

CLUSTERPRO によるクラスタシステム構築ガイド

WebOTX V6 編

目次

1. 概要	2
1.1 片方向スタンバイ型	2
1.2 双方向スタンバイ型	4
2. Windowsでの環境構築の手順	6
2.1 WebOTXのインストール	6
2.2 クラスタ環境の構築	6
2.2.1 片方向スタンバイ型	6
2.2.2 双方向スタンバイ型	15
2.2.3 サンプルスクリプト	25
2.2.4 ドメイン作成プロパティ・ファイル	29
3. Linuxでの環境構築の手順	33
3.1 事前準備	33
3.2 クラスタ環境の構築	33
3.2.1 片方向スタンバイ型	33
3.2.2 双方向スタンバイ型	42
3.2.3 サンプルスクリプト	52
4. クライアントの仮想IPアドレスの指定方法	56
4.1 Java	56
4.1.1 Enterprise Bean (EJB)	56
4.1.2 CORBA	56
4.2 CORBA C++	56
4.2.1 Windows (Visual C++)	57
4.2.2 Linux (gcc)	57
4.3 Visual Basic	57
5. Oracle RAC連携	58

1. 概要

本書は、WebOTX バージョン 6 に対して CLUSTERPRO 製品を導入してクラスタシステムを構築する際に役立つガイドとして提供しています。CLUSTERPRO は、Windows 用と Linux 用の製品があります。WebOTX は、Windows 用に対しては、V6.2 を、Linux 用に対しては、V6.3 で評価を実施しました。2 章からは、各プラットフォームで評価を行った際の実例を元に、WebOTX や CLUSTERPRO の設定や定義の方法を示していきます。WebOTX の評価に用いたバージョンが 6.2 と 6.3 ですが、バージョン 6.1 でも導入実績がありますので、基本的に WebOTX V6 シリーズでは問題なく動作します。読者の方が利用する WebOTX バージョンが本書と異なっている場合は、WebOTX の設定項目名に若干の相違がありますが、適宜読み替えながら参照頂ける内容になっています。

以降、CLUSTERPRO 環境下で WebOTX を利用する際の機能概要について記述します。

WebOTX の運用形態には片方向スタンバイ型と双方向スタンバイ型の 2 つがあります。クラスタ製品によっては、片方向スタンバイのことをシングルスタンバイ、双方向スタンバイのことをマルチスタンバイと呼ぶこともあります。本書では、CLUSTERPRO の用語に合わせて片方向、双方向として表現することにします。

片方向スタンバイ型のクラスタでは、一方のサーバを現用系として業務を稼働させ、他方のサーバを待機系としてサーバを稼働させない運用形態をとります。

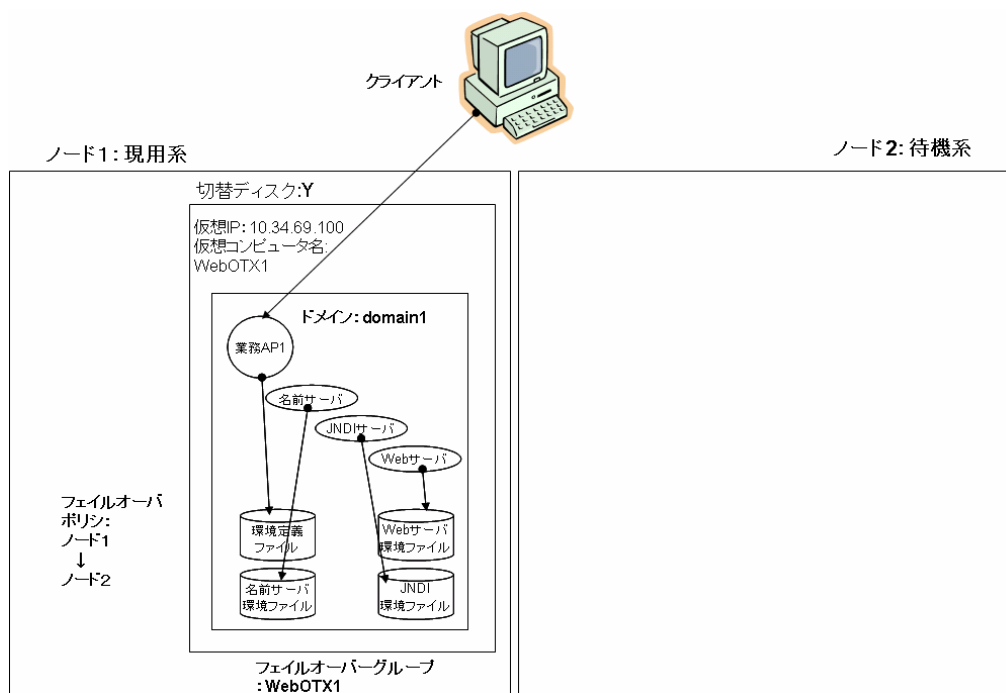
双方向スタンバイ型のクラスタでは、複数のサーバで業務を稼働させ、相互に待機する運用形態です。WebOTX では、1 つのサーバに 2 つのドメインを形成させ、一方のサーバでドメイン A を現用、ドメイン B を待機させ、他方のサーバでドメイン A を待機、ドメイン B を現用にする形態をとります。

WebOTX のドメインとは、アプリケーションサーバのインスタンスのような概念です。それは、アプリケーションサーバを異なる運用管理単位にグループ化されます。このようにドメイン単位にアプリケーションサーバのインスタンスを分割できることで、1 つのアプリケーションサーバのインストールで、複数の異なる管理構成を構築できる利点があります。

1.1 片方向スタンバイ型

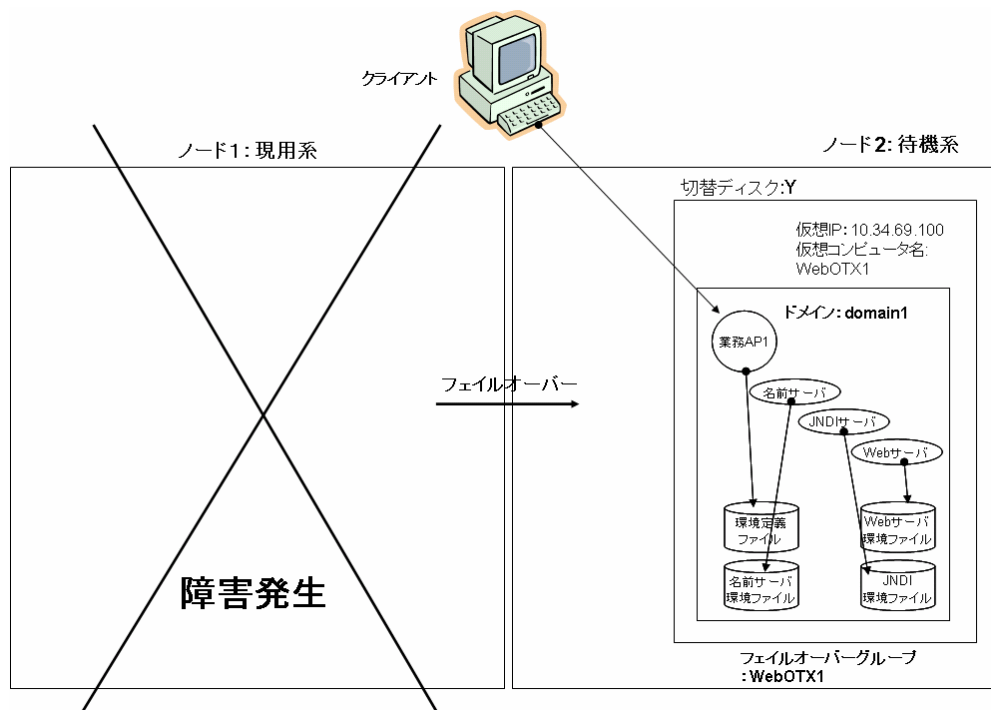
片方向スタンバイ型では、1 つの切り替えパーティションを用い、その切り替えパーティション上に WebOTX のドメインを配置します。現用系で障害等が発生し、現用系の WebOTX を含んだフェイルオーバーグループがフェイルオーバーすると、待機系でフェイルオーバーグループが起動し、WebOTX が起動します。そのときに、フェイルオーバーグループに割り当てられた仮想 IP アドレスを待機系が引き継ぎます。その時、現用系で実行していたアプリケーションなどを引き継ぐことができるので、クライアントは再接続を行うだけで引き続き業務を行うことができます。

下の図は、片方向スタンバイ型を CLUSTERPRO 環境下で動作させるときのイメージ図です。



クライアントはWebOTXシステムが現在どのノードで動作しているのかを意識せずに、定義された仮想 IP アドレスを用いてノードに接続します。対応する仮想 IP アドレスはその時点で移動しているノードに割り当てられています。現用系の WebOTX1 の設定ファイルは切り替えパーティション上に格納されます。

現用系のノードに障害が発生すると、次の図のように遷移します。

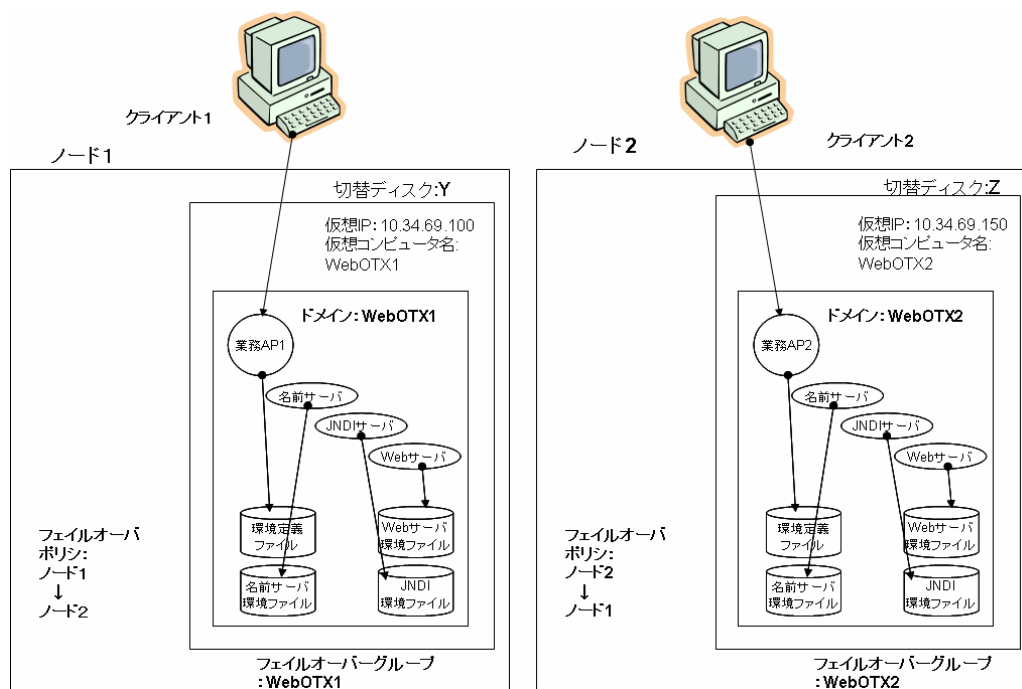


WebOTXはCLUSTERPROのスクリプトに応じて待機系ノードで起動します。障害発生時には、現用系で動作していた環境を待機系に移動させます。また、仮想 IP アドレスが待機系ノードに割り当てられるために、クライアントは再接続を行なうだけで、同一の仮想 IP アドレスで接続することが可能です。

1.2 双方向スタンバイ型

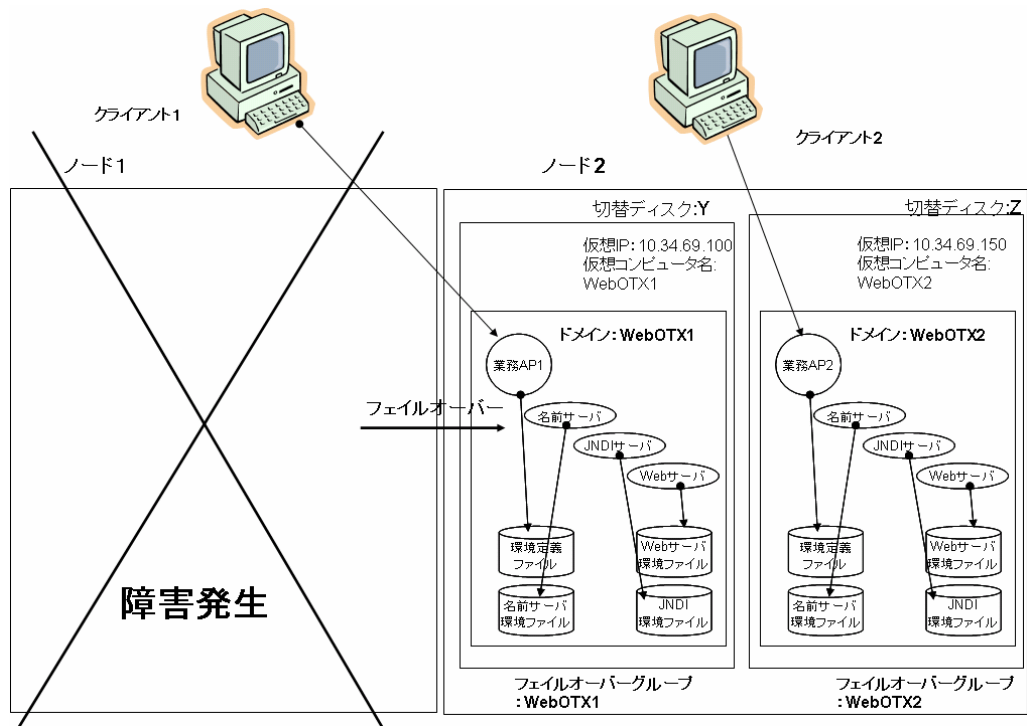
双方向スタンバイ型は、複数の切り替えパーティションを用い、個々のノードに WebOTX を含むフェイルオーバーグループをスタートさせる形態です。各ノードに割り当てられた切り替えパーティション上に WebOTX のドメインを配置します。クライアントは接続時にフェイルオーバーグループで設定した仮想 IP アドレスを用いて接続を行います。接続しているノードで障害が発生した場合、起動していたフェイルオーバーグループはフェイルオーバーし、他のノードに移動します。クライアントは再接続を行うことで、他のノードで動作する WebOTX へ接続することができ、業務を継続して行うことが可能となります。

下図は双方向スタンバイ型を CLUSTERPRO 環境下で動作させるときのイメージ図です。



WebOTXシステムはそれぞれ別のフェイルオーバーグループに属し、個別の仮想IPアドレスを保持しています。クライアントは仮想IPアドレスを用いてノードに接続します。

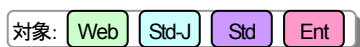
あるノードに障害が発生した場合、フェイルオーバーポリシーに従い、以下の図のように状態が遷移します。



フェイルオーバーによりノード1で動作していたフェイルオーバーグループ WebOTX1 がノード2へ移行されます。これにより、それまでのノード1の WebOTX に接続していたクライアントは、同一の仮想 IP アドレスに再接続するだけでノード2上での WebOTX1 が利用可能となります。

2. Windows での環境構築の手順

この章での説明で用いた CLUSTERPRO 製品は、バージョン 8.0 の共有ディスク型のものです。各節の中に現れる作業手順毎に、対象となる WebOTX のエディションを示しています。



それぞれ、**Web** は Web Edition、**Std-J** は Standard-J Edition、**Std** は Standard Edition、**Ent** は Enterprise Edition に対応しています。

2.1 WebOTX のインストール

通常のインストール手順に従って、各ノード(クラスタ運用したいローカルマシン)に WebOTX 実行環境をインストールしてください。インストール時の設定はすべて同じになるようにしてください。

2.2 クラスタ環境の構築

2.2.1 片方向スタンバイ型

この手順は、後述のスクリプトを含めて、以下の環境を前提にしています。

N1 ノード、N2 ノード		
	フェイルオーバーグループ名	WebOTX1 (WebOTX ドメイン用)
	仮想 IP アドレス	10.34.69.100
	仮想ホスト名	WebOTX1
	切り替えパーティション	Z:

上記のリソースは実際の環境に読み替えてください。

I. CLUSTERPRO Manager の設定

はじめに、フェイルオーバーグループの設定を行います。作業はすべて、CLUSTERPRO Manager より行います。

1. CLUSTERPRO Manager でクラスタ名を選択して、[グループの追加]を行います。

ツールバーより、[クラスタ]→[グループ追加]を選択し、グループの名前の入力(WebOTX1)を行います。その後、プロパティ画面が表示されるので、全般ページに以下の設定を行います。

[リソースの設定]ボタンを押した後、リソースの設定画面が表示されます。

[代替ディスク]→[追加]を選択し、グループに割り当てる共有ディスク「Z:」を選択します。

[IP アドレス]→[フローティング IP 追加]を選択し、仮想アドレス(10.34.69.100)とネットマスクを設定します。

[仮想コンピュータ名]→[追加]で、WebOTX1 を追加します。

設定終了後[OK]を押すと、フェイルオーバーポリシーの設定ダイアログが表示されるので、フェイルオーバーさせたい順番

に追加を行ってください。本手順では、N1→N2 の順番で設定します。

以上で、ノードの下にフェイルオーバーグループの作成が完了します。

II. WebOTX 運用管理ツールでの設定

(注意)

以下の記述では、[] で囲まれた項目名は WebOTX V6.3 の場合で、() で囲まれた項目名は V6.2 での表示となります。運用管理ツールのイメージ図は V6.3 のものです。

1. 対象: WebOTX インストール・デフォルトのドメイン削除

各ノードで WebOTX のインストールを行い、次に N2 でデフォルトのドメインの削除を行います。

まず N1 と N2 ノードで WebOTX のサービス(WebOTX Agent Service)の起動を行います。N1 ノードを起動する目的は、手順 3 以降から N1 ノード上で作業を行うためです。N2 ノードを起動する目的は、N2 ノード内の WebOTX インストール・デフォルトの domain1 ドメインを削除するために WebOTX と対話する必要があるからです。

Windows サービスを GUI で起動操作する場合は、[コントロールパネル]→[管理ツール]→[サービス]から WebOTX Agent Service を起動します。net コマンドで起動する場合は、コマンド・プロンプトから以下のようにコマンド入力します。

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

Standard Edition 以上では WebOTX TPBASEadm サービスもありますが、これは WebOTX Agent Service を起動すると同時に起動されます。

次に N2 ノードでドメインの停止を行います。

```
otxadmin stop-domain domain1
```

その後、N2 ノードで WebOTX インストール・デフォルトのドメインを削除します。N2 ノードで使用するドメインは、後の手順の中で N1 ノードから作成した共有ディスク内のドメイン情報を利用するように再設定します。

```
ant -f setup.xml -Ddomain.name=domain1 delete-domain
```

以下、手順 2～11 は N1 ノードで作業を行います。

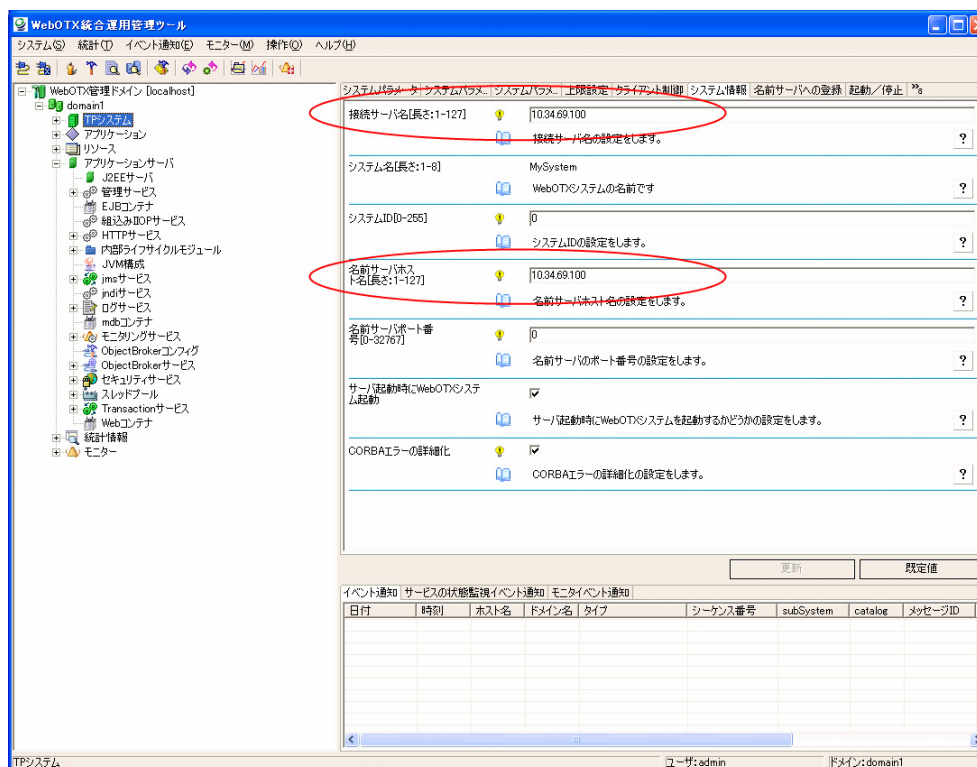
2. 対象: アプリケーションの配備

Web や EJB、CORBA などのアプリケーションを配備します。Standard / Enterprise Edition の場合は、アプリケーショングループの作成、プロセスグループの作成等、環境設定を行った後に配備を行います。

3. 対象: Web Std-J Std Ent TP システムへの仮想 IP アドレス設定

運用管理ツールの左側ツリーから[TP システム](tpsystem)を選択し、その右画面の[システム情報]タブを選んだ時の設定画面内から「接続サーバ名」と「名前サーバのホスト名」を仮想 IP アドレスに変更します。

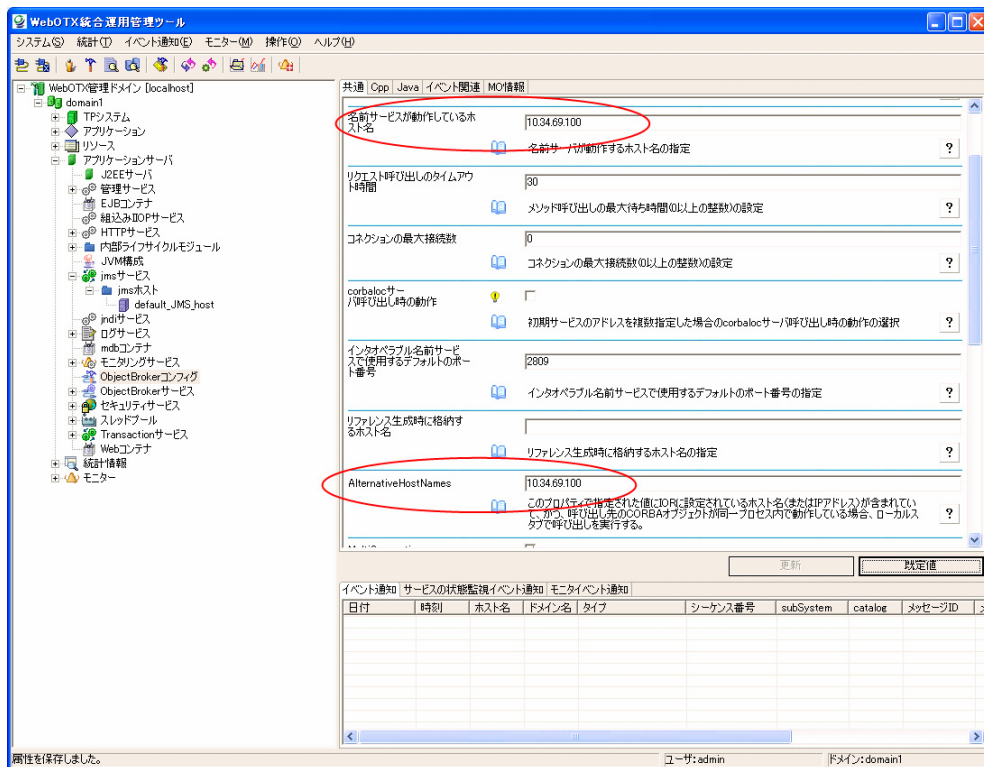
domain1: 10.34.69.100



4. 対象: Web Std-J Std Ent CORBA 通信基盤、Object Broker への仮想 IP アドレス設定

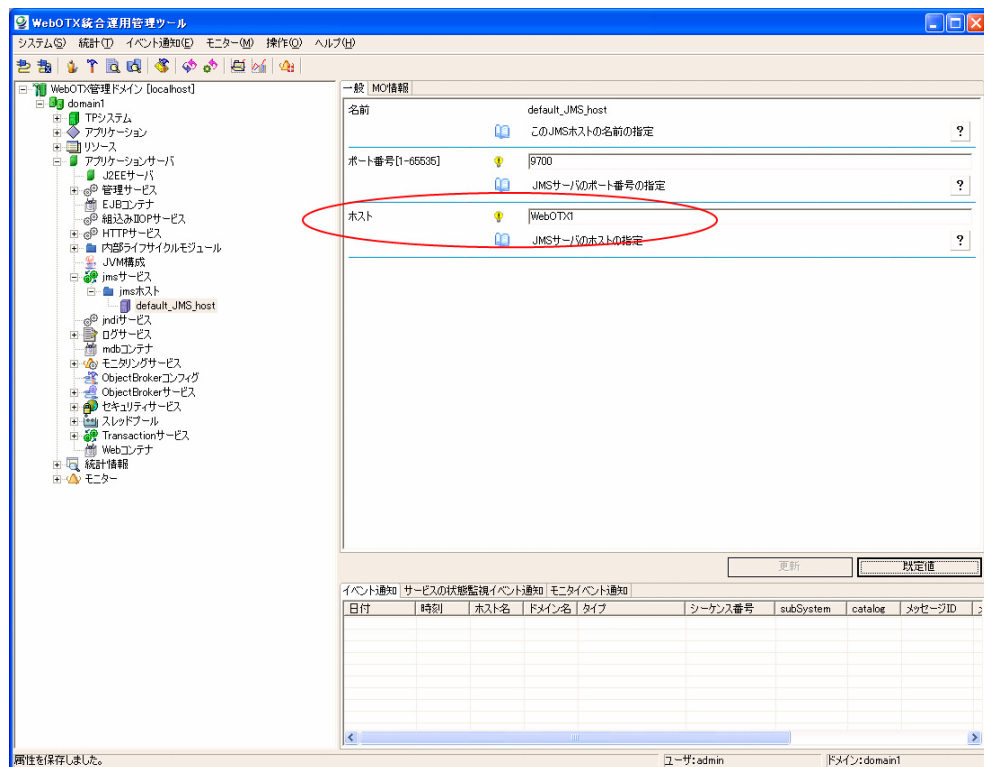
運用管理ツールの[アプリケーションサーバ](server)→[ObjectBroker コンフィグ](ObjectBrokerConfig)→[共通]タブを選択し、名前サービスが動作しているホスト名と AlternativeHostNames の値を仮想 IP アドレスに変更します。

domain1: 10.34.69.100



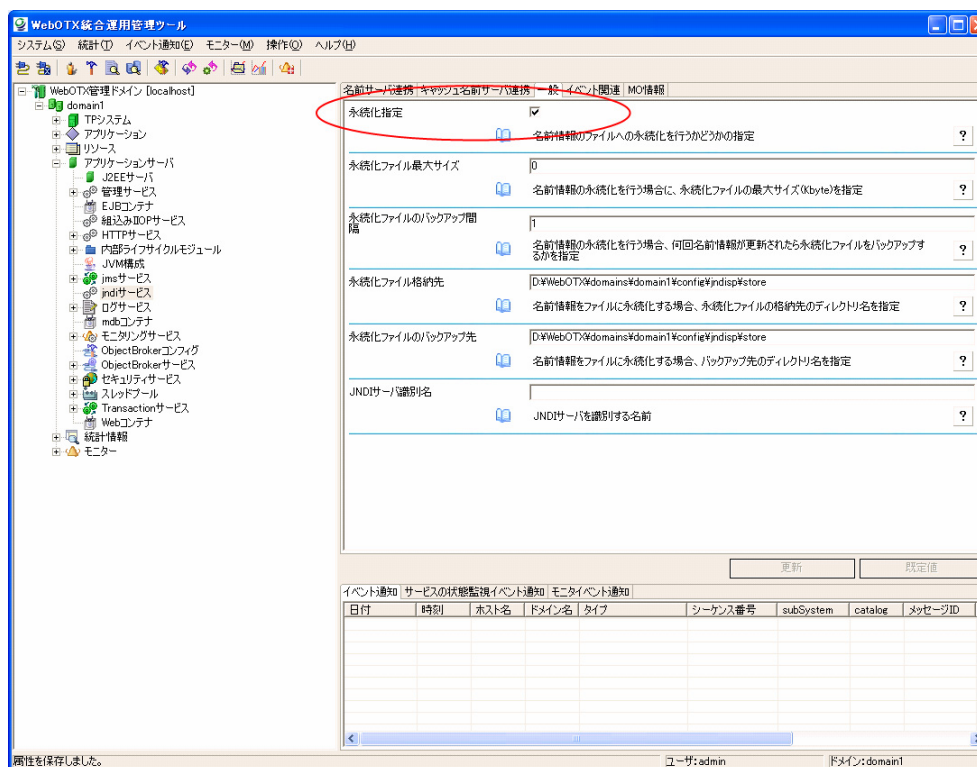
5. 対象: Web Std-J Std Ent JMS (Java Message Service)への仮想 IP アドレス設定

運用管理ツールの[アプリケーションサーバ](server)→[jms サービス](jms-service)→[jms ホスト](jms-host)→[default_JMS_host]→[一般]タブ内のホストを各仮想ホスト名に変更します。ただし、フェイルオーバーグループを N1で起動し、仮想 IP アドレスを有効にした状態で仮想ホスト名を設定してください。



6. 対象: Web Std-J Std Ent JNDI サーバへの永続化によるオブジェクト登録方式の解除

運用管理ツールにおいて、[システム]→[システム設定]を選択し、属性のレベルを全レベルに変更してください。運用管理ツールの[アプリケーションサーバ](server)→[jndi サービス](jndi-service)→[一般]タブ内の永続化をチェックしてください。



7. 対象: Web Std-J Std Ent WebOTX サービス起動方法の変更

N1、N2 ノードにおいて、WebOTX Agent Service サービスを停止してください。その後、サービスの起動方法を手動に変更してください。[コントロールパネル]→[管理ツール]→[サービス]から、またはコマンドから停止してください。

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

[サービス]画面でWebOTX Agent Serviceを右クリックし、プロパティを選択してください。スタートアップの種類を手動に変更してください。Standard Edition 以上の場合は、WebOTX TPBASEadm サービスに対しても同じ操作を行います。

8. 対象: Web Std-J Std Ent Object Broker の名前サーバ永続情報の削除

WebOTX インストールディレクトリ下の「domains¥domain1¥config¥ObjectBroker」フォルダ内にある「namesv.ndf」ファイルを削除してください。

9. 対象: Web Std-J Std Ent トランザクション・サービスへの仮想 IP アドレス設定

WebOTX インストールディレクトリ下の「domains¥domain1¥config¥TS¥jta.conf」ファイルをエディタで開き、[JTA]セッションの下に下記を追加します。

LogicalHostname="仮想 IP アドレス、または、仮想コンピュータ名"

10. 対象: Web Std-J Std Ent ドメイン起動制御ファイルのコピー

N1 ノードにおいて、WebOTX インストールディレクトリ下の「domains¥WebOTXAdmin¥config」フォルダにある、domain-config.xml ファイルを N2 ノードの domain-config.xml に上書きしてください。

11. 対象: Web Std-J Std Ent N1 ノードでのフェイルオーバーグループ起動

N1 ノードでフェイルオーバーグループ WebOTX1 を起動します。起動後、N1 ノードから切替パーティション Z を参照できるようになります。Z ドライブに domains フォルダを作成します。Z ドライブに N1 ノードの domains 配下の domain1 と WebOTXAdmin フォルダを Z:¥domains にコピーします。次からは以下のようなディレクトリ構成になっているとします。

Z:¥domains¥domain1
Z:¥domains¥WebOTXAdmin

以下の手順 12～15 では N1、N2 両方のノードで同じ作業をしてください。

12. 対象: Web Std-J Std Ent ドメイン環境変数の定義ファイルの修正

WebOTX インストールディレクトリ下の「config¥asenv.bat」ファイルを以下のように編集します。

```
set AS_ANT=D:¥WebOTX¥lib¥ant
set AS_ANT_LIB=D:¥WebOTX¥lib¥ant¥lib
set AS_WEBSERVICES_LIB=D:¥WebOTX¥lib
set AS_IMQ_LIB=D:¥WebOTX¥wojms¥lib
set AS_IMQ_BIN=D:¥WebOTX¥wojms¥bin
set AS_CONFIG=D:¥WebOTX¥config
set AS_INSTALL=D:¥WebOTX
set AS_JAVA=C:¥jdk1.5.0_05
set AS_ACC_CONFIG=D:¥WebOTX¥lib¥appclient¥nec-acc.xml
set AS_ICU_LIB=D:¥WebOTX¥lib
set AS_DEF_DOMAINS_PATH=Z:¥domains
set AS_OSPI_LIB=D:¥WebOTX¥ObjectBroker¥lib
```

← ドメイン移動先に変更

13. 対象: Web Std-J Std Ent TP システムへのドメイン情報の登録

手順 1 のように、ant を使って正規の方法でドメイン生成や削除を行った場合は、ant のスクリプトの中から暗黙に TP システムの動作に必要な情報が作成、削除されます。しかし、ここでは手動でドメイン情報をカスタマイズしているため、TP システムへのドメイン情報も tpmregister コマンドを実行して明示的に定義する必要があります。

コマンド・プロンプト等で「WebOTX インストールディレクトリ¥bin」に移動し、以下のようにコマンドを実行してください。

```
tpmregister -u MySystem (N1 のみで実行してください)
tpmregister -r MySystem Z:¥domains¥domain1 "WebOTX インストールディレクトリ"
```

14. 対象: Web Std-J Std Ent Web サーバへのドメイン情報の修正

WebOTX Web サーバをインストールしている場合、次の Web サーバ(Apache)用の定義情報ファイルに設定されているドメインフォルダ情報をすべて変更してください。

- Z:¥domains¥domain1¥bin¥apachectl.bat

[apachectl.bat]

```
...
SET INSTDIR=D:¥WebOTX¥WebServer
SET SERVERROOT=Z:¥domains¥domain1 ← ドメインの移動先に変更
...
```

- Z:¥domains¥domain1¥config¥WebServer¥httpd.conf
- Z:¥domains¥domain1¥config¥WebServer¥ssl.conf

[httpd.conf]

```
...
ServerRoot "Z:/domains/domain1" ← ドメインの移動先に変更
...
DocumentRoot "Z:/domains/domain1/docroot" ← ドメインの移動先に変更
...
<Directory "Z:/domains/domain1/docroot"> ← ドメインの移動先に変更
...
Include "Z:/domains/domain1/config/WebServer/ssl.conf" ← ドメインの移動先に変更
...
```

15. 対象: Web Std-J Std Ent Web サーバのサービス登録

Web サーバのサービスを登録します。次のコマンドを実行してください。

```
Z:¥domains¥domain1¥bin¥apachectl.bat INSTALL
```

16. 対象: Web Std-J Std Ent N1 ノードでのドメイン起動

N1 ノードにおいて、WebOTX Agent Service サービスを開始してください。この時にドメインが起動します。

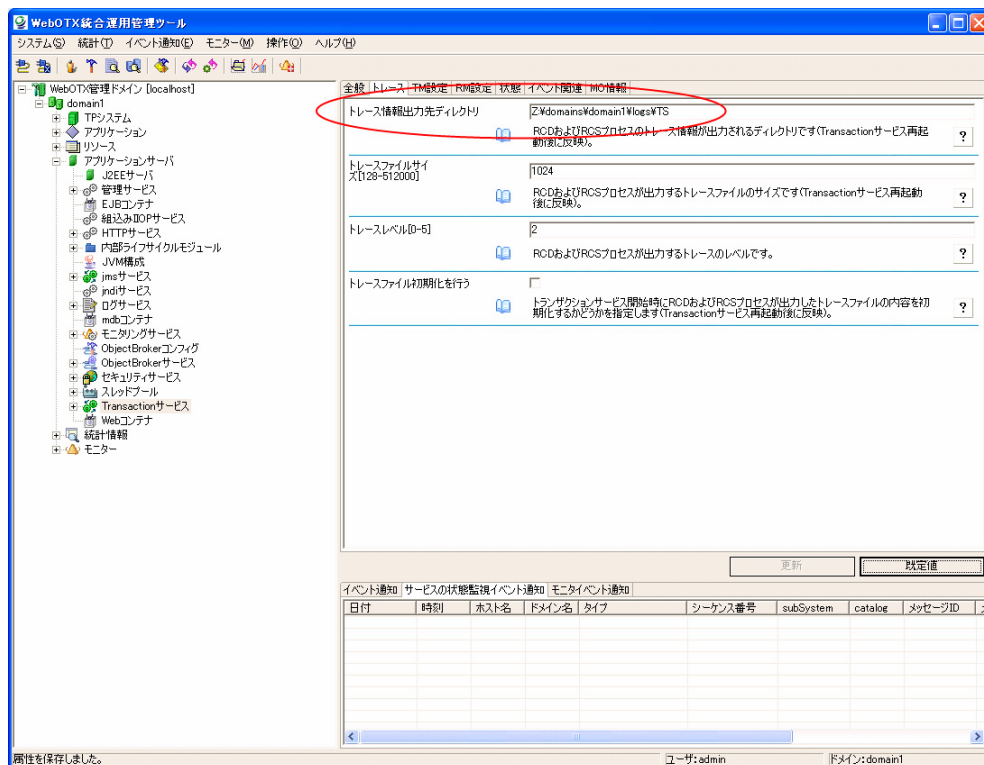
サービスは、[コントロールパネル]→[管理ツール]→[サービス]より起動してください。または、以下のコマンドを実行してください。

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

17. 対象: Web Std-J Std Ent トランザクション・サービスへのトレース情報出力先の変更

運用管理ツールでドメインに接続後、[アプリケーションサーバ](server)→[Transaction サービス](transactionService)→[トレース]タブ内のトレース情報出力ディレクトリの値がドメイン移動先に変更されていることを確認してください。変更していない場合は変更してください。

(例) Z:¥domains¥domain1¥logs¥TS



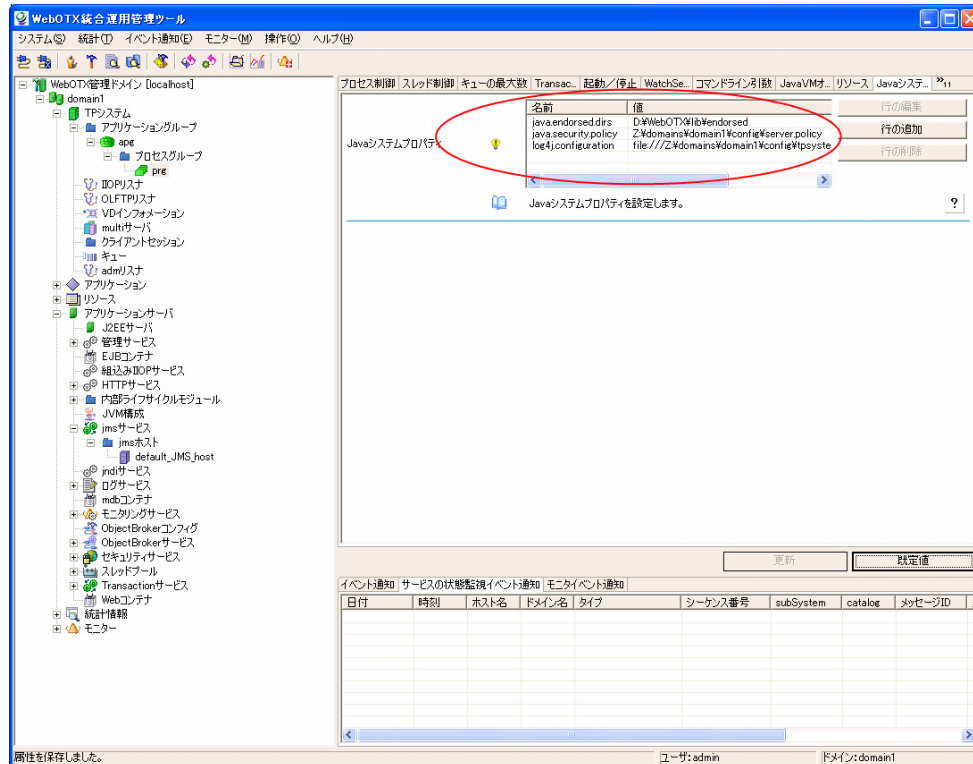
18. 対象: Web Std-J Std Ent プロセスグループへの Java システムプロパティ値の修正

運用管理ツールで[TP システム](tpsystem)→[アプリケーショングループ](applicationGroups)→[作成したアプリケーショングループの名前]→[プロセスグループ](processGroups)→[作成したプロセスグループの名前]→[トレース設

定]タブのトレースファイル名と、[Java システムプロパティ]タブの java.security.policy と log4j.configuration がドメイン移動先に変更されていることを確認してください。変更されていない場合は、移動先ディレクトリに変更してください。

キー	値
java.security.policy	Z:¥domains¥domain1¥config¥server.policy
log4j.configuration	file:///Z:¥domains¥domain1¥config¥tps¥system¥logconf¥apg-prg¥log4j.xml

(注) apg-prg は作成したアプリケーショングループ名、プロセスグループ名をハイフンでつないだ名前です。適宜、作成した名前に読み替えてください。



19. 対象: Web Std-J Std Ent 名前サーバへの永続化情報の登録

配備するコンポーネントに対して名前サーバへ IOR を永続化として登録する必要がある場合は、運用管理ツールから[アプリケーション]→[CORBA アプリケーション]の先のツリーにあるインタフェース・ノード画面から名前サーバへの登録を行ってください。

20. 対象: Web Std-J Std Ent CLUSTERPRO Manager へのスクリプト編集

CLUSTERPRO Manager のスクリプトを編集してください。フェイルオーバーグループ WebOTX1 が起動している場合は停止し、スクリプトの編集を行います。

CLUSTERPRO Manager において、[グループのプロパティ]を選択し、[スクリプト]ページを開きます。2.2.3 節の片方向スタンバイ型スクリプトサンプルのように起動スクリプトと、終了スクリプトを[編集]してください。

片方向スタンバイ型の設定は以上です。

2.2.2 双方向スタンバイ型

本手順は以下の環境を前提にしています。

N1 ノード		
	フェイルオーバーグループ名	WebOTX1 (1 つめの WebOTX ドメイン用)
	仮想 IP アドレス	10.34.69.100
	仮想ホスト名	WebOTX1
	切り替えパーティション	Y:

N2 ノード		
	フェイルオーバーグループ名	WebOTX2 (2 つめの WebOTX ドメイン用)
	仮想 IP アドレス	10.34.69.150
	仮想ホスト名	WebOTX2
	切り替えパーティション	Z:

上記のリソースは実際の環境に読み替えてください。

I. CLUSTERPRO Manager の設定

はじめに、フェイルオーバーグループの設定を行います。作業はすべて、CLUSTERPRO Manager より行います。

1. CLUSTERPRO Manager でクラスタ名を選択して、[グループの追加]を行います。

ツールバーより、[クラスタ]→[グループ追加]を選択し、グループの名前の入力(WebOTX1)を行います。その後、プロパティ画面が表示されるので、全般ページに以下の設定を行います。

[リソースの設定]ボタンを押した後、リソースの設定画面が表示されます。

[切り替ディスク]→[追加]を選択し、グループに割り当てる共有ディスク「Y:」を選択します。

[IP アドレス]→[フローティング IP 追加]を選択し、仮想アドレス(10.34.69.100)とネットマスクを設定します。

[仮想コンピュータ名]→[追加]で、WebOTX1 を追加します。

設定終了後[OK]を押すと、フェイルオーバーポリシーの設定ダイアログが表示されるので、フェイルオーバーさせたい順番に[追加]を行ってください。今回は、WebOTX1 はN1→N2 の順番で設定してください。

WebOTX2 も同様に、WebOTX1 と設定が重ならないように設定してください。(切替パーティションが Z:、IP アドレスが 10.34.69.150、フェイルオーバーポリシーは N2→N1)

以上で、ノードの下にフェイルオーバーグループの作成が完了します。

II. WebOTX 運用管理ツールでの設定

(注意)

以下の記述では、[]で囲まれた項目名は WebOTX V6.3 の場合で、()で囲まれた項目名は V6.2 での表示となります。運用管理ツールのイメージ図は V6.3 のものです。

1. 対象: Web Std-J Std Ent WebOTX インストール・デフォルトのドメイン削除

各ノードで WebOTX のインストールを行い、ドメインの削除を行います。以下の手順で行います。

WebOTX のサービス(WebOTX Agent Service)の起動を行います。[コントロールパネル]→[管理ツール]→[サービス]から、またはコマンドから起動してください。

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

Standard Edition 以上では、WebOTX TPBASEadm サービスもありますが、これは WebOTX Agent Service を起動すると同時に起動されます。

デフォルトのドメインの停止を行います。

```
otxadmin stop-domain domain1
```

ドメインの削除を行います。

```
ant -f setup.xml -Ddomain.name=domain1 delete-domain
```

2. 対象: Web Std-J Std Ent クラスタ用ドメインの新規作成

クラスタ用のドメイン、webotx1(システム名: WebOTX1)、webotx2(システム名: WebOTX2)をN1ノード上で作成し、ドメインを起動します。ただし、利用するポートと SystemID、systemNameW の値はドメイン webotx1 と webotx2 で必ず異なる値を設定してください。ドメイン作成用プロパティ・ファイルは 2.2.5 節を参考にしてください。

```
ant -f setup.xml -Dfile=webotx1.properties create-domain
ant -f setup.xml -Dfile=webotx2.properties create-domain
otxadmin start-domain webotx1
otxadmin start-domain webotx2
```

プロパティ・ファイルの名前、ドメイン名等は適宜読み替えてください。

3. 対象: Web Std-J Std Ent アプリケーションの配備

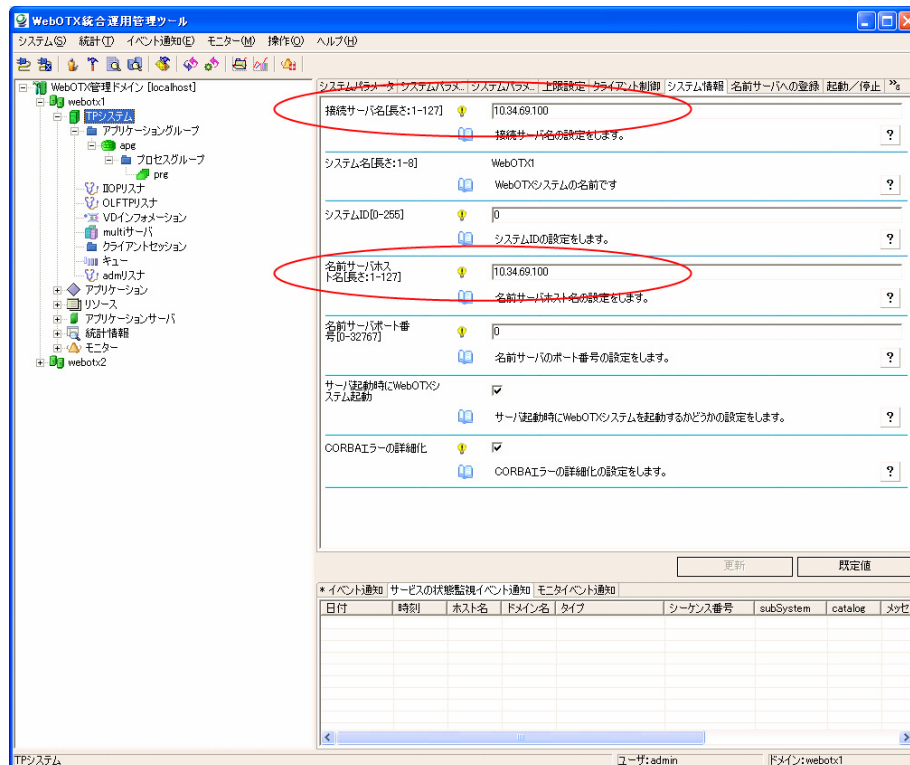
Web や EJB、CORBA などのアプリケーションを配備します。Standard / Enterprise Edition の場合は、アプリケーショングループの作成、プロセスグループの作成等、環境設定を行った後に配備を行います。

以下 4～13 は N1 ノードのドメイン webotx1、webotx2 とともに同じ作業を行います。

4. 対象: Web Std-J Std Ent TP システムへの仮想 IP アドレス設定

運用管理ツールの左側ツリーから[TP システム](tpsystem)を選択し、その右画面の[システム情報]タブを選んだ時の設定画面内から「接続サーバ名」と「名前サーバのホスト名」を仮想 IP アドレスに変更します。

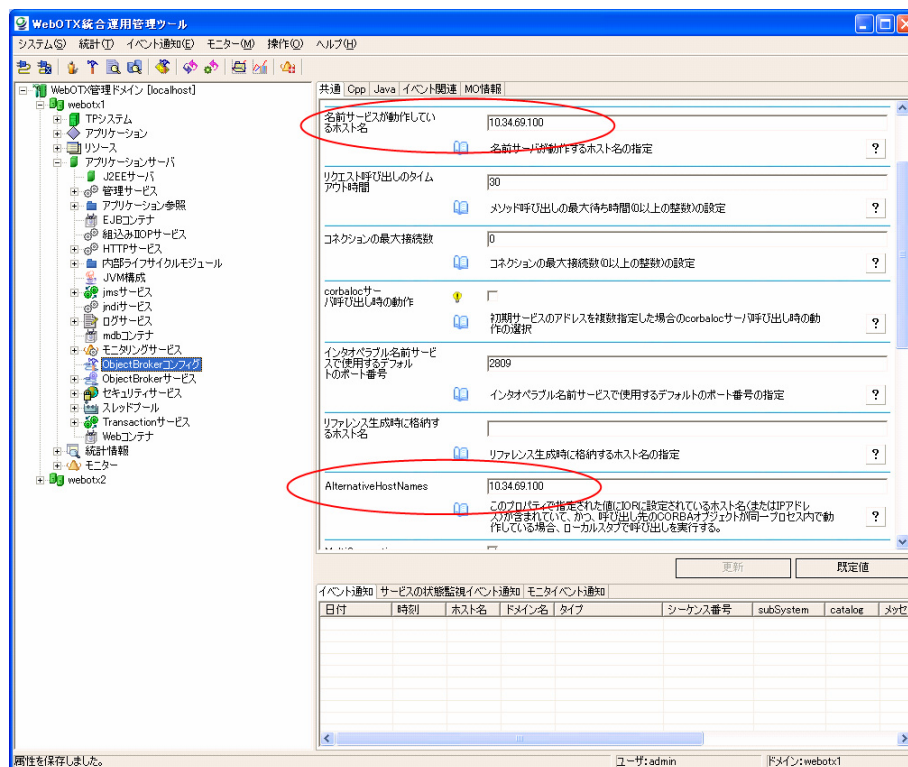
webotx1: 10.34.69.100
webotx2: 10.34.69.150



5. 対象: Web Std-J Std Ent CORBA 通信基盤、Object Broker への仮想 IP アドレス設定

運用管理ツールの[アプリケーションサーバ](server)→[ObjectBroker コンフィグ](ObjectBrokerConfig)→[共通]タブ内の名前サービスが動作しているホスト名と AlternativeHostNames の値を、ドメインを配置する仮想 IP アドレスに変更します。

webotx1: 10.34.69.100
webotx2: 10.34.69.150

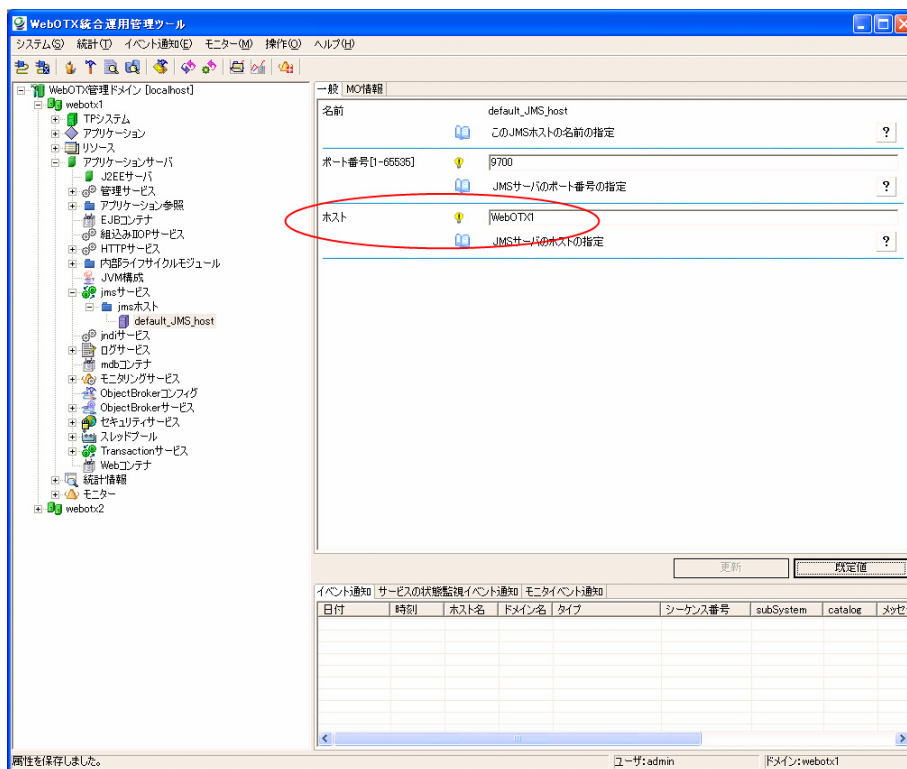


6. 対象: Web Std-J Std Ent JMS (Java Message Service)への仮想 IP アドレス設定

運用管理ツールの[アプリケーションサーバ](server)→[jms サービス](jms-service)→[jms ホスト](jms-host)→[default_JMS_host]→[一般]タブ内のホストを各仮想ホスト名に変更します。ただし、各フェイルオーバーグループを N1 ノードで起動し、仮想 IP アドレスを有効にした状態で仮想ホスト名を設定してください。

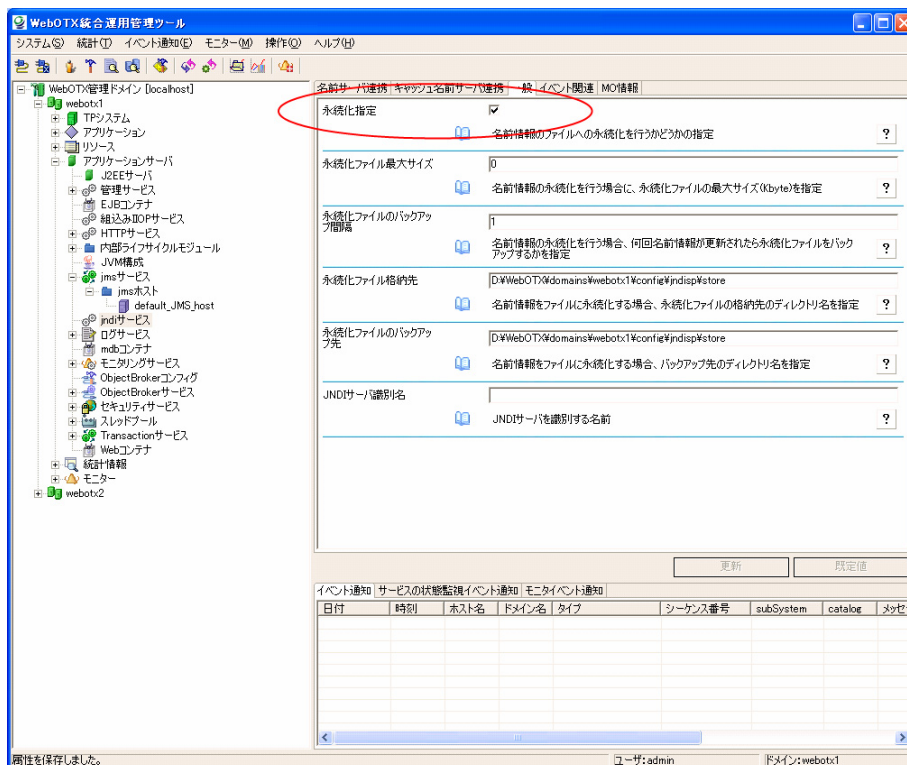
webotx1 では、**仮想ホスト名「WebOTX1」**を指定してください。

webotx2 では、**仮想ホスト名「WebOTX2」**を指定してください。



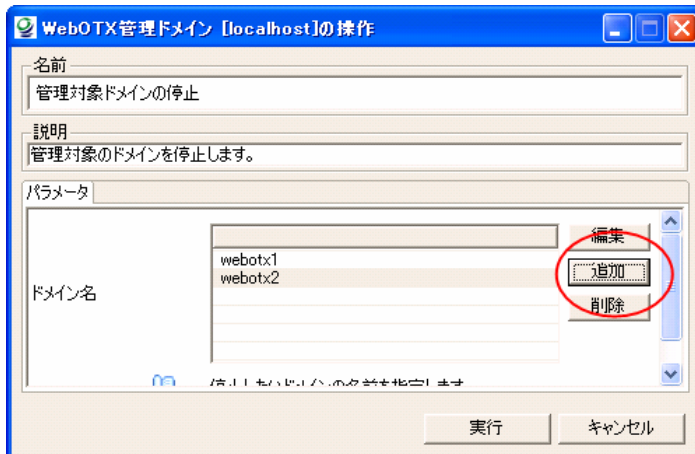
7. 対象: Web Std-J Std Ent JNDI サーバへの永続化によるオブジェクト登録方式の解除

運用管理ツールにおいて、[システム]→[システム設定]を選択し、属性のレベルを全レベルに変更してください。運用管理ツールの[アプリケーションサーバ](server)→[jndi サービス](jndi-service)→[一般]タブ内の永続化をチェックしてください。

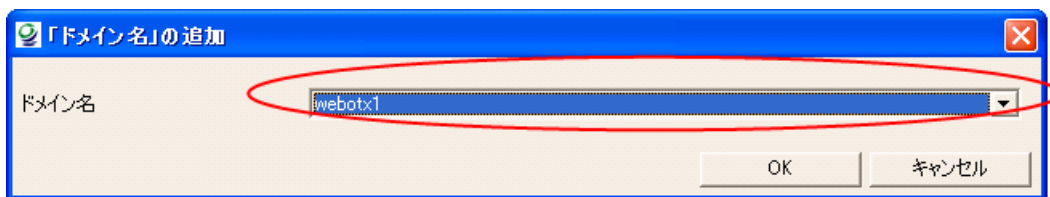


8. 対象: Web Std-J Std Ent 起動中ドメインの停止

WebOTX 運用管理ドメイン(WebOTXAdmin)に運用管理ツールを接続します。WebOTXAdmin を右クリックし、[管理対象ドメインの停止]を選択します。追加ボタンを押し、webotx1、webotx2 を選択し、ドメインの停止を行います。



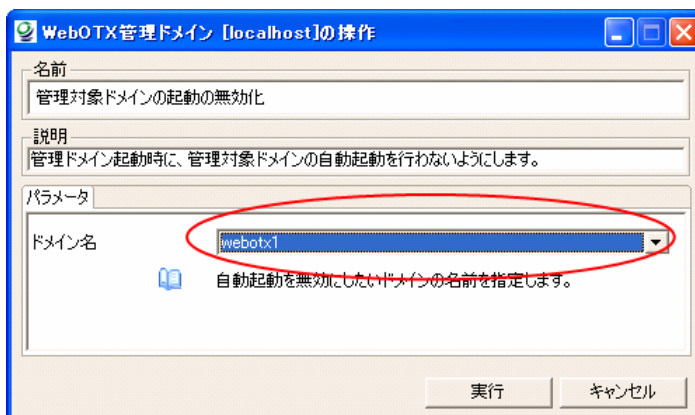
管理対象のドメインの停止



停止するドメインの追加

9. 対象: Web Std-J Std Ent 管理対象ドメインの無効化設定

WebOTXAdmin を右クリックし、[管理対象のドメインの起動の無効化]を選択します。webotx1、webotx2 の両方に対して、無効化を行います。



管理対象ドメインの起動の無効化

10. 対象: Web Std-J Std Ent ドメインの停止

各ノードにおいて、WebOTX Agent Service サービスを停止してください。[コントロールパネル]→[管理ツール]→[サービス]から、またはコマンドから停止してください。

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

11. 対象: Web Std-J Std Ent Object Broker の名前サーバ永続情報の削除

「domains¥webotx1 (webotx2) ¥config¥ObjectBroker」フォルダ内にある「namesv.ndf」ファイルを削除してください。

12. 対象: Web Std-J Std Ent トランザクション・サービスへの仮想 IP アドレス設定

「domains¥webotx1 (webotx2) ¥config¥TS¥jta.conf」ファイルをエディタで開き、[JTA]セッションの下に下記を追加します。

```
LogicalHostname="仮想 IP アドレス、または、仮想コンピュータ名"
```

13. 対象: Web Std-J Std Ent ドメイン起動、停止スクリプトの修正

N1 ノードの「domains¥webotx1 (webotx2) ¥bin」配下の「startserv.bat」と「stopserv.bat」ファイルの編集を行います。それぞれのファイルともに、AS_DEF_DOMAINS_PATH にドメインの移動先を追加してください。

(ドメイン webotx1 の例)

```
if exist "%ASENV_CONF_LOCATION%¥asenv.bat" goto gotCmdPath
rem %0 must have been found by DOS using the %PATH% so we assume that asenv.bat
rem will also be found in the %PATH%
call asenv.bat
goto doneSetenv
:gotCmdPath
call "%ASENV_CONF_LOCATION%¥asenv.bat"
:doneSetenv

set AS_DEF_DOMAINS_PATH=Y:¥domains ← ドメインの移動予定先を追加する
set INSTANCE_ROOT=%AS_DEF_DOMAINS_PATH%¥webotx1
```

14. 対象: Web Std-J Std Ent ドメイン情報の共有準備

N2 ノードで domains 配下に、webotx1、webotx2 フォルダを作成します。さらに、webotx1、webotx2 配下に bin フォルダを作成します。

15. ドメイン起動制御ファイルのコピー

N1 ノードの「webotx1¥bin」配下の startserv.bat、stopserv.bat ファイルを 14 で作成した N2 ノードの「webotx1¥bin」にコピーします。同様に N1 ノードの「webotx2¥bin」配下の startserv.bat、stopserv.bat ファイルを N2 ノードの「webotx2¥bin」配下にコピーします。

16. N1 ノードでのフェイルオーバーグループ起動

N1 ノードでフェイルオーバーグループ WebOTX1、WebOTX2 を起動します。N1 の webotx1 配下すべてを共有ディスク Y に、webotx2 配下すべてを Z にコピーします。今回、Y、Z ともに domains フォルダを作成し、domains 以下にコピーするものとします。

```
Y:¥domains¥webotx1
Z:¥domains¥webotx2
```

17. TP システムへのドメイン情報の登録

手順 1 のように、ant を使って正規の方法でドメイン生成や削除を行った場合は、ant のスクリプトの中から暗黙に TP システムの動作に必要な情報が作成、削除されます。しかし、ここでは手動でドメイン情報をカスタマイズしているため、TP システムへのドメイン情報も tpmregister コマンドを実行して明示的に定義する必要があります。

コマンド・プロンプト等で「WebOTX インストールディレクトリ¥bin」に移動し、以下のようにコマンドを実行してください。

```
tpmregister -u WebOTX1 (N1 で行ってください)
tpmregister -u WebOTX2 (N1 で行ってください)
tpmregister -r WebOTX1 Y:¥domains¥webotx1 "WebOTX インストールディレクトリ" (N1、N2 で行ってください)
tpmregister -r WebOTX2 Z:¥domains¥webotx2 "WebOTX インストールディレクトリ" (N1、N2 で行ってください)
```

18. Web サーバへのドメイン情報の修正

WebOTX Web サーバをインストールしている場合、次の Web サーバ(Apache)用の定義情報ファイルに設定されているドメイン情報をすべて変更してください。N1 と N2 ノードで行います。

- Y:¥domains¥webotx1¥bin¥apachectl.bat
- Z:¥domains¥webotx2¥bin¥apachectl.bat

[apachectl.bat](ドメイン webotx1 の例)

```
...
SET SERVERROOT=Y:¥domains¥webotx1          ← ドメインの移動先に変更
...
SET SVCNAME=WebOTX WebServer webotx1        ← ドメイン名を変更する
...
```

- Y:¥domains¥webotx1¥config¥WebServer¥httpd.conf
- Y:¥domains¥webotx1¥config¥WebServer¥ssl.conf

[httpd.conf] (ドメイン webotx1 の例)

```
...
ServerRoot "Y:/domains/webotx1"             ← ドメインの移動先に変更
...
DocumentRoot "Y:/domains/webotx1/docroot"    ← ドメインの移動先に変更
...
<Directory "Y:/domains/webotx1/docroot">     ← ドメインの移動先に変更
...
Include "Y:/domains/webotx1/config/WebServer/ssl.conf" ← ドメインの移動先に変更
...
```

19. 対象: Web Std-J Std Ent Web サーバのサービス登録

Web サーバのサービスを登録します。次のコマンドを実行してください。N1 と N2 ノードで行います。

```
Y:¥domains¥webotx1¥bin¥apachectl.bat INSTALL
Z:¥domains¥webotx2¥bin¥apachectl.bat INSTALL
```

20. 対象: Web Std-J Std Ent N1 ノードでのドメイン起動

N1 ノードにおいて、WebOTX Agent Service サービスとドメインを起動してください。

サービスは、[コントロールパネル]→[管理ツール]→[サービス]より起動してください。または、以下のコマンドを実行してください。

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

サービスを起動した後、ドメインの起動は以下のコマンドで行ってください。

```
otxadmin start-domain --domaindir "Y:¥domains" webotx1
otxadmin start-domain --domaindir "Z:¥domains" webotx2
```

(参考) ドメインの停止コマンド


```
otxadmin stop-domain --domainindir "Y:¥domains" webotx1
otxadmin stop-domain --domainindir "Z:¥domains" webotx2
```

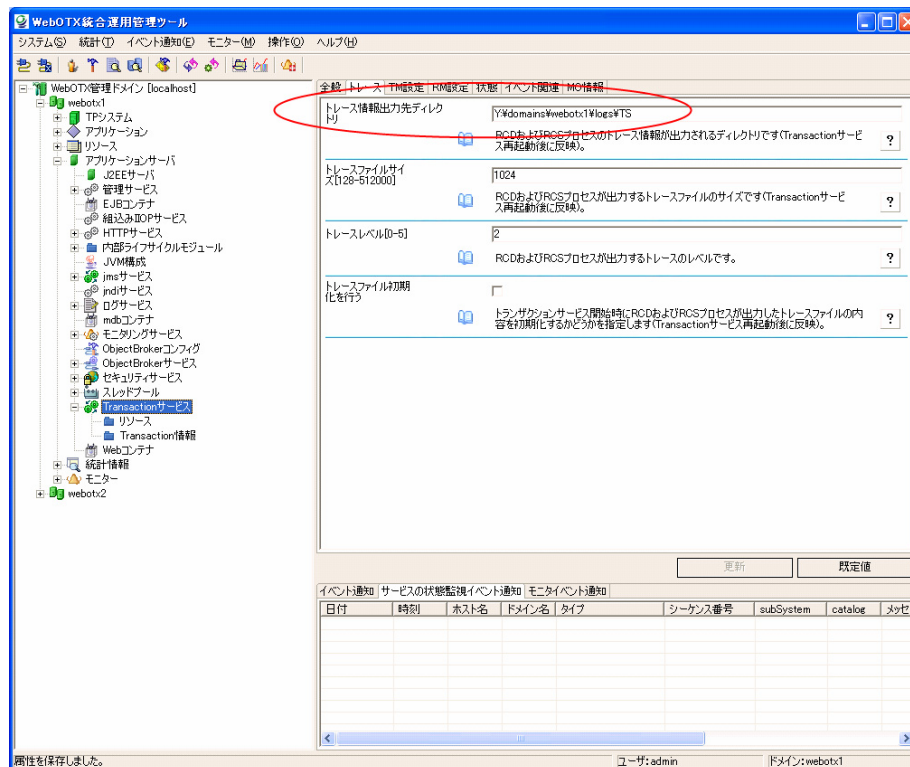
(参考) サービスの停止コマンド

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

21. 対象: Web Std-J Std Ent トランザクション・サービスへのトレース情報出力先の変更

運用管理ツールを N1 ノードの各ドメインに接続後、[アプリケーションサーバ](server)→[Transaction サービス](transactionsservice)→[トレース]タブ内のトレース情報出力ディレクトリの値がドメイン移動先に変更されていることを確認してください。変更していない場合は変更してください。

(例) Y:¥domains¥webotx1¥logs¥TS



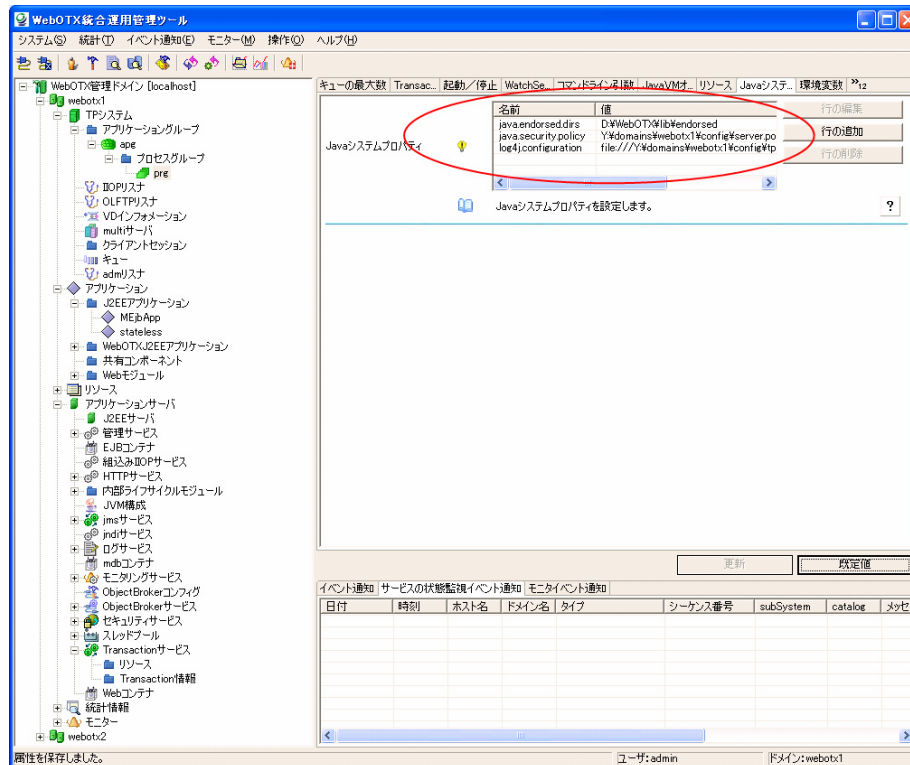
22. 対象: Web Std-J Std Ent プロセスグループへの Java システムプロパティ値の修正

運用管理ツールを N1 ノードの各ドメインに接続します。各ドメインにおいて、運用管理ツールの[TP システム](tpsystem)→[アプリケーショングループ](applicationGroups)→[作成したアプリケーショングループの名前]→[プロセスグループ](processGroups)→[作成したプロセスグループの名前]→[トレース設定]タブのトレースファイル名と、[Java システムプロパティ]タブの java.security.policy と log4j.configuration をドメイン移動先に変更されていることを確認してください。変更されていない場合は、移動先ディレクトリに変更してください。

キー	値
java.security.policy	Y:¥domains¥webotx1¥config¥server.policy

log4j.configuration	file:///Y:¥domains¥webotx1¥config¥tps¥system¥logconf¥ apg-prg¥log4j.xml
---------------------	--

(注) *apg-prg* は作成したアプリケーショングループ名、プロセスグループ名をハイフンでつないだ名前です。適宜、作成した名前に読み替えてください。



23. 対象: Web Std-J Std Ent 名前サーバへの永続化情報の登録

配備するコンポーネントに対して名前サーバへ IOR を永続化として登録する必要がある場合は、運用管理ツールから[アプリケーション]→[CORBA アプリケーション]の先のツリーにあるインタフェース・ノード画面から名前サーバへの登録を行ってください。

24. 対象: Web Std-J Std Ent CLUSTERPRO Manager へのスクリプト編集

CLUSTERPRO Manager のスクリプトを編集してください。フェイルオーバーグループが起動している場合は停止してください。フェイルオーバーグループ、WebOTX1、WebOTX2 の両方で編集を行ってください。

CLUSTERPRO Manager において、[グループのプロパティ]を選択し、[スクリプト]ページを開きます。起動スクリプトと終了スクリプトを[編集]します。2.2.3 節の双方向スタンバイ型スクリプトサンプルのように起動スクリプトと、終了スクリプトを[編集]してください。

双方向スタンバイ型の設定は以上です。

2.2.3 サンプルスクリプト

ここでは、CLUSTERPRO Manager から編集するスクリプトの例を示します。太字の部分を追加してください。

I. 片方向スタンバイ型

起動スクリプト [start.bat]

```
rem *****
rem 業務通常処理
rem *****

rem WebOTX の起動
rem TPBASEadm サービスは、Standard Edition 以上にある
net start "WebOTX TPBASEadm"
net start "WebOTX Agent Service"

rem プライオリティ チェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *****
rem フェイルオーバー後の業務起動ならびに復旧処理
rem *****

rem WebOTX の起動
rem TPBASEadm サービスは、Standard Edition 以上にある
net start "WebOTX TPBASEadm"
net start "WebOTX Agent Service"

rem プライオリティ のチェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER2
```

終了スクリプト [stop.bat]

```
rem *****
rem 業務通常処理
rem *****

rem WebOTX の停止
rem TPBASEadm サービスは、Standard Edition 以上にある
net stop "WebOTX Agent Service"
net stop "WebOTX TPBASEadm"

rem プライオリティ チェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *****
rem フェイルオーバー後の業務起動ならびに復旧処理
rem *****

rem WebOTX の停止
rem TPBASEadm サービスは、Standard Edition 以上にある
net stop "WebOTX Agent Service"
net stop "WebOTX TPBASEadm"

rem プライオリティ のチェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER2
```

II. 双方向スタンバイ型

起動スクリプト [start.bat]

```
rem *****
rem 業務通常処理
rem *****

rem ドメイン名「webotx1」の起動
otxadmin start-domain --domainindir "Y:¥domains" webotx1

rem プライオリティ チェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *****
rem フェイルオーバー後の業務起動ならびに復旧処理
rem *****

rem ドメイン名「webotx1」の起動
otxadmin start-domain --domainindir "Y:¥domains" webotx1

rem プライオリティ のチェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER2
```

終了スクリプト [stop.bat]

```
rem *****
rem 業務通常処理
rem *****

rem ドメイン名「webotx1」の停止
otxadmin stop-domain --domainindir "Y:¥domains" webotx1

rem プライオリティ チェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER1

rem *****
rem フェイルオーバー後の業務起動ならびに復旧処理
rem *****

rem ドメイン名「webotx1」の停止
otxadmin stop-domain --domainindir "Y:¥domains" webotx1

rem プライオリティ のチェック
IF "%ARMS_SERVER%" == "OTHER" GOTO ON_OTHER2
```

2.2.4 ドメイン作成プロパティ・ファイル

この節ではドメイン作成用プロパティ・ファイルのサンプルを示します。ドメインを作成する際に注意することは、以下のとおりです。

- `domain.name` キーでドメインの名前を設定します。一意に設定してください。
- 別に作成するドメインも含めて、使用するポート番号が同じにならないようにしてください。
- `tpsystem.systemNameW` キーでドメインのシステム名を設定します。他のドメインと同じにならないように一意に設定してください。
- `tpsystem.systemID` キーでドメインを識別する値を設定します。他のドメインと同じにならないように一意に設定してください。
- ドメインを管理する ID、パスワードは必要に応じて変更してください。

[webotx1.properties]

```
domain.name=webotx1 ← ドメインの名前を設定します
domain.admin.user=admin ← ドメインを管理するユーザ名を設定します
domain.admin.password=adminadmin ← ドメイン管理ユーザのパスワードを設定します
domain.admin.port=6212
domain.instance.port=80
domain.https.port=443
domain.http.admin.port=4848
domain.http.ajp.port=8009
domain.jms.port=9700

# Lifecycle Services Configs
server.internal-lifecycle-module.ObjectBrokerService.enabled=true
server.internal-lifecycle-module.JNDIService.enabled=true
server.internal-lifecycle-module.J2EEServer.enabled=true
server.internal-lifecycle-module.JMSProvider.enabled=true
server.internal-lifecycle-module.WebContainerService.enabled=true
server.internal-lifecycle-module.TransactionService.enabled=true
server.internal-lifecycle-module.WebServerService.enabled=true
server.internal-lifecycle-module.TPMonitorManagerService.enabled=true

# Embedded IIOP-Service Configs
server.embedded-iiop-service.port=7780

# ObjectBroker Service Configs
server.objectbrokerservice.oadj.Port=9826
server.objectbrokerservice.corbaloc.CorbalocDefaultPort=2809
server.objectbrokerservice.corbaloc.CorbalocServerPort=9827
server.objectbrokerservice.namesv.NameServicePort=2809
server.objectbrokerservice.namesv.NameServiceRoundRobin=true
server.objectbrokerservice.oad.OadPort=9825
server.objectbrokerservice.irsv.InterfaceRepositoryPort=9830
server.objectbrokerservice.cnamesv.CacheNameServicePort=9829

### TPMonitorManagerService Setup Properties (Standard/Enterprise Edition only)
###
tpsystem.systemNameW=WebOTX1 ← ドメインで動作するシステム名を設定します
tpsystem.systemID=0 ← ドメインを識別する ID を設定します
tpsystem.admListener.portNumber=5202
tpsystem.IIOPListener.listenerPortNumber=5151
tpsystem.OLFTPLListener.listenerPortNumber=5251

### Option Package Setup Properties ###

# Transaction Service Configs
server.transactionservice.rcs-port=5964
server.transactionservice.rcs-cpp-port=5965
server.transactionservice.rcs-proxy-port=5966
server.transactionservice.ts-port=5946
```

[webotx2.properties]

```
domain.name=webotx2 ← ドメインの名前を設定します
domain.admin.user=admin ← ドメインを管理するユーザ名を設定します。
domain.admin.password=adminadmin ← ドメイン管理ユーザのパスワードを設定します
domain.admin.port=16212
domain.instance.port=10080
domain.https.port=10443
domain.http.admin.port=14848
domain.http.ajp.port=18009
domain.jms.port=19700

# Lifecycle Services Configs
server.internal-lifecycle-module.ObjectBrokerService.enabled=true
server.internal-lifecycle-module.JNDIService.enabled=true
server.internal-lifecycle-module.J2EEServer.enabled=true
server.internal-lifecycle-module.JMSProvider.enabled=true
server.internal-lifecycle-module.WebContainerService.enabled=true
server.internal-lifecycle-module.TransactionService.enabled=true
server.internal-lifecycle-module.WebServerService.enabled=true
server.internal-lifecycle-module.TPMonitorManagerService.enabled=true

# ObjectBroker Service Configs
server.objectbrokerservice.oadj.Port=19826
server.objectbrokerservice.corbaloc.CorbalocDefaultPort=12809
server.objectbrokerservice.corbaloc.CorbalocServerPort=19827
server.objectbrokerservice.namesv.NameServicePort=12809
server.objectbrokerservice.namesv.NameServiceRoundRobin=true
server.objectbrokerservice.oad.OadPort=19825
server.objectbrokerservice.irsv.InterfaceRepositoryPort=19830
server.objectbrokerservice.cnamesv.CacheNameServicePort=19829

### TPMonitorManagerService Setup Properties (Standard/Enterprise Edition only)
###

tpsystem.systemNameW=WebOTX2 ← ドメインで動作するシステム名を設定します
tpsystem.systemID=10 ← ドメインを識別する ID を設定します
tpsystem.admListener.portNumber=15202
tpsystem.IIOPLListener.listenerPortNumber=15151
tpsystem.OLFTPLListener.listenerPortNumber=15251

### Option Package Setup Properties ###

# Transaction Service Configs
server.transactionservice.rcs-port=15964
server.transactionservice.rcs-cpp-port=15965
server.transactionservice.rcs-proxy-port=15966
server.transactionservice.ts-port=15946
```


3. Linux での環境構築の手順

この章で説明する Linux での環境構築方法は、以下のソフトウェアを利用した場合の例証を行っています。

- WebOTX Standard-J Edition Ver6.31
- WebOTX 運用環境 Ver6.31
- CLUSTERPRO Ver3.1 (CLUSTERPRO CD R5.1)

これ以外の WebOTX バージョンやエディションである場合は、2 章の Windows 版と照らし合わせながら参照してください。

3.1 事前準備

CLUSTERPRO で仮想 IP アドレスの割当や共有ディスクの設定などを施した後に、各ノード(クラスタ環境を構築するマシン)に WebOTX をインストールしてください。WebOTX のバージョンやエディション、および各種設定は各ノードで同じものを利用してください。

3.2 クラスタ環境の構築

3.2.1 片方向スタンバイ型

ここで示す片方向スタンバイ型のクラスタ構築手順では、以下の設定を用います。

フェイルオーバーグループ名		webotx1
CLUSTERPRO の設定	仮想 IP アドレス	10.0.1.10
	仮想ホスト名	aps1
	切替パーティション	/mnt/webotx1
WebOTX の設定	ドメイン名	domain1
	JNDI サーバ識別名	aps1jndi

実際にクラスタ構築を行う場合には、上記の設定を適切に読み替えてください。また環境構築を行うノードを N1 および N2 と表記します。

I. 仮想ホスト名の解決

各ノードで仮想 IP アドレスと仮想ホスト名を解決できるようにする必要があります。この手順書では、`/etc/hosts` ファイルを編集することとしますが、環境によっては DNS 等を利用して名前解決を行うことになります。

以下の例のように、`/etc/hosts` ファイルに仮想 IP アドレスとそれに対応した仮想ホスト名を追加します。

```
[/etc/hosts]
```

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      localhost.localdomain  localhost
127.0.0.1      localhost.domain    localhost

10.0.1.10     aps1
```

II. WebOTX の設定

注意事項

- 運用管理コマンド(otxadmin)を実行する場合は、WebOTX の運用ユーザで行ってください。
- domain1 ドメインのユーザ名は admin、パスワードは adminadmin とします。また、管理ポートは、6212 とします。
- 設定手順は WebOTX のインストールディレクトリで作業していることを前提としています。

1. ノード N1 とノード N2 に WebOTX をインストールします。
2. ノード N2 を管理ドメインだけが存在する状態にします。WebOTX のインストール直後にはデフォルトのドメイン domain1 が存在するため、これを削除します。ドメインの削除は管理ドメインが起動している状態で行う必要があるため、ノード N2 で以下のコマンドを実行し、WebOTX を起動します。

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

次にドメイン domain1 を停止します。

```
# ./bin/otxadmin stop-domain domain1
```

ドメイン domain1 を削除します。

```
# ./lib/ant/bin/ant -f setup.xml -Ddomain.name=domain1 delete-domain
```

3. 手順 4～6 では、ノード N1 上でドメイン domain1 の設定をドメイン domain1 にログインした状態で行います。ここではその準備段階として、ドメイン domain1 の起動と、ドメイン domain1 へのログインを行います。

運用管理コマンドを利用して作業を行う場合は、次のコマンドを実行して domain1 が起動していることを確認してください。起動済みの場合は、「running」と出力されます。

```
# ./bin/otxadmin list-domains
List of domains:
WebOTXAdmin running
domain1 running
```

現在ドメイン domain1 が起動していない(「not running」である)場合は、次のコマンドを実行し、domain1 を起動してください。

```
# ./bin/otxadmin start-domain domain1
Starting Domain domain1, please wait.
Log redirected to /opt/WebOTX/domains/domain1/logs/server.log.
Domain domain1 started.
```

ドメイン domain1 にログインします。

```
# ./bin/otxadmin
otxadmin> login --user admin --password adminadmin --port 6212
login successful.
```

4. Object Broker の設定を行います。

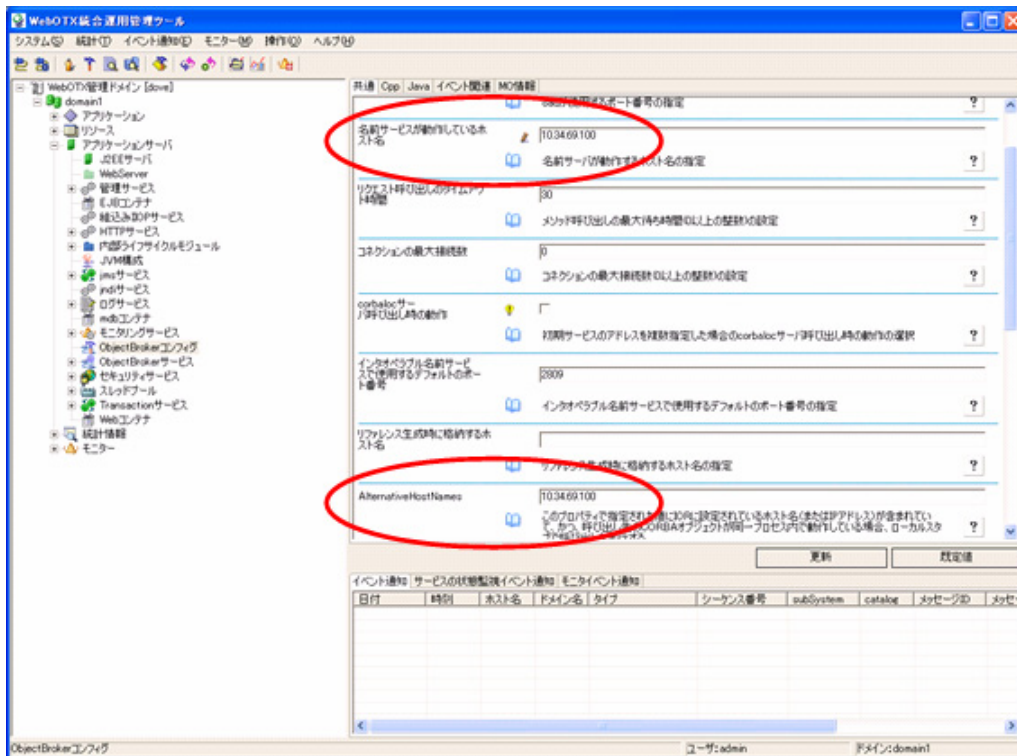
運用管理コマンドで、Object Broker が参照する名前サーバと AlternativeHostNames に仮想 IP アドレス「10.0.1.10」を設定します。以下のコマンドを実行してください。

```
otxadmin> set server.objectbrokerconfig.NameServiceHostName=10.0.1.10
server.objectbrokerconfig.NameServiceHostName = 10.0.1.10
otxadmin> set server.objectbrokerconfig.AlternativeHostNames=10.0.1.10
server.objectbrokerconfig.AlternativeHostNames = 10.0.1.10
```

設定を確認するには、次コマンドを実行します。

```
otxadmin> get server.objectbrokerconfig.NameServiceHostName
server.objectbrokerconfig.NameServiceHostName = 10.0.1.10
otxadmin> get server.objectbrokerconfig.AlternativeHostNames
server.objectbrokerconfig.AlternativeHostNames = 10.0.1.10
```

運用管理ツールから設定を行う場合は、[アプリケーションサーバ]→[ObjectBroker コンフィグ]→[共通]タブを選択し、[名前サービスが動作しているホスト名]と、[AlternativeHostNames]に仮想 IP アドレス「10.0.1.10」を設定してください。



5. JMS の設定を行います。設定に先立ち、フェイルオーバーグループ webotx1 をノード N1 で起動して仮想 IP アドレスを有効にしておく必要があります。

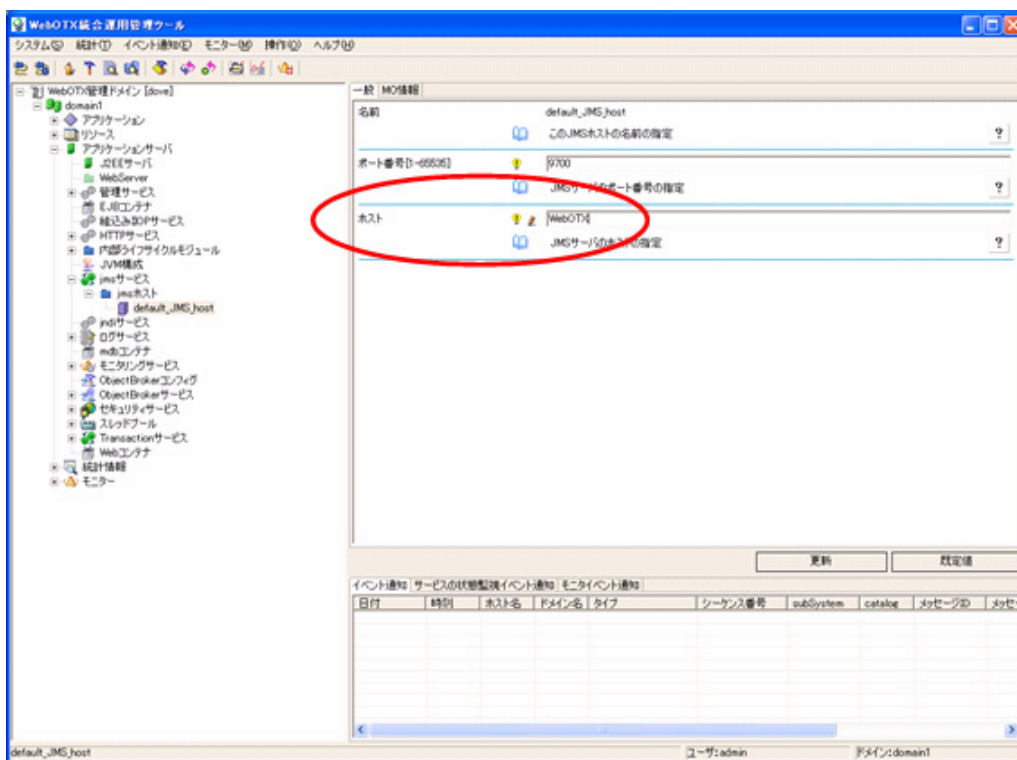
運用管理コマンドで、JMS サービスのホスト名を仮想ホスト名「aps1」に設定します。以下のコマンドを実行してください。

```
otxadmin> set server.jms-service.jms-host.default_JMS_host.host=aps1
host - Please restart domain.
server.jms-service.jms-host.default_JMS_host.host = aps1
```

設定を確認するには次コマンドを実行します。

```
otxadmin> get server.jms-service.jms-host.default_JMS_host.host
server.jms-service.jms-host.default_JMS_host.host = aps1
```

運用管理ツールから設定を行う場合は、[アプリケーションサーバ]→[jms サービス]→[jms ホスト]→[default_JMS_host]→[一般]タブ内の[ホスト]を仮想ホスト名「aps1」に変更します。



6. JNDI サービスの設定を行います。

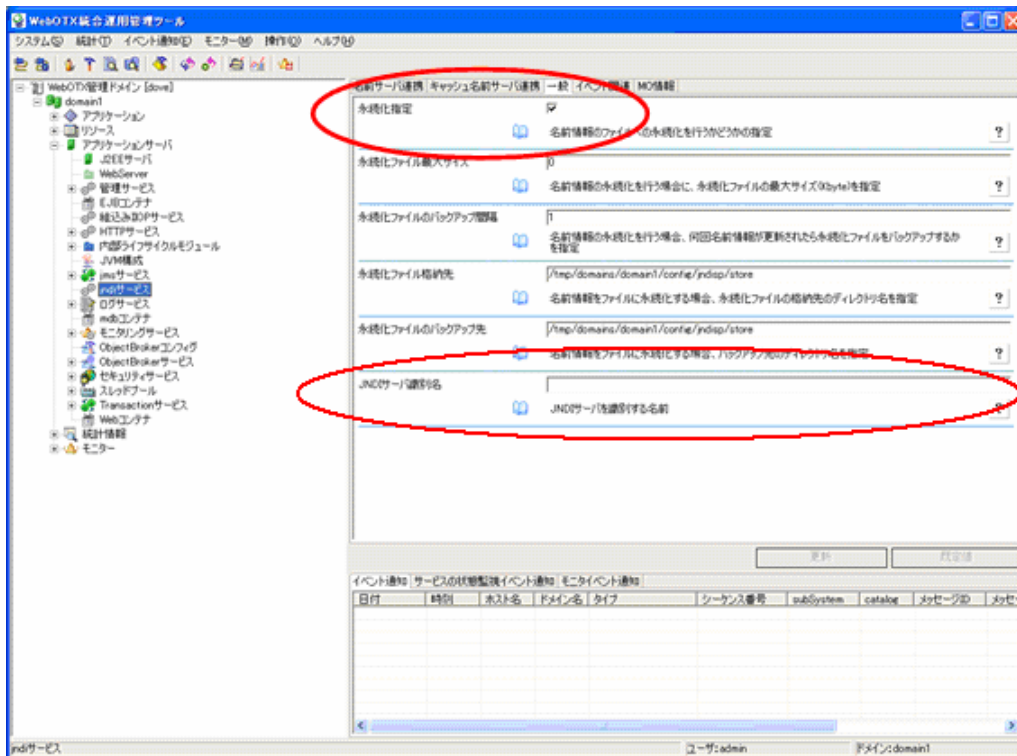
運用管理コマンドで JNDI サーバが管理するオブジェクトを永続化するように設定し、さらに JNDI サーバ識別名を「aps1jndi」に設定します。

```
otxadmin> set server.jndi-service.persistent=true
server.jndi-service.persistent = true
otxadmin> set server.jndi-service.server-name=aps1jndi
server.jndi-service.server-name = aps1jndi
```

設定を確認するには次コマンドを実行します。

```
otxadmin> get --visibility=2 server.jndi-service.persistent
server.jndi-service.persistent = true
otxadmin> get --visibility=2 server.jndi-service.server-name
server.jndi-service.server-name = aps1jndi
```

運用管理ツールから設定を行う場合は、[システム]→[システムの設定]を選択し、[属性の表示レベル]を「詳細レベルの情報表示」に変更してください。その後、運用管理ツールの[アプリケーションサーバ]→[JNDI サービス] → [一般]の[永続化指定]にチェックを入れ、同じタブ内の[JNDI サーバ識別名]を「aps1jndi」に設定してください。

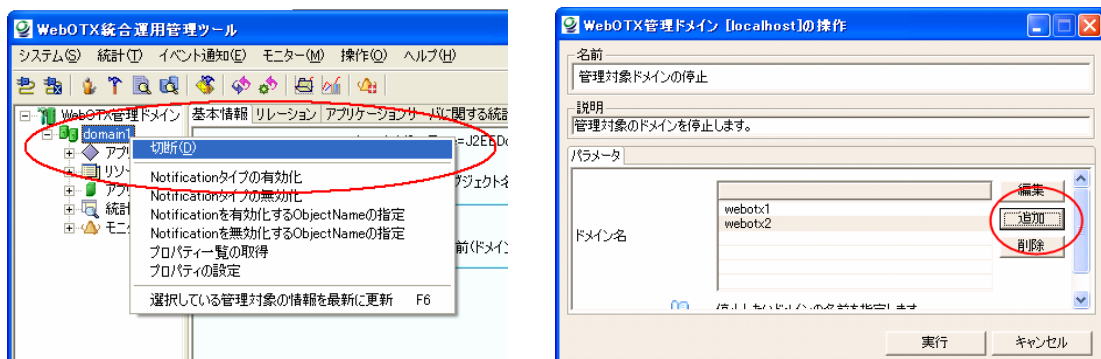


7. 手順8～9では、ドメイン domain1 の設定を、ドメイン domain1 が停止した状態で行います。ここではその準備段階として、ドメイン domain1 からのログアウトと、ドメイン domain1 の停止を行います。

運用管理コマンドで作業をしている場合は、次のコマンドを実行してログアウトとドメイン domain1 の停止を行います。

```
otxadmin> logout
logout successful.
otxadmin> stop-domain domain1
Domain domain1 stopped.
```

運用管理ツールで作業をしている場合は、まずツリーの[domain1]を選択状態にし、[切断]のアイコンをクリックします。切断後、ツリーの[WebOTX 管理ドメイン]を右クリックし、[管理対象ドメインの停止]を選択します。追加ボタンをクリックして domain1 を選択し、停止を行います。



8. 名前サーバのキャッシュをクリアします。「/opt/WebOTX/domains/domain1/config/ObjectBroker」ディレクトリ下にあるファイル「namesv.ndf」を削除してください。

9. トランザクション・サービスの論理ホスト名に仮想ホスト名「aps1」を設定します。

「/opt/WebOTX/domains/domain1/config/TS/jta.conf」ファイルをエディタで開き、[JTS]セッションの下に次の行を追加してください。

```
LogicalHostname="aps1"
```

10. 手順 11～14 でノード N1 とノード N2 の環境を同じになるように設定していきます。その準備段階として、ノード N1 とノード N2 の両方で WebOTX 管理ドメインとドメイン domain1 が全て停止していることを確認してください。停止していないものがある場合は停止してください。

11. OS 起動時の起動スクリプトを無効にします。

ここでは単純に起動スクリプト(正確には起動スクリプトへのシンボリック・リンク)をリネームし、起動対象から外れるようにします。ノード N1 およびノード N2 の両方で、以下のコマンドを実行します。

```
# mv /etc/rc0.d/K30WOAgentSvc /etc/rc0.d/k30WOAgentSvc
# mv /etc/rc0.d/K50ObjectSpinner /etc/rc0.d/k50ObjectSpinner
# mv /etc/rc1.d/K30WOAgentSvc /etc/rc1.d/k30WOAgentSvc
# mv /etc/rc1.d/K50ObjectSpinner /etc/rc1.d/k50ObjectSpinner
# mv /etc/rc2.d/S81WOAgentSvc /etc/rc2.d/s81WOAgentSvc
# mv /etc/rc2.d/S75ObjectSpinner /etc/rc2.d/s75ObjectSpinner
# mv /etc/rc3.d/S81WOAgentSvc /etc/rc3.d/s81WOAgentSvc
# mv /etc/rc3.d/S75ObjectSpinner /etc/rc3.d/s75ObjectSpinner
# mv /etc/rc4.d/S75ObjectSpinner /etc/rc4.d/s75ObjectSpinner
# mv /etc/rc5.d/S81WOAgentSvc /etc/rc5.d/s81WOAgentSvc
# mv /etc/rc5.d/S75ObjectSpinner /etc/rc5.d/s75ObjectSpinner
# mv /etc/rc6.d/K30WOAgentSvc /etc/rc6.d/k30WOAgentSvc
# mv /etc/rc6.d/K50ObjectSpinner /etc/rc6.d/k50ObjectSpinner
```

12. WebOTX 管理ドメインとドメイン domain1 を共有ディスクに移動します。共有ディスクを手動でマウントするか、フェイルオーバーグループ webotx1 を起動するかして、共有ディスクをノード N1 の「/mnt/webotx1」にマウントした状態にします。

ノード N1 の「/opt/WebOTX/domains」ディレクトリを「/mnt/webotx1」にコピーします。コピー先のディレクトリは WebOTX の運用ユーザで読み書きができる必要があります。コマンドは次のとおりです。

```
# cp -pR /opt/WebOTX/domains/ /mnt/webotx1
```

次に、ノード N1 およびノード N2 の「/opt/WebOTX/config/asenv.conf」ファイルを以下のように編集します。

[asenv.conf]


```

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/j2se"
AS_ACC_CONFIG="/opt/WebOTX/lib/appclient/nec-acc.xml"
AS_ICU_LIB="/opt/WebOTX/lib"
AS_DEF_DOMAINS_PATH="/mnt/webotx1/domains"
AS_OSPI_LIB="/opt/ObjectSpinner/lib"

```

← ドメイン移動先に変更

13. WebOTX Web サーバをインストールしている場合、共有ディスク上に格納されている次の Web サーバ (Apache)用の各定義情報ファイルについて、設定されているドメインのディレクトリ情報を全て変更してください。なお、本ファイルの編集は root ユーザで行います。

- /mnt/webotx1/domains/domain1/bin/apachectl
- /mnt/webotx1/domains/domain1/config/WebServer/httpd.conf

SSL 通信用パッケージ(mod_ssl パッケージ)をインストールしている場合、次のファイルも編集対象になります。

- /mnt/webotx1/domains/domain1/config/WebServer/ssl.conf

下記に示す例は、domain1ドメインのものです。

[apachectl]

```

...
SERVERROOT=/mnt/webotx1/domains/domain1
CONFFILE=/mnt/webotx1/domains/domain1/config/WebServer/httpd.conf
...
PIDFILE=/mnt/webotx1/domains/domain1/logs/WebServer/httpd.pid
...

```

[httpd.conf]

```

...
ServerRoot "/mnt/webotx1"
...
PidFile "/mnt/webotx1/logs/WebServer/httpd.pid"
...
DocumentRoot "/mnt/webotx1/docroot"
...
<Directory "/mnt/webotx1/docroot">
...
ErrorLog "/mnt/webotx1/logs/WebServer/error_log"
...
CustomLog "/mnt/webotx1/logs/WebServer/access_log" common
...
include "/mnt/webotx1/config/WebServer/ssl.conf"
...

```

14. トランザクション・サービスのトレースファイルの位置を設定します。

まず、ノードN1でフェイルオーバーグループ webotx1 を起動し、共有ディスクが「/mnt/webotx1」にマウントされた状態にしてください。次にノード N1 で WebOTX 管理ドメインとドメイン domain1 を起動してください。

```

# /etc/init.d/WOAgentSvc start
# ./bin/otxadmin start-domain --domaindir "/mnt/webotx1/domains" domain1

```

起動完了後、domain1 ドメインのトランザクション・サービスのトレースファイルの位置を「/mnt/webotx1/domains/domain1/logs/TS」に設定します。

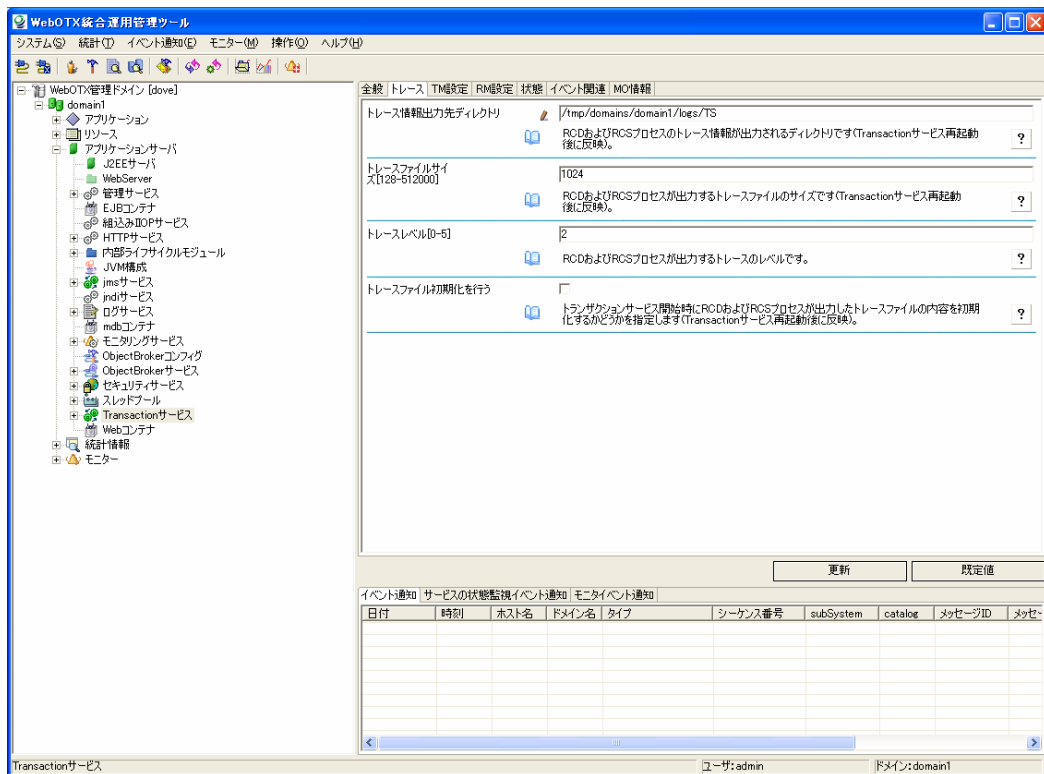
運用管理コマンドで domain1 ドメインの設定を行う場合は、次のコマンドを実行してください。

```

# ./bin/otxadmin
otxadmin> login --user admin --password adminadmin --port 6212
otxadmin> set server.transactionservice.trace-file-path=/mnt/webotx1/domains/d
omain1/logs/TS
server.transactionservice.trace-file-path = /mnt/webotx1/domains/domain1/logs/
TS
otxadmin> logout
otxadmin> exit

```

運用管理ツールで設定する場合は次のように操作します。N1 の domain1 ドメインに接続後、[アプリケーションサーバ]→[Transaction サービス]→[トレース]タブ内の[トレース情報出力ディレクトリ]の値を「/mnt/webotx1/domains/domain1/logs/TS」に設定してください。



15. CLUSTERPRO Manager のスクリプトを編集してください。フェイルオーバーグループが起動している場合は停止してください。「3.2.3 サンプルスクリプト」を参考にフェイルオーバーグループ、webotx1 で編集を行ってください。

3.2.2 双方向スタンバイ型

ここで示す双方向スタンバイ型のクラスタ構築手順では、以下の設定を用います。

フェイルオーバーグループ名		webotx1	webotx2
CLUSTERPRO の設定	仮想 IP アドレス	10.0.1.10	10.0.1.11
	仮想ホスト名	aps1	aps2
	切替パーティション	/mnt/webotx1	/mnt/webotx2
WebOTX の設定	ドメイン名	domain1	domain2
	JNDI サーバ識別名	aps1jndi	aps2jndi

実際にクラスタ構築を行う場合には、上記の設定を適切に読み替えてください。また環境構築を行うノードを N1 および N2 と表記します。

I. 仮想ホスト名の解決

各ノードで仮想 IP アドレスと仮想ホスト名を解決できるようにする必要があります。この手順書では、`/etc/hosts` ファイルを編集することとしますが、環境によっては DNS 等を利用して名前解決を行うことになります。

以下の例のように、`/etc/hosts` ファイルに仮想 IP アドレスとそれに対応した仮想ホスト名を追加します。

```
[/etc/hosts]
```

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1      localhost.localdomain  localhost
127.0.0.1      localhost.domain    localhost

10.0.1.10      aps1
10.0.1.11      aps2
```

II. WebOTX の設定

注意事項

- 運用管理コマンド(otxadmin)を実行する場合は、WebOTX の運用ユーザで行ってください。
- ドメインのユーザ名とパスワードは、domain1、domain2 ドメインともに、ユーザ名: admin、パスワード: adminadmin とします。また、管理ポートは、domain1: 6212、domain2: 16212 とします。
- 設定手順は WebOTX のインストールディレクトリで作業していることを前提としています。

1. ノード N1 にドメイン domain1 とドメイン domain2 を作成します。WebOTX のインストール直後の状態ではドメイン domain1 が既に存在するため、ここではこれを流用します。ドメイン domain2 はデフォルトでは存在しないため、新規に作成します。ドメイン作成用のプロパティ・ファイルを「2.2.4 ドメイン作成プロパティ・ファイル」を参考にして作成してください。ドメインの作成は管理ドメインが起動している状態で行う必要があるため、ノード N1 で以下のコマンドを実行し、WebOTX を起動します。

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

次にドメイン domain2 を作成します。ここでは、ドメイン domain2 作成用のプロパティ・ファイルのファイル名を domain2.properties としています。

```
# ./lib/ant/bin/ant -f setup.xml -Dfile=domain2.properties create-domain
```

2. ノード N2 を管理ドメインだけが存在する状態にします。WebOTX のインストール直後にはデフォルトのドメイン domain1 が存在するため、これを削除します。ドメインの削除は管理ドメインが起動している状態で行う必要があるため、ノード N2 で以下のコマンドを実行し、WebOTX を起動します。

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

次にドメイン domain1 を停止します。

```
# ./bin/otxadmin stop-domain domain1
```

ドメイン domain1 を削除します。

```
# ./lib/ant/bin/ant -f setup.xml -Ddomain.name=domain1 delete-domain
```

3. 手順4～6では、ノードN1上でドメイン domain1 の設定をドメイン domain1 にログインした状態で行います。ここではその準備段階として、ドメイン domain1 の起動と、ドメイン domain1 へのログインを行います。

運用管理コマンドを利用して作業を行う場合は、次のコマンドを実行して domain1 が起動していることを確認してください。起動済みの場合は、「running」と出力されます。

```
# ./bin/otxadmin list-domains
List of domains:
WebOTXAdmin running
domain1 running
domain2 not running
```

現在ドメイン domain1 が起動していない(「not running」である)場合は、次のコマンドを実行し、domain1 を起動してください。

```
# ./bin/otxadmin start-domain domain1
Starting Domain domain1, please wait.
Log redirected to /opt/WebOTX/domains/domain1/logs/server.log.
Domain domain1 started.
```

ドメイン domain1 にログインします。

```
# ./bin/otxadmin
otxadmin> login --user admin --password adminadmin --port 6212
login successful.
```

4. Object Broker の設定を行います。

運用管理コマンドで、Object Broker が参照する名前サーバと AlternativeHostNames に仮想 IP アドレス「10.0.1.10」を設定します。以下のコマンドを実行してください。

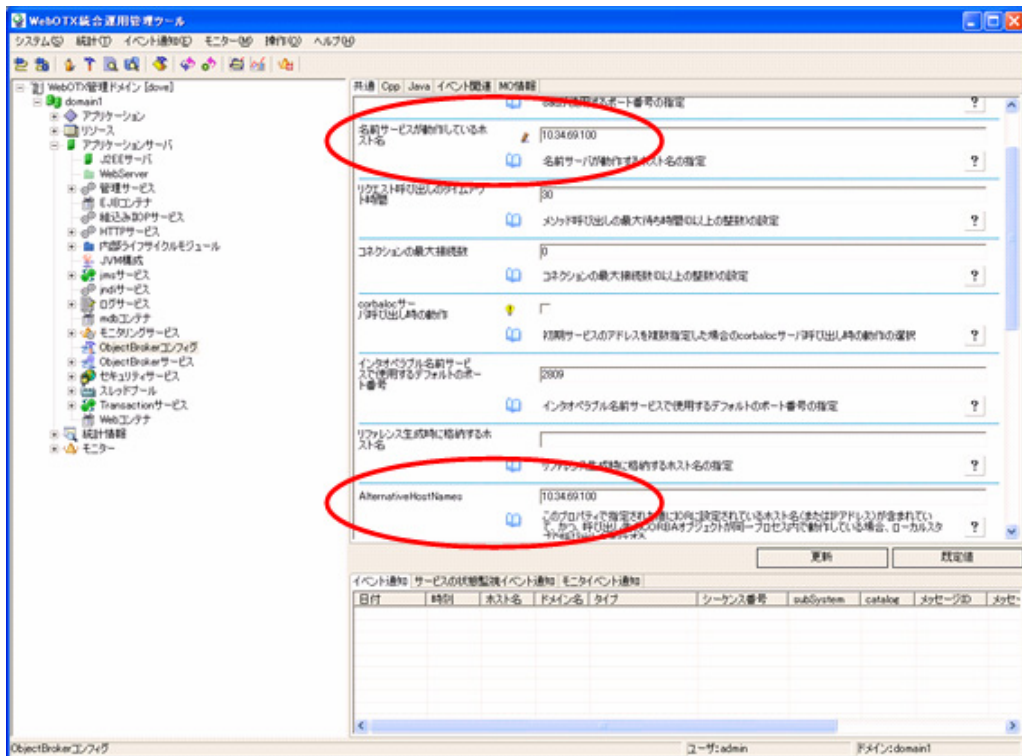
```
otxadmin> set server.objectbrokerconfig.NameServiceHostName=10.0.1.10
server.objectbrokerconfig.NameServiceHostName = 10.0.1.10
otxadmin> set server.objectbrokerconfig.AlternativeHostNames=10.0.1.10
server.objectbrokerconfig.AlternativeHostNames = 10.0.1.10
```

設定を確認するには、次コマンドを実行します。

```
otxadmin> get server.objectbrokerconfig.NameServiceHostName
server.objectbrokerconfig.NameServiceHostName = 10.0.1.10
otxadmin> get server.objectbrokerconfig.AlternativeHostNames
server.objectbrokerconfig.AlternativeHostNames = 10.0.1.10
```

運用管理ツールから設定を行う場合は、[アプリケーションサーバ]→[ObjectBroker コンフィグ]→[共通]タブを選択し、

[名前サービスが動作しているホスト名]と、[AlternativeHostNames]に仮想 IP アドレス「10.0.1.10」を設定してください。



5. JMS の設定を行います。設定に先立ち、フェイルオーバーグループ webotx1 をノード N1 で起動して仮想 IP アドレスを有効にしておく必要があります。

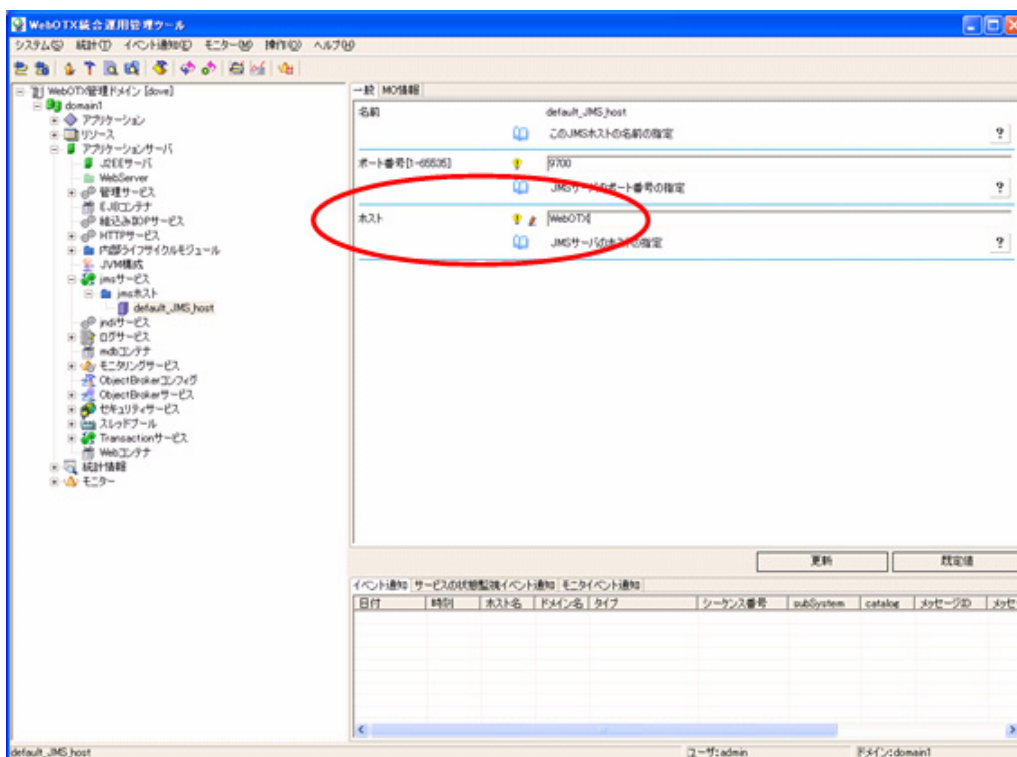
運用管理コマンドで、JMS サービスのホスト名を仮想ホスト名「aps1」に設定します。以下のコマンドを実行してください。

```
otxadmin> set server.jms-service.jms-host.default_JMS_host.host=aps1
host - Please restart domain.
server.jms-service.jms-host.default_JMS_host.host = aps1
```

設定を確認するには次コマンドを実行します。

```
otxadmin> get server.jms-service.jms-host.default_JMS_host.host
server.jms-service.jms-host.default_JMS_host.host = aps1
```

運用管理ツールから設定を行う場合は、[アプリケーションサーバ]→[jms サービス]→[jms ホスト]→[default_JMS_host]→[一般]タブ内の[ホスト]を仮想ホスト名「aps1」に変更します。



6. JNDI サービスの設定を行います。

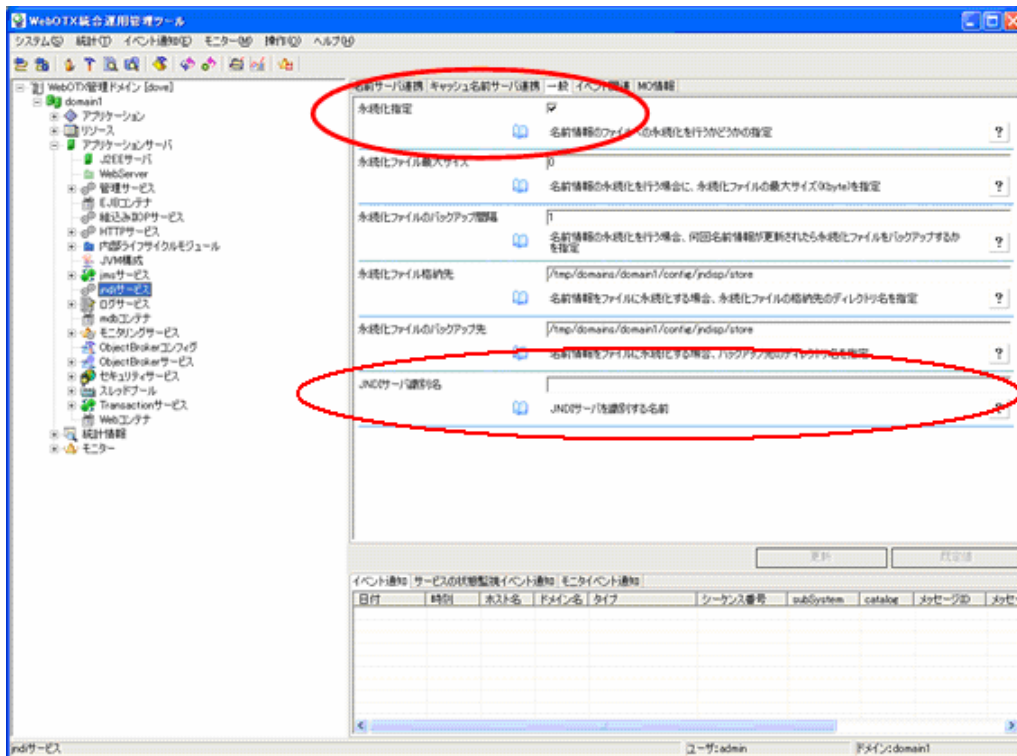
運用管理コマンドで JNDI サーバが管理するオブジェクトを永続化するように設定し、さらに JNDI サーバ識別名を「aps1jndi」に設定します。

```
otxadmin> set server.jndi-service.persistent=true
server.jndi-service.persistent = true
otxadmin> set server.jndi-service.server-name=aps1jndi
server.jndi-service.server-name = aps1jndi
```

設定を確認するには次コマンドを実行します。

```
otxadmin> get --visibility=2 server.jndi-service.persistent
server.jndi-service.persistent = true
otxadmin> get --visibility=2 server.jndi-service.server-name
server.jndi-service.server-name = aps1jndi
```

運用管理ツールから設定を行う場合は、[システム]→[システムの設定]を選択し、[属性の表示レベル]を「詳細レベルの情報表示」に変更してください。その後、運用管理ツールの[アプリケーションサーバ]→[JNDI サービス] → [一般]の[永続化指定]にチェックを入れ、同じタブ内の[JNDI サーバ識別名]を「aps1jndi」に設定してください。

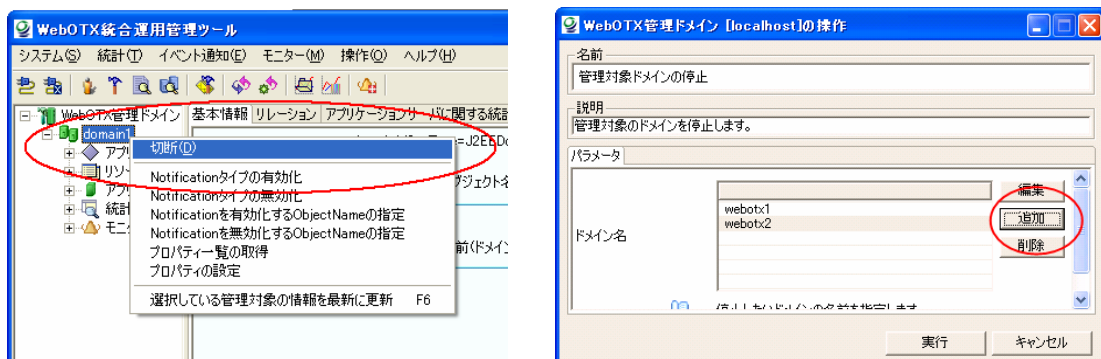


7. 手順 8～11 では、ドメイン domain1 の設定を、ドメイン domain1 が停止した状態で行います。ここではその準備段階として、ドメイン domain1 からのログアウトと、ドメイン domain1 の停止を行います。

運用管理コマンドで作業をしている場合は、次のコマンドを実行してログアウトとドメイン domain1 の停止を行います。

```
otxadmin> logout
logout successful.
otxadmin> stop-domain domain1
Domain domain1 stopped.
```

運用管理ツールで作業をしている場合は、まずツリーの[domain1]を選択状態にし、[切断]のアイコンをクリックします。切断後、ツリーの[WebOTX 管理ドメイン]を右クリックし、[管理対象ドメインの停止]を選択します。追加ボタンをクリックして domain1 を選択し、停止を行います。



8. 名前サーバのキャッシュをクリアします。「/opt/WebOTX/domains/domain1/config/ObjectBroker」ディレクトリ下にあるファイル「namesv.ndf」を削除してください。

9. トランザクション・サービスの論理ホスト名に仮想ホスト名「aps1」を設定します。

「/opt/WebOTX/domains/domain1/config/TS/jta.conf」ファイルをエディタで開き、[JTS]セッションの下に次の行を追加してください。

```
LogicalHostname="aps1"
```

10. domain1 ドメインの起動用のスクリプトを作成します。「/opt/WebOTX/domains/domain1/bin」ディレクトリ下にあるスクリプト「startserv」と「stopserv」ファイルを編集します。両スクリプトとも、環境変数「INSTANCE_ROOT」を設定している行の直前で、環境変数「AS_DEF_DOMAINS_PATH」を再設定します。設定する値は共有ディスク上のドメインの移動予定の場所です。以下は startserv のスクリプトの例です。

```
# echo "PRGDIR => $PRGDIR"
if [ -r "$ASENV_CONF_LOCATION/asenv.conf" ]; then
    . "$ASENV_CONF_LOCATION/asenv.conf"
fi

AS_DEF_DOMAINS_PATH=/mnt/webotx1/domains
INSTANCE_ROOT="$AS_DEF_DOMAINS_PATH/domain1"
```

11. WebOTX の起動時に domain1 が自動起動しないように設定します。

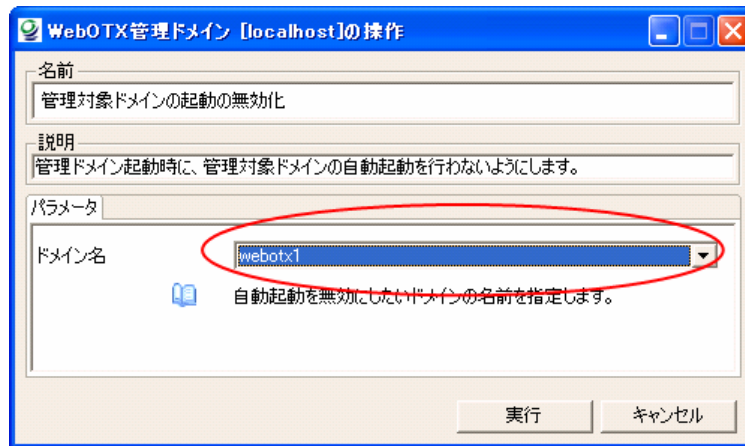
運用管理ツールを使用しないで設定するには、WebOTX 管理ドメインが停止している状態で「/opt/WebOTX/domains/WebOTXAdmin/config/domains-config.xml」ファイルをエディタで編集します。ドメイン domain1 に対応する<managed-domain>要素の属性 enabled の値を「false」に変更します。

[domains-config.xml]

```
<?xml version="1.0" encoding="UTF-8"?>

<!-- Copyright (c) NEC Corporation 2004. All rights reserved. -->
<nec-webotx-managed-domain-configurations xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="domains-config_1_0.xsd">
    <managed-domain name="domain1" enabled="false" port="6212" starting-order="1" status="0"/>
    <managed-domain name="domain2" enabled="true" port="16212" starting-order="1" status="0"/>
</nec-webotx-managed-domain-configurations>
```

運用管理ツールで設定するには、ツリーの[WebOTX 管理ドメイン]を右クリックし、[管理対象のドメインの起動の無効化]を選択します。ドメイン名として domain1 を選択して実行します。



12. 手順 3～11 を、フェイルオーバーグループ webotx2 についても同様に繰り返します。各設定はフェイルオーバーグループ webotx2 のものに読み替えてください。

13. 手順 14～17 でノード N1 とノード N2 の環境を同じになるように設定していきます。その準備段階として、ノード N1 とノード N2 の両方で WebOTX 管理ドメインとドメイン domain1 とドメイン domain2 が全て停止していることを確認してください。停止していないものがある場合は停止してください。

14. 起動スクリプトを配置します。ノード N2 の「/opt/WebOTX/domains」ディレクトリの中に、ディレクトリ「domain1/bin」と「domain2/bin」を作成します。そして、ノード N1 の「/opt/WebOTX/domains/domain1/bin」下の「startserv」と「stopserv」ファイルをノード N2 の「/opt/WebOTX/domains/domain1/bin」ディレクトリにコピーします。

同様にノード N1 の「/opt/WebOTX/domains/domain2/bin」の中の「startserv」と「stopserv」ファイルをノード N2 の「/opt/WebOTX/domains/domain2/bin」にコピーします。

15. 各ドメインを共有ディスクに移動します。共有ディスクを手動でマウントするか、フェイルオーバーグループ webotx1 と webotx2 を起動するかして、共有ディスクをノード N1 の「/mnt/webotx1」と「/mnt/webotx2」ディレクトリにマウントした状態にします。

ノード N1 の「/opt/WebOTX/domains/domain1」ディレクトリを、「/mnt/webotx1/domains」にコピーします。コピー先のディレクトリは WebOTX の運用ユーザで読み書きができる必要があります。コマンドは次のとおりです。

```
# mkdir /mnt/webotx1/domains
# cp -pR /opt/WebOTX/domains/domain1 /mnt/webotx1/domains
```

同様に、ノード N1 の「/opt/WebOTX/domains/domain2」ディレクトリを「/mnt/webotx2/domains」にコピーします。

次にノード N1 の「/opt/WebOTX/domains/WebOTXAdmin/config」下にある「domains-config.xml」ファイルを、ノード N2 の「/opt/WebOTX/domains/WebOTXAdmin/config」にコピーします。

16. WebOTX Web サーバをインストールしている場合、共有ディスク上に格納されている次の Web サーバ (Apache) 用の各定義情報ファイルについて、設定されているドメインのディレクトリ情報を全て変更してください。なお、本ファイルの編集は root ユーザで行います。

- /mnt/webotx1/domains/domain1/bin/apachectl
- /mnt/webotx2/domains/domain2/bin/apachectl
- /mnt/webotx1/domains/domain1/config/WebServer/httpd.conf
- /mnt/webotx2/domains/domain2/config/WebServer/httpd.conf

SSL 通信用パッケージ(mod_ssl パッケージ)をインストールしている場合、次のファイルも編集対象になります。

- /mnt/webotx1/domains/domain1/config/WebServer/ssl.conf
- /mnt/webotx2/domains/domain2/config/WebServer/ssl.conf

下記に示す例は、domain1 ドメインのものです。

[apachectl]

```
...
SERVERROOT=/mnt/webotx1/domains/domain1
CONFFILE=/mnt/webotx1/domains/domain1/config/WebServer/httpd.conf
...
PIDFILE=/mnt/webotx1/domains/domain1/logs/WebServer/httpd.pid
...
```

[httpd.conf]

```
...
ServerRoot "/mnt/webotx1"
...
PidFile "/mnt/webotx1/logs/WebServer/httpd.pid"
...
DocumentRoot "/mnt/webotx1/docroot"
...
<Directory "/mnt/webotx1/docroot">
...
ErrorLog "/mnt/webotx1/logs/WebServer/error_log"
...
CustomLog "/mnt/webotx1/logs/WebServer/access_log" common
...
include "/mnt/webotx1/config/WebServer/ssl.conf"
...
```

17. トランザクション・サービスのトレースファイルの位置を設定します。

まず、ノード N1 でフェイルオーバーグループ webotx1 と webotx2 を起動し、共有ディスクが「/mnt/webotx1」と「/mnt/webotx2」にマウントされた状態にしてください。次にノード N1 で WebOTX 管理ドメインとドメイン domain1 とドメイン domain2 を起動してください。

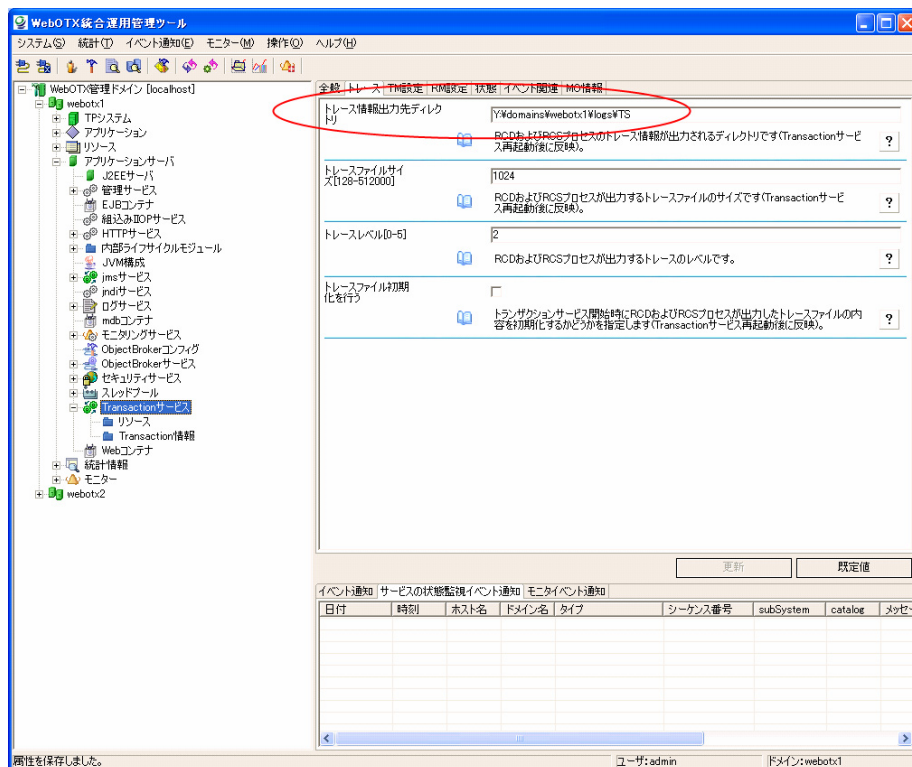
```
# /etc/init.d/WOAgentSvc start
# ./bin/otxadmin start-domain --domainindir "/mnt/webotx1/domains" domain1
# ./bin/otxadmin start-domain --domainindir "/mnt/webotx2/domains" domain2
```

起動完了後、domain1 ドメインのトランザクション・サービスのトレースファイルの位置を「/mnt/webotx1/domains/domain1/logs/TS」に、domain2 ドメインのトランザクション・サービスのトレースファイルの位置を「/mnt/webotx2/domains/domain2/logs/TS」に設定します。

運用管理コマンドで domain1 ドメインの設定を行う場合は、次のコマンドを実行してください。domain2 ドメインの設定を行う場合は、適時設定を読み替えてください。

```
# ./bin/otxadmin
otxadmin> login --user admin --password adminadmin --port 6212
otxadmin> set server.transactionservice.trace-file-path=/mnt/webotx1/domains/d
omain1/logs/TS
server.transactionservice.trace-file-path = /mnt/webotx1/domains/domain1/logs/
TS
otxadmin> logout
otxadmin> exit
```

運用管理ツールで設定する場合は次のように操作します。N1 の domain1 ドメインに接続後、[アプリケーションサーバ]→[Transaction サービス]→[トレース]タブ内の[トレース情報出力ディレクトリ]の値を「/mnt/webotx1/domains/domain1/logs/TS」に設定してください。domain2 ドメインについては設定を適時読み替え、同様に設定してください。



18. CLUSTERPRO Manager のスクリプトを編集してください。フェイルオーバーグループが起動している場合は停止してください。「3.2.3 サンプルスクリプト」を参考にフェイルオーバーグループ、webotx1、webotx2 の両方で編集を行ってください。

3.2.3 サンプルスクリプト

I. 片方向スタンバイ

起動スクリプト [start.sh]

```
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
###    WebOTX start
    /etc/init.d/WOAgentSvc start ← WebOTX の起動
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi

###    WebOTX start
    /etc/init.d/WOAgentSvc start ← WebOTX の起動
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

終了スクリプト [stop.sh]

```
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
###    WebOTX stop
        /etc/init.d/WOAgentSvc stop                                ← WebOTX の停止
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
###    WebOTX stop
        /opt/init.d/WOAgentSvc stop                                ← WebOTX の停止
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

II. 双方向スタンバイ

起動スクリプト [start.sh]

```
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
###    WebOTX start
        /opt/WebOTX/bin/otxadmin start-domain --domainindir /mnt/webotx1/domains d
omainl ← ドメイン起動コマンドを追加
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
###    WebOTX start
        /opt/WebOTX/bin/otxadmin start-domain --domainindir /mnt/webotx1/domains d
omainl ← ドメイン起動コマンドを追加
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```

終了スクリプト [stop.sh]

```
if [ "$CLP_EVENT" = "START" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "NORMAL1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "NORMAL2"
        else
            echo "ON_OTHER1"
        fi
    else
        echo "ERROR_DISK from START"
    fi
###    WebOTX stop
        /opt/WebOTX/bin/otxadmin stop-domain --domainindir /mnt/webotx1/domains do
main1                                ← ドメイン停止コマンドを追加
elif [ "$CLP_EVENT" = "FAILOVER" ]
then
    if [ "$CLP_DISK" = "SUCCESS" ]
    then
        echo "FAILOVER1"
        if [ "$CLP_SERVER" = "HOME" ]
        then
            echo "FAILOVER2"
        else
            echo "ON_OTHER2"
        fi
    else
        echo "ERROR_DISK from FAILOVER"
    fi
###    WebOTX stop
        /opt/WebOTX/bin/otxadmin stop-domain --domainindir /mnt/webotx1/domains do
main1                                ← ドメイン停止コマンドを追加
else
    echo "NO_CLP"
fi
echo "EXIT"
exit 0
```


4. クライアントの仮想 IP アドレスの指定方法

この章では、クライアントから仮想 IP アドレスを指定する方法について説明します。

4.1 Java

クライアントは、この節で示した各サーバ側コンポーネントの種類に応じて以下のプロパティを Java VM に設定してください。

クライアント側とサーバ側で利用する JDK のマイナー・バージョンは同一でなければならないことに注意してください。JDK 1.4.2 と 5.0 との間の呼び出しは動作保証していません。バージョンの識別方法は、コマンドラインから「java -fullversion」と実行して出力されるバージョン情報を参照します。例えば、出力結果が「java full version "1.4.2_11-b06"」だった場合、メジャー・バージョン「1.4」、マイナー・バージョン「2」、アップデート・バージョン「11」になります。この時、クライアント側とサーバ側で動作する JDK バージョンは、1.4.2 でなければなりません。なお、アップデート・バージョンは任意のものを利用できます。

4.1.1 Enterprise Bean (EJB)

EJB のクライアント・プログラムは、WebOTX のサーバ側実行環境で動作する EJB コンポーネントにアクセスする前に、まず JNDI ルックアップを行ってコンポーネントの位置を特定します。この時、JNDI サーバに接続するための JNDI 用プロパティに仮想 IP アドレスで動作するサーバを指定します。JNDI のプロパティは次のとおりです。

プロパティ名	値
java.naming.provider.url	corbaname://<仮想 IP>:<NameServicePort>

<NameServicePort>で指定するポートは、2.2.4 節で定義した、「server.objectbrokerservice.namesv.NameServicePort」キーの値です。

4.1.2 CORBA

CORBA のクライアント・プログラムは、WebOTX のサーバ側実行環境で動作する CORBA コンポーネントにアクセスするために Object Broker にアクセスし、オブジェクト・リファレンスを取得します。Object Broker に接続するために、次のように仮想 IP アドレスで動作するサーバを指定します。

プロパティ名	値
org.omg.CORBA.ORBInitialHost	割り当てた仮想 IP アドレス
org.omg.CORBA.ORBInitialPort	ドメイン作成プロパティ・ファイルの OadPort の値

Transaction サービスを使用してクライアントでトランザクションを開始する場合は、次のプロパティも設定します。

プロパティ名	値
TFDecision	1
TFMachine	「domains/<domain_name>/config/TS/jta.conf」ファイル内の LogicalHostname で設定した仮想 IP アドレス、あるいは仮想コンピュータ名

4.2 CORBA C++

CORBA のクライアント・プログラムは、WebOTX のサーバ側実行環境で動作する CORBA コンポーネントにアクセスするために Object Broker にアクセスし、オブジェクト・リファレンスを取得します。Object Broker に接続するために、

次のように仮想 IP アドレスで動作するサーバを指定します。

4.2.1 Windows (Visual C++)

Windows の[スタート] メニューから[すべてのプログラム]→[WebOTX Object Broker]→[ORB Custom]を選択し、ORB Custom を起動します。次に、現れた画面の左下にある[詳細設定]ボタンをクリックします。そして、以下の項目を設定します。

項目名	値
OAD のポート番号	ドメイン作成プロパティ・ファイルの OadPort の値 デフォルトのポートを指定する場合は 0 を入力
名前サーバのホスト名	仮想コンピュータ名
URL のデフォルトポート番号	ドメイン作成プロパティ・ファイルの CorbalocDefaultPort の値

Transaction サービスを使用してクライアントでトランザクションを開始する場合は、次の設定をレジストリに追加してください。

[HKEY_LOCAL_MACHINE\SOFTWARE\NEC\WebOTX_S\Client]

キー	データ型	値
TFDecision	REG_DWORD	1
TFMachine	REG_SZ	「domains/<domain_name>/config/TS/jta.conf」ファイル内の LogicalHostname で設定した仮想 IP、あるいは仮想コンピュータ名

4.2.2 Linux (gcc)

/opt/ObjectSpinner/conf/orbconf ファイルを編集します。以下に示すキーが未定義の場合は追加します。

キー	値
OadPort	ドメイン作成プロパティ・ファイルの OadPort の値
NameServiceHostName	仮想コンピュータ名
CorbalocDefaultPort	ドメイン作成プロパティ・ファイルの CorbalocDefaultPort の値

Transaction サービスを使用してクライアントでトランザクションを開始する場合は、/etc/WebOTX/TS/WebOTX_TS.conf ファイルの[Client]セクションを編集します。以下に示すキーが未定義の場合は追加します。

キー	値
TFDecision	1
TFMachine	「domains/<domain_name>/config/TS/jta.conf」ファイル内の LogicalHostname で設定した仮想 IP、あるいは仮想コンピュータ名

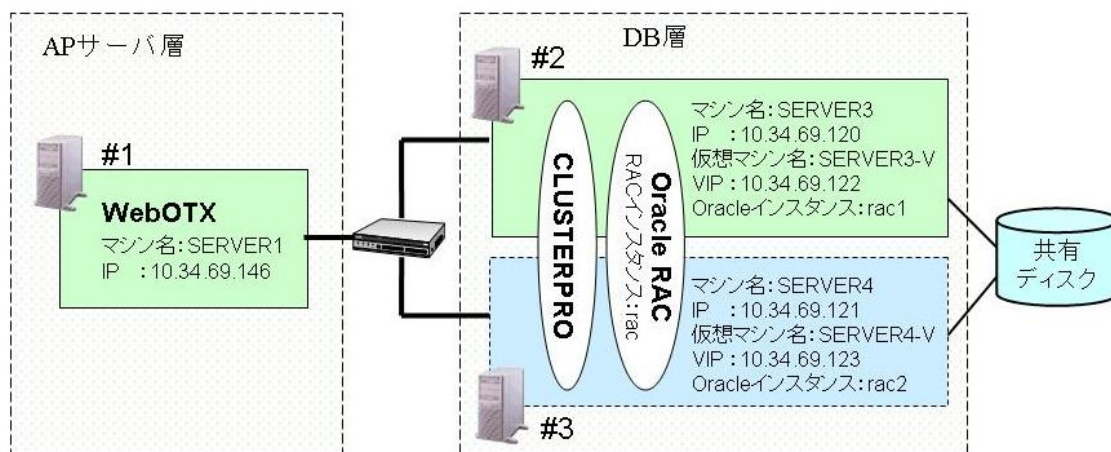
4.3 Visual Basic

マイクロソフト Visual Basic がクライアントになる場合、クライアント側で設定する項目は、「4.2.1 Windows」と同じです。その節の説明を参照してください。

5. Oracle RAC 連携

この章では、Oracle RAC と連携する場合の WebOTX の設定について説明します。

ここでは、以下の図のような構成でセットアップが完了していることを前提にしています。WebOTX がクラスタ構成の場合でも設定方法に違いはありませんので、クラスタを組んだ後に設定を行ってください。



マシン#1 (WebOTX)		
	IP アドレス	10.34.69.146

マシン#2 (Oracle)		
	IP アドレス	10.34.69.120
	仮想コンピュータ名	SERVER3-V
	仮想 IP アドレス	10.34.69.122
	Oracle インスタンス名	rac1

マシン#3 (Oracle)		
	IP アドレス	10.34.69.121
	仮想コンピュータ名	SERVER4-V
	仮想 IP アドレス	10.34.69.123
	Oracle インスタンス名	rac2

Oracle RAC インスタンス名: rac

次からは設定手順を示していきます。

1. XA のリカバリを可能にするため、データベースユーザに「DBA_PENDING_TRANSACTIONS」の SELECT 権限を付与します。
2. データベースクラスタを使用するため、データベースユーザに sys.dbms_system パッケージへの EXECUTE 権限を付与します。(※ 2 フェーズ・コミット使用時のみ)
3. WebOTX のドメインを停止します。
4. 「domains¥domain1¥lib¥ext」ディレクトリに Oracle の JDBC ドライバ(ojdbc14.jar、orai18n.jar

[または nls_charset12.jar))を配置します。

5. WebOTX のドメインを起動します。
6. JDBC データソースの登録をします。特に変更が必要な設定は次の 2 つです。

- データソースの種別: JDBCCEX_Oracle
- データベースクラスタの使用有無: true (※ 2 フェーズ・コミット使用時のみ)

その他設定は必要に応じて変更します。

次にデータソース名の設定例を示します。

- JDBC Type 4 ドライバ使用の場合
- ※ 1 行で記述し、優先使用する接続先のアドレスを先に書きます。

```
jdbc:oracle:thin:@(DESCRIPTION = (ENABLE=BROKEN)
  (LOAD_BALANCE=OFF) (FAILOVER=ON)
  (ADDRESS_LIST=
    (ADDRESS = (PROTOCOL = TCP) (HOST = SERVER3-V) (PORT = 1521))
    (ADDRESS = (PROTOCOL = TCP) (HOST = SERVER4-V) (PORT = 1521))
  )
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = rac)
  ))
```

- JDBC Type 2 ドライバ使用の場合

```
jdbc:oracle:oci8:@RAC
```

なお、JDBC Type 2 ドライバを使用の場合は、以下に示すとおり、Oracle への tnsnames.ora ファイル設定が必要です。

```
[%ORACLE_HOME%\network\admin\tnsnames.ora]
```

※ 優先使用する接続先のアドレスを先に書きます。

```

RAC =
  (DESCRIPTION = (ENABLE=BROKEN)    ← クライアント側の KeepAlive を有効にする
    (LOAD_BALANCE=OFF)
    (FAILOVER=on)                    ← 接続先フェイルオーバを有効にする
    (ADDRESS_LIST=
      (ADDRESS = (PROTOCOL = TCP) (HOST = SERVER3-V) (PORT = 1521))
      (ADDRESS = (PROTOCOL = TCP) (HOST = SERVER4-V) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = rac)
    )
  )

```

7. 登録後、WebOTX 統合運用管理ツール[<domain_name>]-[リソース]-[jdbc データソース]を右クリックして「JDBC データソースのテスト」を実施します。