用すると、設定ファイルが正常に更新されません。連携設定前に下記の設定ファイルを削除してから環境設定ツールを起動してください。

WebOTX Web サーバ 1.3 を使用する場合

```
${INSTANCE_ROOT}/config/WebCont/mod_jk_om.conf-auto
```

WebOTX Web サーバ 2.0 を使用する場合

```
${INSTANCE_ROOT}/config/WebCont/mod_jk_om-20.conf-auto
```

- Web サービスの利用に関する注意事項

WebOTX V6.4 以前で外部 Web サーバとの連携を行った環境で EJB サービスエンドポイントを動作させる場合、EJB サービスエンドポイントを配備したドメインに、下記 Java VM オプションを追加する必要があります。※WebOTX V6.5 以降を使用する場合は不要です。

オプションの追加方法

運用管理コマンドを起動し、次の例のように実行します。

```
otxadmin> login --port 6212 --user admin --password adminadmin
otxadmin> create-jvm-options -Dwebotx.webcontainer.serverconfig.NoRoot=false
otxadmin> exit
```

実行後、ドメインを再起動してください。

- 外部 Web サーバ(Java ベースの内蔵 Web サーバ以外の Web サーバ)と連携したときには、複数の virtual server は利用できません。
- Web サーバと連携中に配備解除した Web アプリケーションにアクセスするなどした場合(通常は HTTP404 エラー)、ブラウザの設定によりダウンロード処理が始まる場合があります。これはブラウザの設定を変更する事で回避することが出来ます。

例)IE6 の場合

メニューより「ツール」-「インターネットオプション」-「詳細設定」で「ブラウズ」-「HTTP エラーメッセージを簡易表示する」のチェックを ON にする。

- V6.22 以降のリリースにおけるドキュメントルートについての注意事項

V6.22 以降のリリースでは、外部 Web サーバ連携時に表示するドキュメントルートが、V6.20、および V6.21 と異なります。V6.20 および V6.21 では、デフォルトの状態ではドメインの docroot 配下をドキュメントルートとして表示しますが、V6.22 以降のリリースでは、連携した Web サーバのドキュメントルートを表示します。WebOTX Web サーバを使用している場合、ドメインのドキュメントルートと WebOTX Web サーバのドキュメントルートは同じであるため、V6.20 および V6.21 と動作は変わりません。

V6.22 以降のリリースで V6.20 および V6.21 と同じようにドメインの docroot 配下を表示するには、JavaVM のオプションに「-Dwebotx.webcontainer.serverconfig.NoRoot=false」を設定してドメインと Web サーバを再起動してください。

また、V6.22 以降のリリースでルートコンテキスト("/")に割り当てたアプリケーションにアクセスするためには JavaVM のオプションに「-Dwebotx.webcontainer.serverconfig.NoRoot=false」を設定してドメインと Web サーバを再起動する必要があります。

- 外部 Web サーバと連携した場合、logs ディレクトリに「jk_runtime_status」と「jk_runtime_status.lock」というファイルが生成されます。これらは、JK コネクタ (Web サーバプラグイン) の制御ファイルです。
- IIS+Web コンテナの環境で、リクエスト URL に「/scripts/jrun.dll/」という文字列が先頭に付加されている場合に JSP ファイルを実行するとエラー (404) が発生します。この現象は IIS+Jrun の環境を過去に構築した事があると発生します。IIS の script ディレクトリに「jrun.ini」があるので、その中で「scripts/jrun.dll」の文字列が付加されていれば、「jrun.ini」を削除 (退避) してください。