



Oracle Service Busの高可用性 およびサーバー全体の移行

ORACLE®

目次

1. はじめに.....	1
1.1. OSB HA リファレンス・トポロジ.....	1
2. インストールおよび事前構成	3
2.1. インストール・ソフトウェア	3
2.2. データベースおよび事前設定	3
2.2.1. RACデータベース要件	3
2.2.2. 初期化パラメータ	3
2.2.3. データベース・サービス	4
2.3. ネットワーク	4
2.3.1. IPおよび仮想IP	4
2.3.2. ポート	5
2.4. 共有ストレージおよび推奨ディレクトリ構造	5
2.4.1. ディレクトリおよびディレクトリ環境変数の用語	5
2.4.2. 異なるディレクトリの推奨場所	6
3. OSBドメインの作成	9
3.1. Oracle Service Busミドルウェア・ホームのインストール	9
3.1.1. Oracle Service Busサーバーのインストール	9
3.2. インストールのバックアップ	10
3.3. OSBHOST1でOSBHOST1VHN1の有効化	10
3.4. OSBHOST1でConfiguration Wizardを実行してドメインの作成	10
3.5. 管理対象サーバー用のドメイン・ディレクトリの作成	13
3.6. OSBHOST1で管理サーバー用のboot.propertiesの作成	14
3.7. OSBHOST1で管理サーバーの起動	14
3.8. 管理サーバーの検証	14
3.9. 管理対象サーバーWLS_OSB1のホスト名検証の無効化	15
3.9.1. ノードマネージャのプロパティの変更	15
3.10. OSBHOST1とOSBHOST2でノードマネージャの起動または再起動	15
3.11. ドメインの変更を管理対象サーバーのドメイン・ディレクトリに反映	16
3.12. OSB_Clusterのクラスタ・アドレスの設定	16
3.13. OSBHOST1で管理対象サーバーWLS_OSB1の起動	16
3.14. 管理対象サーバーWLS_OSB1の検証	16
3.15. unpackユーティリティを使用して、OSBHOST2へドメイン構成を反映	17
3.16. 管理対象サーバーWLS_OSB2のホスト名検証の無効化	17
3.17. OSBHOST2でノードマネージャの再起動	17
3.18. 管理対象サーバーWLS_OSB2の起動および検証	17
3.19. 共有JMS永続ストアの構成	18
3.20. トランザクション・リカバリためのデフォルト永続ストアの構成	18
3.21. Oracle RAC DatabaseのJDBCデータソースの構成	18
3.22. 手動で管理サーバーをOSBHOST2にフェイルオーバー	20
3.23. 管理サーバーをOSBHOST1にフェイルバック	21
3.24. インストールのバックアップ	21
4. ノードマネージャの設定	22
4.1. ノードマネージャについて	22
4.2. OSBHOST1でのノードマネージャ用のホスト検証証明書の有効化	22
4.2.1. utils.GertGenユーティリティを使用して、自己署名証明書の生成	22
4.2.2. utils.ImportPrivateKeyユーティリティを使用して、IDキースタの作成	23
4.2.3. keytoolユーティリティを使用して、信頼キースタの作成	23
4.2.4. カスタム・キースタを使用するためのノードマネージャの構成	24
4.3. OSBHOST1でのノードマネージャの起動	25
4.4. OSBHOST2でのノードマネージャ用のホスト検証証明書の有効化	25
4.4.1. utils.CertGenユーティリティを使用して、自己署名証明書の生成	25
4.4.2. utils.ImportPrivateKeyユーティリティを使用して、IDキースタの作成	26
4.4.3. keytoolユーティリティを使用して、信頼キースタの作成	26
4.4.4. カスタム・キースタを使用するためのノードマネージャの構成	27

4.5. OSBHOST2でのノードマネージャの起動.....	28
5. サーバーの移行.....	29
5.1. サーバー移行用のユーザー、表領域およびテーブルの作成.....	29
5.2. マルチデータソースの作成.....	29
5.3. OSBHOST1とOSBHOST2と管理サーバーの間でのホスト名検証証明書の有効化.....	30
5.4. ノードマネージャのプロパティ・ファイルの編集.....	30
5.5. wlsifconfig.shスクリプトの環境変数およびスーパーユーザーの権限の設定.....	31
5.6. サーバー移行ターゲットの構成.....	32
5.7. サーバー移行のテスト.....	33
付録A.....	34
A.1 用語.....	34
A.2 共有ストレージの設定.....	35
A.3 ロードバランサの要件.....	36
A.4 ユニキャスト要件.....	37
A.5 OSB参考資料.....	37

1. はじめに

Oracle Service Bus(以下OSB)は、Oracle Fusion Middleware製品ファミリーにおけるSOA SuiteおよびEvent Driven Architecture Suiteのキー・コンポーネントです。OSBは、効果的で、シームレスなユーザー・エクスペリエンスで、1つの製品でオペレーショナル・サービス管理とともに独自にエンタープライズ・サービス・バス(ESB)の統合能力を達成します。その柔軟なデプロイメント・オプションとSOAガバナンスとの相互運用で、OSBはデプロイメント、管理、そして部門から企業スケールまでサービス指向アーキテクチャ(SOA)を実装しようとするガバナンス・チャレンジを取り扱うようにデザインされています。

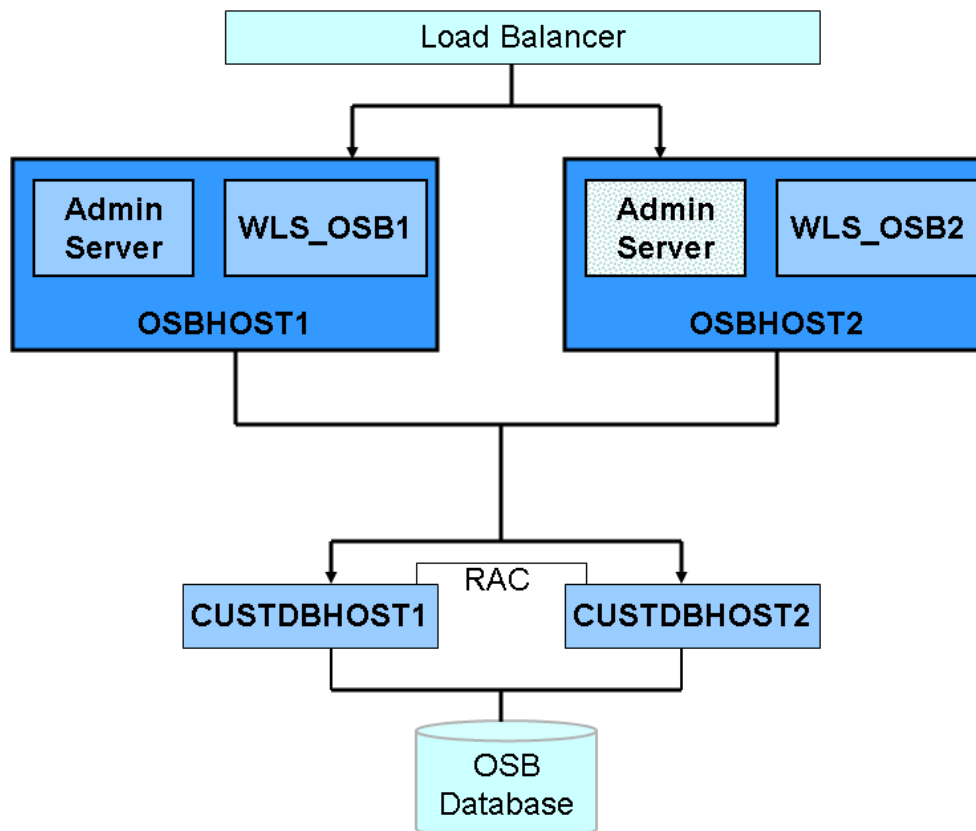
OSBは証明された、軽量なSOA統合プラットフォームです。Webサービスだけではなく、Javaや、.Net、メッセージング・サービス、レガシー・エンドポイントといった異機種サービス間の相互作用をコネクティング、仲介、そして管理するようにデザインされています。このドキュメントでは、OSBの高可用性の設計と実装について、詳細な手順を提供します。高可用性は、ミッションクリティカルなプロダクション・デプロイメントにとって重要です。

このドキュメントで説明するベストプラクティスは、高可用性、スケーラビリティおよび全体OSBテクノロジー・スタック(Oracle RAC Database、WebLogic管理サーバー、Oracle OSB管理対象サーバー)でのサーバーの移行を含みます。

1.1. OSB HA リファレンス・トポロジ

推奨の最も基本的なアーキテクチャは図1に表示されています。OSBサーバー層は、ロードバランサの背後にあり、クラスタ化された2つOSBサーバーで構成されています。データベース層は2つノードのRACクラスタで構成されています。この構成は、OSBノードの障害とDBノードの障害に対して保護を与えます。完備性のために、ロードバランサも構成されています。クラスタにデプロイされたOSBシステムがHTTP呼び出しによって開始される場合のみ、ロードバランサが役立ちます。

図1: OSB HAリファレンス・トポロジ図



このアーキテクチャには、Oracle WebLogic Serverを実行する2つのノードOSBHOST1とOSBHOST2があります。OSBHOST1とOSBHOST2は、OSBコンポーネントを実行する管理対象サーバーで構成されています。管理対象サーバーは、アクティブ-アクティブ方式で構成されています。

OSBHOST1とOSBHOST2は、アクティブ・パッシブ構成でOracle WebLogic Server Administration Consoleを実行します。手動で管理サーバーをフェイルオーバーできます(項3.22「手動で管理サーバーをOSBHOST2にフェイルオーバー」を参照してください)。または、CFC/CRSを設定し、自動的に別のハードウェア・クラスタ(アーキテクチャには表示されていません)にフェイルオーバーすることもできます。

RACデータベースは、ノードCUSTDBHOST1とCUSTDBHOST2の上で動作しています。データベースには、OSBコンポーネントに必要なスキーマが含まれています。OSBコンポーネントは、データベース層に接続したアプリケーション層で動作しています。

2. インストールおよび事前構成

この項では、必要なインストールおよび設定手順について、説明します。

2.1. インストール・ソフトウェア

表2-1では、各ソフトウェア・コンポーネントのインストール・ソースを示しています。詳細については、「[インストール・ガイド](#)」を参照してください。

表2-1コンポーネントおよびインストール・ソース

コンポーネント	配布メディア
Oracle Database 10gまたは11g	Oracle Database CD(10.2.0.4以降、または11.1.0.7以降)
Oracle OSB 10gR3 Server	Oracle WebLogic Server(10gR3)DVD
Oracle HTTP Server	Oracle Fusion Middleware WebTier and Utilities 11g(11.1.1.1.0)DVD

2.2. データベースおよび事前設定

データベースには、様々なOSBコンポーネントとサーバー全体移行に必要なスキーマが含まれています。

2.2.1. RACデータベース要件

Oracle RACデータベースをインストールします。以下のバージョンをサポートします。

- ✧ Oracle Database 11g(11.1.0.7以降)。災害復旧が計画または予想される場合、Enterprise Editionを使用することをお勧めします。
- ✧ Oracle Database 10g(10.2.0.4以降)。Oracle Fusion Middleware 11gのリリース1は、Oracle Database 10gのExpress Edition(OracleデータベースXE)をサポートしません。災害復旧が計画または予想される場合、Enterprise Editionを使用することをお勧めします。

データベースのリリースを確認するには、以下のようにPRODUCT_COMPONENT_VERSIONビューを問合せします。

```
SQL> SELECT VERSION FROM SYS.PRODUCT_COMPONENT_VERSION WHERE PRODUCT LIKE 'Oracle%';
```

2.2.2. 初期化パラメータ

以下の初期化パラメータが必要な最小値に設定されていることを確認します。

表2-1必要な初期化パラメータ

スキーマ	パラメータ	必要な値	パラメータ・クラス
OSB	PROCESSES	300以上	Static

初期化パラメータの値を確認するには、SQL*Plusを使用して、SHOW PARAMETERコマンドを実行します。

SYSユーザーとして、以下のようにSHOW PARAMETERコマンドを実行します。

```
SQL> SHOW PARAMETER processes;
```

以下のコマンドを実行して、初期化パラメータを設定します。

```
SQL> ALTER SYSTEM SET processes=300 SCOPE=SPFILE;
```

データベースを再起動します。

注: パラメータの値を変更する方法は、パラメータが静的であるか動的であるか、またデータベースがパラメータ・ファイルを使用しているか、サーバー・パラメータ・ファイルを使用しているかによって異なります。パラメータ・ファイル、サーバー・パラメータ・ファイル、およびパラメータの変更方法については、「Oracle Database管理者ガイド」を参照してください。

2.2.3. データベース・サービス

OSBサーバーがデータベースに接続するために、データベース・サービスを作成する必要があります。データベース・サービスの作成手順については、「Oracle ClusterwareおよびOracle Real Application Cluster管理およびデプロイメント・ガイド」の「ワークロード管理の概要」章を参照してください。

以下の手順に従って、SQL*Plusを使用して、データベース・サービスを作成します。

1. サブプログラムCREATE_SERVICEを使用して、データベース・サービスosbsvc.mycompany.comを作成します。sysdbaユーザーとして、SQL*Plusにログインし、以下のコマンドを実行します。

```
SQL> EXECUTE  
DBMS_SERVICE.CREATE_SERVICE(SERVICE_NAME=>'osbsvc.mycompany.com',NETWORK_NAME=>  
osbsvc.mycompany.com');
```

2. srvctlを使用して、データベースにサービスを追加し、インスタンスに割り当てます。

```
prompt> srvctl add service -d osbdb -s osbsvc -r osbdb1,osbdb2
```

3. srvctl1を使用して、サービスを起動します。

```
prompt> srvctl start service -d osbdb -s osbsvc
```

注: SRVCTLコマンドの詳細については、「Real Application Cluster管理およびデプロイメント・ガイド」を参照してください。

2.3. ネットワーク

この項では、以下の事項について説明します。

項2.3.1 IPおよび仮想IP

項2.3.2 ポート

2.3.1. IPおよび仮想IP

異なる仮想IP(VIP)と物理IPアドレスをリスニングするように、管理サーバーと管理対象サーバーを設定します。各VIPとIPは、WebLogicサーバーに接続され、使用されます。OSBHOST2で管理サーバーを再起動するために、手動でVIP1をフェイルします。VIP2とVIP3は、Oracle WebLogic Serverの移行機能によって、OSBHOST1からOSBHOST2へ、OSBHOST2からOSBHOST1へフェイルオーバーされます。物理IP(非仮想)が各ノードに固定されています。IP1はOSBHOST1の物理IPで、IP2はOSBHOST2の物理IPです。

表2-2には、様々な仮想ホストの説明が記載されています。

表2-2仮想ホスト

仮想IP	VIPのマップ	説明
VIP1	OSBHOST1VHN1	OSBHOST1VHN1は、管理サーバーの仮想ホスト名で、管理サーバーのリسن・アドレスで、手動でフェイルオーバーされます。OSBHOST1VHN1は、管理サーバー・プロセスを実行しているノード(デフォルトではOSBHOST1)で有効になっています。
VIP2	OSBHOST1VHN2	OSBHOST1VHN2は、仮想ホスト名で、WLS_OSB1のリسن・アドレスにマップされます。OSBHOST1VHN2は、管理対象サーバー(WLS_OSB1)の移行に伴って、フェイルオーバーされます。OSBHOST1VHN2は、WLS_OSB1プロセスを実行しているノード(デフォルトではOSBHOST1)で有効になっています。
VIP3	OSBHOST2VHN1	OSBHOST2VHN1は、仮想ホスト名で、WLS_OSB2のリسن・アドレスにマップされます。OSBHOST2VHN1は、管理対象サーバー(WLS_OSB2)の移行に伴って、フェイルオーバーされます。OSBHOST2VHN1は、WLS_OSB2プロセスを実行しているノード(デフォルトではOSBHOST2)で有効になっています。

2.3.2. ポート

Oracle Service Busミドルウェアの多くのコンポーネントおよびサービスがポートを使用します。管理者として、これらのサービスが使用するポート番号を知っている必要があります。同じポート番号がホスト上の2つのサービスによって使用されないように注意します。

ほとんどのポート番号はインストール時に割り当てられます。

表2-3では、OSBトポロジで使用しているポートを示しています。

表2-3使用されるポート

タイプ	ポート および ポート範囲	プロトコル・ アプリケーション	他の考慮事項とタイムアウトのガイドライン
管理サーバー	7001	HTTP/t3	タイムアウトを短い期間(5~10秒)に設定します。
OSB管理対象サーバー	8001	HTTP/ WLS_OSBn	タイムアウトは、OSBが使用したプロセス・モデルの種類によって異なります。
OSBクラスタ・メンバー間の通信	8001	TCP/IP	デフォルトでは、サーバーのリسن・アドレスと同じポートを使用します。
WebLogicサーバー・クラスタ内のセッション・レプリケーション	n/a	n/a	デフォルトでは、サーバーのリسن・アドレスと同じポートを使用します。
ノードマネージャ	5556	TCP/IP	n/a
データベース接続	1521	SQL*Net	タイムアウトは、データベースのコンテンツとOSBが使用したプロセス・モデルのタイプに依存します。

2.4. 共有ストレージおよび推奨ディレクトリ構造

この項では、使用したディレクトリおよびディレクトリ構成について説明します。他のディレクトリ・レイアウトも可能ですが、このガイドで使用したモデルは、最高のコンポーネントの分離と対称性を提供し、バックアップおよび災害復旧を容易にし、可用性を最大化にすることができます。ドキュメントの残りの部分は、以下のディレクトリ構造とディレクトリ用語を使用しています。

2.4.1. ディレクトリおよびディレクトリ環境変数の用語

- ✧ ORACLE_BASE:この環境変数と関連ディレクトリ・パスは、Oracle製品がインストールされているベースディレクトリを指します。
- ✧ MW_HOME:この環境変数と関連ディレクトリ・パスは、OSBミドルウェアが存在している場所を指します。
- ✧ WL_HOME:この環境変数と関連ディレクトリ・パスには、WebLogicサーバーをホストするために必要なインストール・ファイルが含まれています。
- ✧ ドメイン・ディレクトリ:このディレクトリ・パスは、Oracle WebLogicドメイン情報(コンフィグレーション・アーティファクト)が格納されている場所を指します。次に説明したように、同じノード上の異なるWLSサーバーは、異なるドメイン・ディレクトリを使用することができます。

2.4.2. 異なるディレクトリの推奨場所

Oracle Service Busミドルウェアは、単一のバイナリ・インストールから複数のOSBサーバーを作成することができます。共有ストレージ上の1つ場所にバイナリをインストールし、異なるノード上のサーバーがこのインストールを使用することができます。しかし、可用性を最大化するために、冗長なバイナリをインストールすることをお勧めします。例えば、共有ストレージに2つMW ホーム(各MWホームには、WL_HOMEと各製品スイート用のORACLE_HOMEが含まれています)をインストールします。同じ種類のサーバーを追加する場合(スケールアウトまたはスケールアップ)、インストールを必要とせずに、この2つの場所のいずれかを使用できます。冗長なバイナリをインストールするために、2つの異なるボリュームを使用することをお勧めします。これにより、ボリュームの障害を分離できます。保護のために、これらのボリュームをミラー化することをお勧めします。複数のボリュームが使用できない場合、マウント・ポイントを使用して、共有ストレージ上の異なるディレクトリを同じ場所にマウントします。これは複数のボリュームが提供する保護を保証できませんが、ユーザーの削除とファイルの破損に対して、保護を提供します。

ORACLE_HOMEまたはWL_HOMEが異なるノード上の複数のサーバーによって共有されている場合、バッチおよびインストールの一貫性のために、各ノード上のOracle Inventoryとミドルウェア・ホーム・リストの更新を保持することをお勧めします。ノードでOraInventoryを更新し、共有ストレージにあるインストールをノードに接続するには、ORACLE_HOME/oui/bin/attachHome.shを使用します。ミドルウェア・ホーム・リストを更新し、WLS_HOMEを追加または削除するには、<user_home>/bea/beahomelistファイルを編集します。これは、クラスタ中のすべてのノードに対して必要となります。

このトポロジでは、管理対象サーバーが使用するドメイン・ディレクトリから管理サーバーが使用するドメイン・ディレクトリを分離します。これにより、管理対象サーバーが使用するドメイン・ディレクトリの対称構成を実現できるし、管理サーバーのフェイルオーバーを分離します。管理サーバーのドメイン・ディレクトリは、同じ構成を持つ別のノードにフェイルオーバーするために、共有ストレージに配置する必要があります。管理対象サーバーのドメイン・ディレクトリは、ローカルまたは共有ストレージに配置することができます。

異なるノード上の管理対象サーバーは共有ストレージ上のドメイン・ディレクトリを使用するか、個別のドメイン・ディレクトリを使用することができます。管理対象サーバーのドメイン・ディレクトリを共有すると、スケールアウトの手続きが容易にできます。共有ボリュームをマウントするために、デプロイメントはストレージ・システムの要件(要件がある場合)に準拠する必要があります。

複数のローカル・ドメインに適用されるすべてのプロシージャは、単一の共有ドメインにも適用されます。ここで、各ノードが個別のドメイン・ディレクトリを使用するモデルを使用します。ドメイン・ディレクトリがローカルであるか、共有ストレージに配置することもできます。

JMSファイル・ストアとJTAトランザクション・ログは、サーバーの障害が発生し、移行する場合、回復のために複数のボックスから使用できるよう共有ストレージに配置する必要があります。

上記の前提に基づいて、次は推奨ディレクトリについて説明します。共有ストレージの場所が指定されている場合、共有ストレージがそのディレクトリに必要とされることを意味します。ローカル・ディスクまたは共有ストレージが両方使用可能な場合、共有ディスクを使用する場合は、マウント仕様が修飾されています。共有ストレージの場所は例であり、変更することができます。しかし、この構造が一貫性を保持し、簡単です。

- ✧ ORACLE_BASE: /u01/app/oracle
- ✧ MW_HOME(アプリケーション層): ORACLE_BASE/product/osbmw
- ✧ WL_HOME: MW_HOME/wlserver_10.3
- ✧ OSB_HOME: MW_HOME/osb_10.3

- 管理サーバーのドメイン・ディレクトリ(二番目のdomain_nameは、設定ウィザードによって追加されます。)

ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>

- 管理対象サーバーのドメイン・ディレクトリ

ORACLE_BASE/admin/<domain_name>/mserver/<domain_name>

注:この手順は、共有ストレージに依存します。上記の例は、NASに固有のものですが、他のストレージ・タイプは異なるマッピング・タイプを使用して、冗長性を提供します。

- JMSのファイルベースのストアとTlogsの場所(OSBのみ):

ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/jms

ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/tlogs

- アプリケーション・ディレクトリの場所

ORACLE_BASE/admin/<domain_name>/apps

図2-1ディレクトリ構造

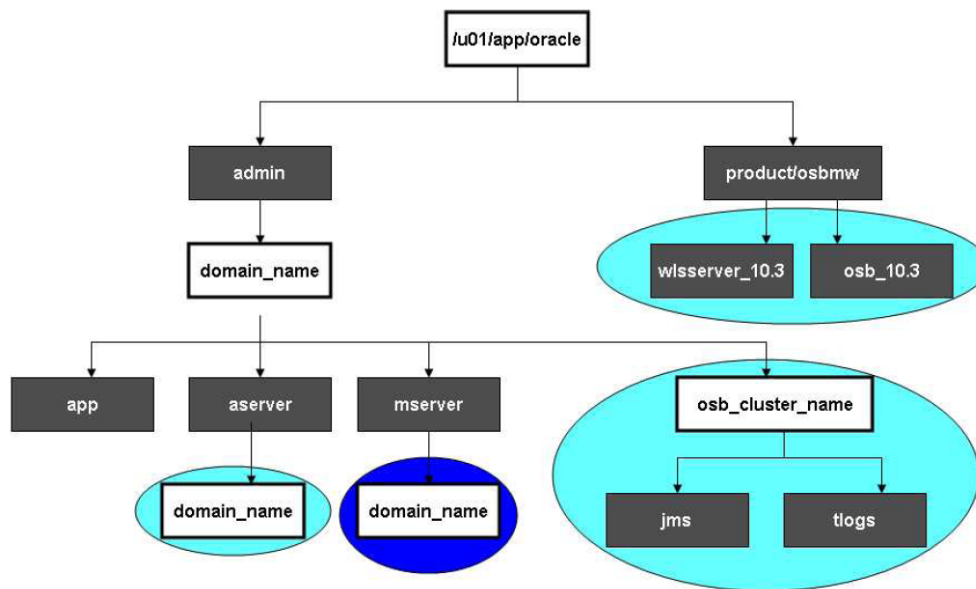


表2-4ディレクトリ構造要素

要素	説明
	管理サーバーのドメイン・ディレクトリ、アプリケーション、デプロイメント・プラン、ファイル・アダプタ制御ディレクトリ、JMSとTXのログ、および全体MW_HOMEは、共有ディスクに格納されます。
	管理対象サーバーのドメイン・ディレクトリは、ローカル・ディスクまたは共有ディスクに配置することができます。複数のノードで管理対象サーバーのドメイン・ディレクトリを共有する場合、各ノードは同じ共有ディスクの場所にマウントする必要があります。Web層のinstance_nameディレクトリは、ローカル・ディスクまたは共有ディスクに配置することができます。
	固定された名前
	インストールに依存する名前

表2-5はドメインのディレクトリ構造を纏めます。

表2-5共有ストレージの内容

サーバー	データ・タイプ	共有ストレージ のボリューム	ディレクトリ	ファイル
WLS_OSB 1/ WLS_OSB 2	Txログ	VOL1	ORACLE_BASE/adm in/<domain_name>/ <osb_cluster_name>/ tlogs	トランザクション・ディレクトリは共通ですが (WebLogic サーバーにより決定される)、フ ァイルは個別です。
WLS_OSB 1/ WLS_OSB 2	JMSストア	VOL1	ORACLE_BASE/adm in/<domain_name>/ <osb_cluster_name>/ jms	トランザクション・ディレクトリは共通ですが (WebLogic サーバーにより決定される)、フ ァイルは個別です。例えば、OSBJMSStore1 やUMSJMSStore1など。
WLS_OSB 1	WLS インストール	VOL1	ORACLE_BASE/pro duct/osbmw	各ボリュームで個別ですが、両方のサーバ ーは同じディレクトリ構造を参照します。
WLS_OSB 2	WLS インストール	VOL2	ORACLE_BASE/pro duct/osbmw	各ボリュームで個別ですが、両方のサーバ ーは同じディレクトリ構造を参照します。
WLS_OSB 1	OSB インストール	VOL1	ORACLE_BASE/pro duct/osbmw/osb_10. 3	各ボリュームで個別ですが、両方のサーバ ーは同じディレクトリ構造を参照します。
WLS_OSB 2	OSB インストール	VOL2	ORACLE_BASE/pro duct/osbmw/osb_10. 3	各ボリュームで個別ですが、両方のサーバ ーは同じディレクトリ構造を参照します。
WLS_OSB 1	ドメイン構成	VOL1	ORACLE_BASE/adm in/<domain_name>/ msserver/<domain_na me>	各ボリュームで個別ですが、両方のサーバ ーは同じディレクトリ構造を参照します。
WLS_OSB 2	ドメイン構成	VOL2	ORACLE_BASE/adm in/<domain_name>/ msserver/<domain_na me>	各ボリュームで個別ですが、両方のサーバ ーは同じディレクトリ構造を参照します。

共有ストレージの作成に関する詳細な指示は、「A2 共有ストレージの設定」で記述されています。

3. OSBドメインの作成

この項では、構成ウィザード、Oracle WebLogic Server Administration ConsoleおよびOracle Service Busを使用してドメインを作成する方法について説明します。

最初の手順は、OSBサーバー用のユーザーと表領域を作成します。

1. wls_osbと呼ばれる表領域を作成します。SYSDBAユーザーとしてSQL*Plusにログオンし、以下のコマンドを実行します。

```
SQL> create tablespace wls_osb_ts logging datafile 'DB_HOME/oradata/orcl/wls_osb_ts.dbf' size 32m
autoextend on next 32m maxsize 2048m extent management local;
```

2. wls_osbというユーザーを作成し、wls_osb_ts表領域をユーザーに割り当てます。

```
SQL> create user wls_osb identified by welcome1;
SQL> grant create table to wls_osb;
SQL> grant create session to wls_osb;
SQL> alter user wls_osb default tablespace wls_osb_ts;
SQL> alter user wls_osb quota unlimited on wls_osb_ts;
```

3.1. Oracle Service Busミドルウェア・ホームのインストール

項2.4「共有ストレージおよび推奨ディレクトリ構造」で説明したように、冗長性のために、少なくとも2つのストレージ場所にOracle Service Busミドルウェアをインストールします。

3.1.1. Oracle Service Busサーバーのインストール

以下の手順を実行して、OSBHOST1とOSBHOST2にOSBサーバーをインストールします。

1. OSBサーバーのインストーラを起動します。
2. 「Welcome」ページで、「Next」をクリックします。
3. 「Choose BEA Home Directory」ページで、以下の手順を実行します。
 - ✧ 「Create a New BEA Home」を選択します。
 - ✧ 「BEA Home Directory」として、「ORACLE_BASE/product/osbmw」と入力します。

注: ORACLE_BASEは、Oracle製品がインストールされているベースディレクトリです。推奨値は、/u01/app/oracleです。詳細については、項2.4「共有ストレージおよび推奨ディレクトリ構造」を参照してください。

4. 「Next」をクリックします。
5. 「Choose Install Type」ページで、「Typical」を選択し、「Next」をクリックします。以下の製品とコンポーネントを選択します。
 - ✧ WebLogic Server
 - ✧ Workshop
 - ✧ Oracle Service Bus
6. 「Choose Product Installation Directories」で、以下の値を受け取ります。「Next」をクリックします。
 - ✧ ORACLE_BASE/product/osbmw/wlserver_10.3
 - ✧ ORACLE_BASE/product/osbmw/workshop_10.3
 - ✧ ORACLE_BASE/product/osbmw/osb_10.3
7. 「Installation Summary」ページで、「Next」をクリックします。
8. 「Installation Complete」ページで、「Run QuickStart」のチェックを外し、「Done」をクリックします。

3.2. インストールのバックアップ

ミドルウェア・ホームをバックアップする必要があります。(サーバーが停止していることを確認してください)
以下のコマンドを実行して、Oracle WebLogic ServerおよびOSB用のミドルウェア・ホームをバックアップします。

```
OSBHOST1> tar -cvpf osbmwhomeback.tar ORACLE_BASE/product/osbmw
```

3.3. OSBHOST1でOSBHOST1VHN1の有効化

この手順は、OSBがインストールされているかどうかに関係なく、管理サーバーのフェイルオーバーに必要とされます。

管理サーバーを仮想IP(OSBHOST1VHN1)に関連付けます。OSBHOST1VHN1がOSBHOST1で有効になっていることを確認してください。

Linux上で仮想IPを有効にするには、rootとしてifconfigコマンドを実行します。

```
/sbin/ifconfig/ <interface:index> <IPAddress> netmask <netmask>  
/sbin/arping/ -q -U -c 3 -I <interface> <IPAddress>
```

例:

```
/sbin/ifconfig ethX:Y 100.200.140.206 netmask 255.255.255.0
```

```
/sbin/ifconfig ethX:Y 100.200.140.206 netmask 255.255.255.0
```

ネットワークを有効にし、仮想IPの新しい場所を登録します。

例:

```
/sbin/arping -q -U -c 3 -I ethX 100.200.140.206
```

以下のコマンド(ifconfigコマンドではなく)を実行することができます。

```
ip addr add 100.200.140.206/24 dev eth0
```

別のノートからpingを実行して、アドレスが有効になっていることを確認します。

例:

```
/bin/ping 100.200.140.206
```

この例では、「ethX」はイーサネット・インターフェイス(eth0またはeth1)であり、「Y」はインデックス(0、1、2など)です。

3.4. OSBHOST1でConfiguration Wizardを実行してドメインの作成

WLSホーム・ディレクトリから「Configuratin Wizard」を実行して、管理サーバーとOracle Service Busを含むドメインを作成します。

1. OSBスキーマをインストールしたデータベースが実行されていることを確認します。RACデータベースの場合、有効化の確認を信頼できるために、すべてのインスタンスが実行されていることをお勧めします。
2. 「Configuration Wizard」のディレクトリに移動します。
OSBHOST1>cd ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin
3. 以下のコマンドを実行して、「Oracle Fusion Middleware Configuration Wizard」を実行します。

```
OSBHOST1> ./config.sh
```

4. 「Welcome」ページで、「Create a New WebLogic Domain」を選択し、「Next」をクリックします。
5. 「Select Domain Source」ページが表示されます。「Generate a domain configured automatically」を選択し、以下の製品を選択します。
 - a. WebLogic Server(必要)
 - b. Workshop for WebLogic 10.3
 - c. Oracle Service Bus
 「Next」をクリックします。
6. 「Configure Administrator Username and Password」ページで、ドメイン管理者の名前とパスワードを入力します
「Next」をクリックします。
7. 「Configure Server Start Mode and JDK」ページで、以下の設定を行います。
「WebLogic Domain Startup Mode」として、「Production Mode」を選択します。
JDKとして、「JROCKIT SDK1.6.0_05」を選択します。
「Next」をクリックします。
8. 「Customize Environment and Services Settings」ページで、「Yes」を選択し、「Next」をクリックします。
9. 「Configure RDBMS Security Store Database」ページで、「Next」をクリックして、デフォルトの設定をそのまま使用します。
10. 「Configure the Administration Server」ページで、以下の値を入力します。
Name: AdminServer
Listen Address: OSBHOST1VHN1
Listen Port: 7001
SSL listen port: N/A
SSL enabled: 選択しない
「Next」をクリックします。
11. 「Configure Managed Servers」ページで、「Add」をクリックして、以下の管理対象サーバーを追加します。

表3-1 管理対象サーバー

Name	Listen Address	Listen Port	SSL Listen Port	SSL Enabled
WLS_OSB1	VIP1	8001	n/a	No
WLS_OSB2	VIP2	8001	n/a	No

「Next」をクリックします。

12. 「Configure Clusters」ページで、「Add」をクリックして、以下のクラスタを追加します。

表3-2 クラスタ

Name	Multicast Address	Multicast Port	Cluster Address
OSB_Cluster	デフォルトでは239.192.0.0	デフォルトでは7001	空のまま

「Next」をクリックします。

13. 「Assign Servers to Cluster」ページで、以下のようにサーバーをクラスタに割り当てます。

OSB_Cluster:

- a. WLS_OSB1
- b. WLS_OSB2

「Next」をクリックします。

14. 「Configure Machines」ページで、「Unix Machine」タブをクリックし、「Add」をクリックして、以下のマシンを追加します。

表3-3マシン

Name	Node Manager Listen Address	Node Manager Listen Port
OSBHOST1	OSBHOST1	5556
OSBHOST2	OSBHOST2	5556

他のフィールドの値をそのまま使用します。

「Next」をクリックします。

15. 「Assign Servers to Machines」ページで、以下のようにサーバーをマシンに割り当てます。

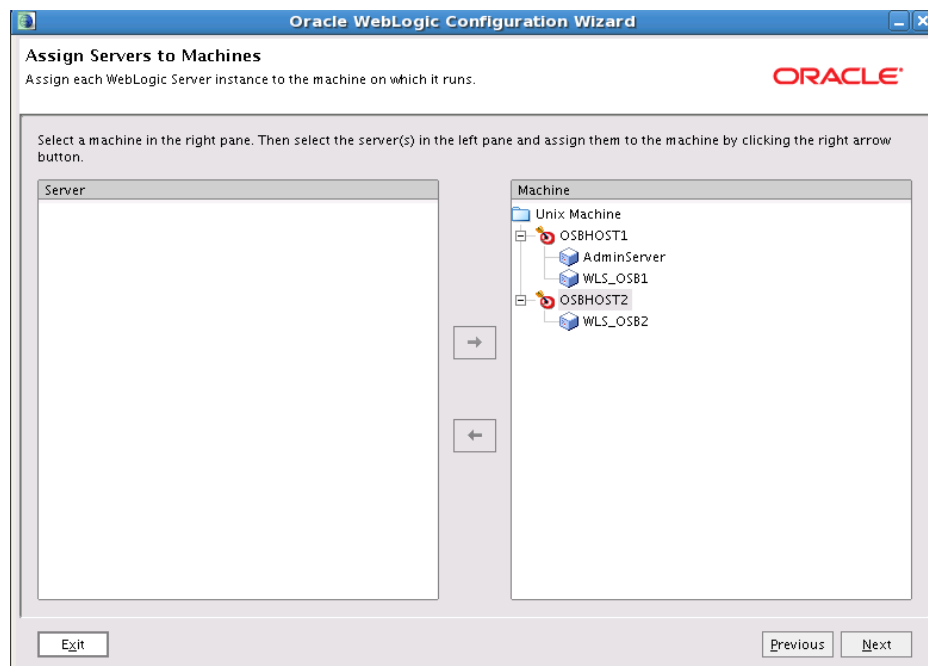
OSBHOST1:

✧ AdminServer

✧ WLS_OSB1

OSBHOST2:

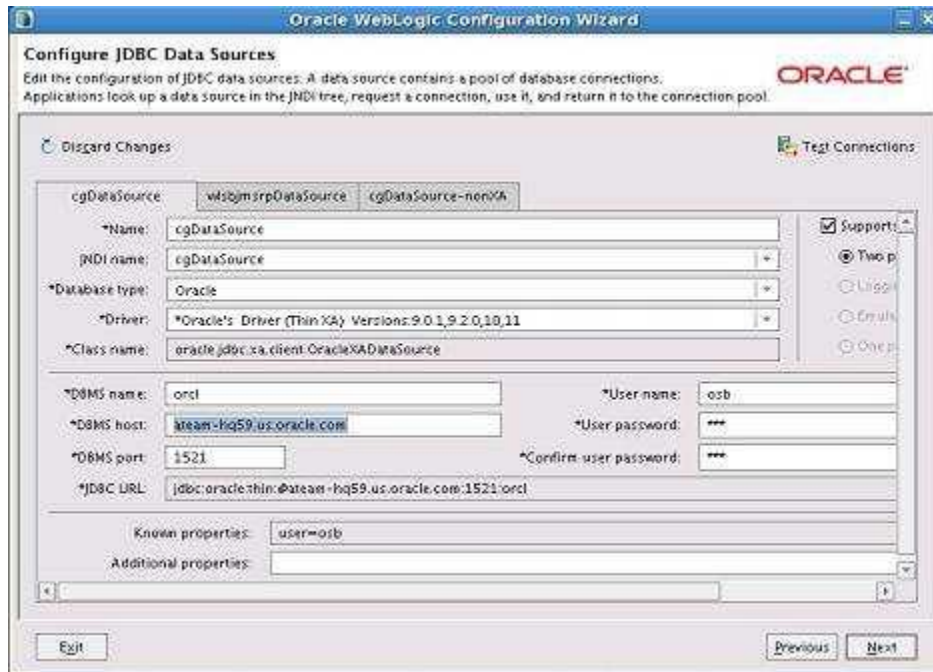
✧ WLS_OSB2



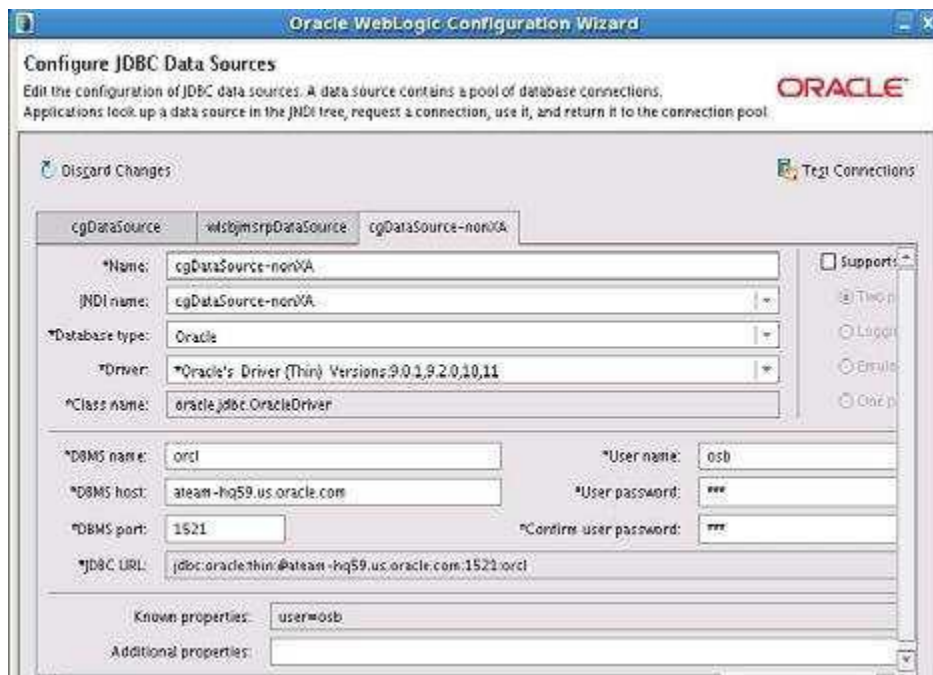
「Next」をクリックします。

16. 「Configure JDBC Data Sources」ページで、Oracleデータベースの資格証明および関連情報を入力します。項3.21「Oracle RAC DatabaseのJDBCデータソースの構成」では、WebLogicマルチデータソースを使用するように変更します。ユーザー名として、項3の最初で作成した「wls_osb」を使用する必要があります。

cgDataSourceおよびwlsjmsrpDataSourceは以下のようになります。



cgDataSource-nonXAは以下ようになります。



「Next」をクリックします。

17. 「Run Database Scripts」ページで、「Run Scripts」をクリックします。完了したら、「Next」をクリックします。
18. 「Configure JMS File Store」ページで、デフォルトの値をそのまま使用します。「Next」をクリックします。項3.19「共有JMS永続ストアの構成」で、これを共有ストレージに変更します。
19. 「Review WebLogic Domain」ページで、「Next」をクリックします。
20. 「Create Domain」ページで、ドメイン名を入力し、「Done」をクリックします。

3.5. 管理対象サーバー用のドメイン・ディレクトリの作成

項2.2「データベースと環境の事前構成」で推奨されているように、packおよびunpackコマンドを使用してOSBHOST1で管理対象サーバーによって使用されるドメイン・ディレクトリから、管理サーバーによって使用されるドメイン・ディレクトリを分離します。

1. 以下のように、OSBHOST1でpackコマンドを使用してテンプレート・パックを作成します。

```
OSBHOST1> cd ORACLE_BASE/product/osbmw/osb/common/bin
OSBHOST1> ./pack.sh -managed=true -
domain=ORACLE_BASE/admin/<domain_name>/aserver/<domain_name> -
template=osbdomaintemplate.jar -template_name=osb_domain_template
```

2. 以下のように、unpackコマンドでテンプレートを解凍して、管理対象サーバーのドメイン・ディレクトリを作成します。

```
OSBHOST1> cd ORACLE_BASE/product/osbmw/osb/common/bin
OSBHOST1> ./unpack.sh -domain=ORACLE_BASE/admin/<domain_name>/msserver/<domain_name> -
template=osbdomaintemplate.jar
```

3.6. OSBHOST1で管理サーバー用のboot.propertiesの作成

OSBHOST1で管理サーバー用のboot.propertiesファイルを作成します。これにより、管理者のユーザー名およびパスワードを入力せずに、管理サーバーを起動することができます。

管理サーバーで以下の操作を行います。

1. 以下のディレクトリ構造を作成します。

```
mkdir ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers
mkdir ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers/AdminServer
mkdir ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers/AdminServer/security
```

2. テキスト・エディタを使用して、前の手順で作成した最後のディレクトリ (ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/servers/AdminServer/security) に boot.propertiesというファイルを作成し、以下の内容を入力します。

username=<adminuser>

password=<password>

注: 管理サーバーを起動すると、ファイル内のユーザー名とパスワードエントリが暗号化されます。項3.7「OSBHOST1で管理サーバーの起動」で、管理サーバーを起動します。

セキュリティのために、ファイル内のエントリが暗号化されていない時間を最小限にします。ファイルを編集した後、エントリを暗号化するために、できるだけ早くサーバーを起動します。

3.7. OSBHOST1で管理サーバーの起動

以下のコマンドを実行して、管理サーバーを起動します。

```
OSBHOST1> cd ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/bin
OSBHOST1> ./startWebLogic.sh
```

3.8. 管理サーバーの検証

以下の手順を実行して、管理サーバーが正しく構成されていることを確認します。

1. ブラウザで、<http://OSBHOST1VHN1:7001/console>に移動します。

2. 管理者としてログインします。
3. 管理対象サーバーWLS_OSB1とWLS_OSB2が表示されていることを確認します。
4. OSB_Clusterクラスタが表示されていることを確認します。

3.9. 管理対象サーバーWLS_OSB1のホスト名検証の無効化

管理サーバーでノードを認証するための証明書を設定していない場合、この項が必要となります(項4「ノードマネージャの設定」を参照してください)。サーバー証明書を設定していない場合、WebLogicサーバーを管理する際に、エラーが表示されます。エラーを回避するために、設定およびトポロジを検証する時のホスト名検証を無効にします。項4で説明されているように設定が完了したら、再びホスト名検証を有効にします。

以下の手順を実行して、ホスト名検証を無効にします。

1. 「Oracle WebLogic Server Administration Console」にログインします。
2. 「Lock and Edit」をクリックします。
3. 「Domain Structure」ウィンドウで、「Environment」ノードを展開します。
4. 「Servers」をクリックします。「Summary of Servers」ページが表示されます。
5. テーブルの「Names」列から「WLS_OSB1」(ハイパーリンクとして表示されている)を選択します。「Settings」ページが表示されます。
6. 「SSL」タブを選択します。
7. 「Advanced」セクションを展開します。
8. 「Hostname Verification」を「None」に設定します。
9. 「Save」をクリックします。
10. 管理対象サーバーWLS_OSB2に対して上記の手順を繰り返します。
11. 保存して変更を有効にします。

3.9.1. ノードマネージャのプロパティの変更

この手順は、SSLが有効になっている、キーストアが項4.2「OSBHOST1でのノードマネージャ用のホスト検証証明書の有効化」のように設定されていない、またセキュリティがOracleデータベースを使用するように設定されていない場合必要となります。

ORACLE_BASE/wlserver_10.3/common/bin/commEnv.sh を開きます。

以下のコマンドを追加します。

```
export JAVA_OPTIONS="$JAVA_OPTIONS -Djava.security.egd=file:/dev/./urandom"
export JAVA_OPTIONS="$JAVA_OPTIONS -Dweblogic.nodemanager.sslHostNameVerificationEnabled=false"
```

ファイルを保存します。

3.10. OSBHOST1とOSBHOST2でノードマネージャの起動または再起動

OSBHOST1とOSBHOST2で、ノードマネージャを起動または再起動するには、以下の手順を実行します。

1. ノードマネージャを停止します(起動している場合)。

例:

```
OSBHOST1> kill -9 <pid> (pid => ノードマネージャのプロセスID)
```

2. ノードマネージャを起動します。

例:

```
OSBHOST1> cd /ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
./startNodeManager.sh
```

3.11. ドメインの変更を管理対象サーバーのドメイン・ディレクトリに反映

スタート・スクリプトおよびクラス・パス設定を管理サーバーのドメイン・ディレクトリから管理対象サーバーのドメイン・ディレクトリに反映するには、以下の手順を実行します。

1. 管理対象サーバーのドメイン・ディレクトリおよびアプリケーション・ディレクトリをコピーします。
2. 以下のコマンドを実行して(一行で)、ディレクトリに移動します。

```
OSBHOST1> mv ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>  
ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>backup
```

3. OSBHOST1で以下のようにpackコマンドを実行して、テンプレート・パックを作成します。

```
OSBHOST1> cd ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin  
OSBHOST1> ./pack.sh -managed=true  
-domain=ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>  
-template=osbdomaintemplateExtOSB.jar -template_name=osb_domain_templateExtOSB
```

4. OSBHOST1で以下のようにunpackコマンドを実行して、テンプレートを管理対象サーバーに解凍します。

```
OSBHOST1> ./unpack.sh  
-domain=ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>/ -  
template=osbdomaintemplateExtOSB.jar -  
app_dir=ORACLE_BASE/admin/<domain_name>/msserver/apps
```

3.12. OSB_Clusterのクラスタ・アドレスの設定

1. 「Oracle WebLogic Server Administration Console」にログインし、「Environment」→「cluster」に移動します。
2. 「OSB_Cluster」をクリックします。
3. 「Cluster Address」で「<OSBHOST1>:7001,<OSBHOST2>:7001」と入力します。
4. 項3.7「OSBHOST1で管理サーバーの起動」を参照して、管理サーバーを再起動します。

3.13. OSBHOST1で管理対象サーバーWLS_OSB1の起動

OSBHOST1で管理対象サーバーWLS_OSB1を起動するには、以下の手順を実行します。

1. 以下のURLを入力して、「Oracle WebLogic Server Administration Console」に登録します。
`http://OSBHOST1VHN1:7001/console`
2. 「Servers」をクリックします。
3. 「Control」タブを開きます。
4. 「WLS_OSB1」を選択します。
5. 「Start」をクリックします。

注: OSBHOST1VHN1は、仮想ホスト名で、管理サーバーがリスニングしている仮想IPにマップされます。

3.14. 管理対象サーバーWLS_OSB1の検証

管理対象サーバーWLS_OSB1を検証するには、以下の手順を実行します。

「Oracle WebLogic Server Administration Console」でサーバーのステータスが「Running」であることを確認します。「Starting」または「Resuming」の場合、「Running」に変更するまで待ちます。別のステータス（「Admin」または「Failed」）が表示されている場合、サーバーの出力ログ・ファイルを確認してください。

3.15. unpackユーティリティを使用して、OSBHOST2へドメイン構成を反映

ドメイン構成を反映するには、以下の手順を実行します。

1. OSBHOST1で以下のコマンドを実行して、前の手順で作成したテンプレート・ファイルをOSBHOST2にコピーします。

```
OSBHOST1> cd /ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin
OSBHOST1> scp osbdomaintemplateExtOSB.jar
oracle@node2:ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin
```

2. OSBHOST2で以下のunpackコマンドを実行して、テンプレートを解凍します。

注:OSBHOST2の既存のドメイン・ディレクトリを削除する必要があります。削除しないと、操作は失敗になります。unpackコマンドは、新しいドメイン・ディレクトリを作成します。

OSBHOST2で動作している管理対象サーバーに対して行った構成の変更を再び行う必要があります。

```
OSBHOST2> cd ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin
OSBHOST2> ./unpack.sh
-domain=ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>/
-template=osbdomaintemplateExtOSB.jar
-app_dir=ORACLE_BASE/admin/<domain_dir>/msserver/apps
```

OSBHOST2ノードマネージャを管理サーバーへ登録するには、以下の手順を実行します。

```
OSBHOST2>cd ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>/bin
OSBHOST2>Source setDomainEnv.cmd
OSBHOST2>java weblogic.WLST
OSBHOST2>connect('weblogic','welcome1','t3://osbhost1:7001')
OSBHOST2>nmEnroll('/ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>','/ORACLE_BASE/wlserver_10.3/common/nodemanager')
```

3.16. 管理対象サーバーWLS_OSB2のホスト名検証の無効化

OSBHOST2で項3.9「管理対象サーバーWLS_OSB1のホスト名検証の無効化」の手順を実行します。

3.17. OSBHOST2でノードマネージャの再起動

OSBHOST2で項3.10「OSBHOST1とOSBHOST2でノードマネージャの起動または再起動」の手順を実行します。

3.18. 管理対象サーバーWLS_OSB2の起動および検証

以下の手順を実行して、管理対象サーバーWLS_OSB2を起動し、正しく構成されていることを確認します。

1. 「Oracle WebLogic Server Administration Console」を使用して、管理対象サーバーWLS_OSB2を起動します。
2. サーバーのステータスが「Running」と表示されていることを確認します。「Starting」または「Resuming」と表示されている場合、「Started」に変更するまで待ちます。別のステータス（「Admin」または「Failed」）が表示される場合、サーバーの出力ファイルを確認してください。

3.19. 共有JMS永続ストアの構成

永続ストアの場所を両方のノードから接続できるディレクトリに設定します。詳細については、項2.4「共有ストレージおよび推奨ディレクトリ構造」を参照してください。以下の手順を実行して、すべての永続ストアがこの共有ディレクトリを使用するように変更します。

1. 「Oracle WebLogic Server Administration Console」にログインします。
2. 「Domain Structure」ウィンドウで、「Services」ノードを展開し、「Persistence Stores」ノードをクリックします。「Summary of Persistence store」ページが表示されます。
3. テーブルの「Name」列から「persistence store」(ハイパーリンクで表示されている)を選択します。「Setting page for the persistence store」が表示されます。
4. 「Configuration」タブで、「Directory」フィールドにクラスタ内のサーバーが使用する永続ストレージ・ソリューション (NASやSANなど) 上の場所を入力します。この場所を指定すると、JMSメッセージの送信が保留されることができません。場所は、以下のようなディレクトリ構造に従う必要があります。

ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/jms

注: WLS_OSB1とWLS_OSB2両方がこのディレクトリに接続できる必要があります。サーバーを起動する前に、このディレクトリが存在している必要があります。

5. 「Save」をクリックして、変更を有効にします。
6. サーバーを再起動して、永続ストアの変更を有効にします。

3.20. トランザクション・リカバリためのデフォルト永続ストアの構成

各サーバーには、コミットされたトランザクションの情報を格納するトランザクション・ファイルがあります。コミットされたトランザクションは、完了されていないサーバーによって調整されます。WebLogicサーバーはトランザクション・ファイルを使用して、システムのクラッシュやネットワーク障害からサーバーを復旧します。クラスタ内のサーバーでトランザクション・リカバリ・サービスの移行機能を利用するために、トランザクション・ログをサーバーおよびサーバーのバックアップから接続できる場所に格納する必要があります。

注: トランザクション・ログの場所をデュアルポートSCSIディスクまたはストレージ・エリア・ネットワーク (SAN) 上の場所に設定することをお勧めします。

デフォルト永続ストアの場所を設定するには、以下の手順を実行します。

1. 「Oracle WebLogic Server Administration Console」にログインします。
2. 「Domain Structure」ウィンドウで、「Environment」ノードを展開し、「Servers」ノードをクリックします。「Summary of Servers」ページが表示されます。
3. テーブルの「Name」列からサーバーの名前 (ハイパーリンクで表示されている) をクリックします。選択したサーバーの設定ページが表示され、デフォルトでは「Configuration」タブが表示されています。
4. 「Services」タブをクリックします。
5. 「Default Store」セクションで、デフォルト永続ストアがデータファイルを格納するフォルダのパスを入力します。パスのディレクトリ構造は以下のようになります。

ORACLE_BASE/admin/<domain_name>/<osb_cluster_name>/tlogs

6. 「Save」をクリックします。

注: トランザクション・リカバリ・サービスを有効するには、クラスタ内のサーバーが使用できる永続ストレージ・ソリューション上の場所を指定します。WLS_OSB1とWLS_OSB2両方がこのディレクトリに接続できる必要があります。サーバーを起動する前に、このディレクトリが存在している必要があります。

3.21. Oracle RAC DatabaseのJDBCデータソースの構成

WebLogicマルチプール・データソースを設定して、Oracle RAC Databaseに接続します。RACのすべてのインスタンスが稼働していて、接続できることを確認します。マルチデータソースcgDataSource、wlsjmsrpDataSourceおよびcgDataSource-nonXAを作成します。各マルチデータソースを作成する時、RACの各インスタンスを指定する必要があります。

以下の設定で、データソースcgDataSourceとwlsjmsrpDataSourceを作成します。

- ✧ フォーマット<MultiDS>-rac0,<MultiDS>-rac1などを使用して、マルチデータベースの名前を付けます。
- ✧ 「Oracle's Driver(Thin XA)Version 9.0.1,9.0.2,10,11」を使用します。
- ✧ 「☐Supports Global Transactions」を選択し、「Two-Phase Commit」を選択します。
- ✧ データベースの「Service Name」を指定します。
- ✧ データソースをOSBクラスタにターゲットします。

以下の設定で、データソースcgDataSource-nonXAを作成します。

- ✧ フォーマット<MultiDS>-rac0,<MultiDS>-rac1などを使用して、マルチデータソースの名前を付けます。
- ✧ 「Oracle's Driver (Thin) Version 9.0.1, 9.2.0, 10, 11」を使用します。
- ✧ 「Supports Global Transactions」を選択しません。
- ✧ データベースの「Service Name」を指定します。
- ✧ データソースをOSBクラスタにターゲットします。

注:マルチデータソースの名前は、スタンドアロンのJDBCデータソースと同じにする必要があります。同じ名前を持つマルチデータソースを作成するために、OSBサーバーをシャットダウンし、個々のJDBCデータソースを削除します。

マルチデータソースを作成します。マルチデータソースを作成するには、各データソースに対して、以下の手順を実行します。

1. 「Oracle WebLogic Server Administration Console」の「Domain Structure」ウィンドウで、「Services」ノードを展開し、「JDBC」ノードを展開します。
2. 「Multi Data Source」をクリックします。「Summary of JDBC Multi Data Source」ページが表示されます。
3. 「Lock and Edit」をクリックします。
4. 「New」をクリックします。「Create a New JDBC Multi Data Source」ページが表示されます。
5. 「New」をクリックします。
6. 「Name」として、<data_source> (例:wlsjmsrpDataSource)を入力します。
7. 「JNDI」として、<data_source> (例:wlsjmsrpDataSource)を入力します。
8. 「algorithm」として「Failover」を選択します。これはデフォルトの設定です。
9. 「Next」をクリックします。
10. ターゲットとして、「OSB_Cluster」を選択します。
11. 「Next」をクリックします。
12. データソースcgDataSource-nonXAを作成する時、「non-XA driver」を選択します。データソースcgDataSourceとwlsjmsrpDataSourceを作成する時、「XA driver」を選択します。
13. 「Next」をクリックします。
14. 「Create New Data Source」をクリックします。
15. 「Name」として<data_source>-rac0を入力します。「JNDI Name」に個別のJNDI(<data_source>-rac0)を入力します。
16. 「Next」をクリックします。
17. データソースcgDataSource-nonXAを作成する時、「Support Global Transactions」を選択しません。データソースcgDataSourceとwlsjmsrpDataSourceを作成する時、「Support Global Transactions」を選択します。
18. 「Next」をクリックします。
19. OSBスキーマの「Service Name」、「Database Name」、「Host Port」、「Password」を入力します。
20. 「Next」をクリックします。
21. 「Test Configuration」をクリックして、接続を確認します。
22. データソースを「OSB_Cluster」にターゲットします。
23. データソースを選択して、右のスクリーンに追加します。
24. 「Create a New Data Source」をクリックして、RAC Databaseの一番目のインスタンスのデータソースを作成します。作成したデータソースを「OSB_Cluster」にターゲットします。この操作を繰り返して、RAC Databaseの二番目のインスタンスのデータソースを作成します。
25. 二番目のデータソースをマルチデータソースに追加します。

26. 「Activate Changes」をクリックします。
27. 管理サーバーおよび管理対象サーバーを再起動します。

3.22. 手動で管理サーバーをOSBHOST2にフェイルオーバー

ノードで障害が発生した場合、管理サーバーを別のノードにフェイルオーバーすることができます。この項では、管理サーバーをOSBHOST1からOSBHOST2にフェイルオーバーする方法について説明します。

前提:

- ◇ 管理サーバーがOSBHOST1VHN1のみをリスニングするように設定されています。
- ◇ これらの手順は、2つのノードが個別のドメイン・ディレクトリを使用することを想定します。ドメイン・ディレクトリはローカル・ストレージまたは共有ストレージの個別のボリュームに存在しています。
- ◇ 管理サーバーはOSBHOST1からOSBHOST2にフェイルオーバーされます。2つのノードは以下のIPアドレスを持っています。
 - OSBHOST1: 100.200.140.165
 - OSBHOST2: 100.200.140.205
 - VIPHOST1: 100.200.140.206. これは管理サーバーの仮想IPで、ethX:Yにアサインされ、OSBHOST1およびOSBHOST2で使用されています。
- ◇ OSBHOST1で実行されている管理サーバーのドメイン・ディレクトリは、共有ストレージにあり、OSBHOST2からもマウントされています。

次の手順では、管理サーバーを異なるノード(OSBHOST2)にフェイルオーバーする方法を示していますが、管理サーバーはまだ同じWebLogicサーバーマシン(物理マシンではなく、論理マシンです)を使用します。

1. 管理サーバーを停止します。
2. IPアドレスを2番目のノードに移行します。

OSBHOST1でrootとして以下のコマンドを実行します。(「X:Y」がOSBHOST1VHN1によって使用されているインタフェースです)

```
OSBHOST1> /sbin/ifconfig ethX:Y down
```

3. OSBHOST2で以下のコマンドを実行します。

```
OSBHOST2> /sbin/ifconfig <interface:index> <IP_Address> netmask <netmask>
```

例:

```
/sbin/ifconfig eth0:1 10.0.0.1 netmask 255.255.255.0
```

注: 使用するネットマスクとインタフェースがOSBHOST2の使用可能なネットワーク構成と一致することを確認します。

4. arpingを使用してルーティング・テーブルを更新します。

例:

```
OSBHOST2> /sbin/arping -b -A -c 3 -I eth0 10.0.0.1
```

5. OSBHOST2で管理サーバーを起動します。

```
OSBHOST2> cd ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/bin  
OSBHOST2> ./startWebLogic.sh
```

6. 以下のように、OSBHOST2から管理サーバーに接続できることを確認します。
 - ✧ 以下のURLを入力して、「Oracle WebLogic Server Administration Console」に接続できることを確認します。
`http://OSBHOST1VHN1:7001/console`
 - ✧ 以下のURLを入力し、「Oracle Enterprise Manager」に接続し、コンポーネントのステータスを検証できることを確認します。
`http://OSBHOST1VHN1:7001/em`

3.23. 管理サーバーをOSBHOST1にフェイルバック

この項では、管理サーバーのフェイルバック、つまりOSBHOST2で管理サーバーを停止し、OSBHOST1で実行できるかどうか確認します。これを実現するには、以下のようにOSBHOST1VHN1をOSBHOST1に戻します。

1. OSBHOST2で以下のコマンドを実行します。

```
OSBHOST2> /sbin/ifconfig ethZ:N down
```

2. OSBHOST1で以下のコマンドを実行します。

```
OSBHOST1> /sbin/ifconfig ethX:Y 100.200.140.206 netmask 255.255.255.0
```

注:使用するネットマスクとインタフェースがOSBHOST1の使用可能なネットワーク構成と一致することを確認します。

3. arpingを使用して、ルーティング・テーブルを更新します。OSBHOST1で以下のコマンドを実行します。

```
OSBHOST1> /sbin/arping -b -A -c 3 -I ethZ 100.200.140.206
```

4. OSBHOST1で管理サーバーを起動します。

```
OSBHOST1> cd ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/bin
OSBHOST1> ./startWebLogic.sh
```

5. 以下のURLを入力して、「Oracle WebLogic Server Administration Console」に接続できることを確認します。
`http://OSBHOST1VHN1:7001/console`
6. 以下のURLを入力し、「Oracle Enterprise Manager」に接続し、コンポーネントのステータスを検証できることを確認します。
`http://OSBHOST1VHN1:7001/em`

3.24. インストールのバックアップ

ドメイン構成をバックアップします。バックアップする前に、サーバーを停止する必要があります。設定ファイルは以下のディレクトリに存在しています。

ディレクトリ:

`ORACLE_BASE/admin/<domain_name>`

以下のコマンドを実行して、ドメイン構成をバックアップします。

```
OSBHOST1> tar -cvpf mydomainback.tar ORACLE_BASE/admin/<domain_name>
```


4. ノードマネージャの設定

この項では、ノードマネージャと管理サーバー間の通信に必要なホスト名検証を有効するために、ノードマネージャの設定方法について説明します。異なるアドレスのサーバーが管理サーバーと通信するために、証明書を必要となります。ここでは、ホスト名検証のためにOSBHOST1とOSBHOST2の証明書の設定手順について、説明します。

4.1. ノードマネージャについて

ノードマネージャを使用して、管理サーバーおよび管理対象サーバーを起動、停止することができます。

4.2. OSBHOST1でのノードマネージャ用のホスト検証証明書の有効化

以下の手順を実行して、ノードマネージャと管理サーバー間の通信に必要なホスト名検証証明書を設定します。

手順1: `utils.CertGen`ユーティリティを使用して、自己署名証明書の生成

手順2: `utils.ImportPrivateKey`ユーティリティを使用して、IDキーストアの作成

手順3: `keytool`ユーティリティを使用して、信頼キーストアの作成

手順4: カスタム・キーストアを使用するためのノードマネージャの構成

4.2.1. `utils.CertGen`ユーティリティを使用して、自己署名証明書の生成

以下の手順を実行して、OSBHOST1.mycompany.comで自己署名証明書を作成します。証明書はネットワークを使用して作成されます。信頼CA証明書を使用して証明書を作成する方法については、「Oracle Fusion Middleware Oracle WebLogic Serverのセキュリティ」を参照してください。

1. 以下のスクリプトを実行して、環境を設定します。

```
ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin/setWLSEnv.sh
```

Bourneシェルでは、以下のコマンドを実行します。

```
OSBHOST1> ./setWLSEnv.sh
```

以下のコマンドを実行して、CLASSPATH環境変数が設定されていることを確認します。

```
OSBHOST1> echo $CLASSPATH
```

2. 証明書用のユーザー定義のディレクトリを作成します。例えば、ORACLE_BASE/product/osbmwディレクトリの下にcertsというディレクトリを作成します。証明書はWLSドメイン間で共有できることに注意してください。

```
OSBHOST1> cd ORACLE_BASE/product/osbmw
OSBHOST1> mkdir certs
```

3. ユーザー定義のディレクトリに移動します。

```
OSBHOST1> cd certs
```

4. ユーザー定義のディレクトリから`utils.CertGen`ツールを実行して、OSBHOST1とOSBHOST2VHN1両方の証明書を作成します。

構文:

```
java utils.CertGen <key_passphrase> <cert_file_name> <key_file_name> [export | domestic] [hostname]
```

例:

```
OSBHOST1> java utils.CertGen welcome1 OSBHOST1_cert OSBHOST1_key domestic
OSBHOST1.mycompany.com
OSBHOST1> java utils.CertGen welcome1 VIPHOST1_cert VIPHOST1_key domestic
OSBHOST1VHN1.mycompany.com
```

4.2.2. **utils.ImportPrivateKey**ユーティリティを使用して、IDキーストアの作成

以下の手順を実行して、OSBHOST1.mycompany.comでアイデンティティ・キーストアを作成します。

1. **utils.ImportPrivateKey**ユーティリティを使用して、**appIdentityKeyStore**という新しいアイデンティティ・キーストアを作成します。

証明書と同じディレクトリ(ORACLE_BASE/product/osbmw/certs)の下でこのキーストアを作成します。

注:**utils.ImportPrivateKey**ユーティリティを使用して、証明書および対応するキーをキーストアにインポートします。

OSBHOST1とVIPHOST1両方の証明書および秘密鍵をアイデンティティ・ストアにインポートします。各証明書/キーペアは異なる別名を使用します。

構文:

```
java utils.ImportPrivateKey <keystore_file> <keystore_password>
<certificate_alias_to_use> <private_key_passphrase> <certificate_file>
<private_key_file> [<keystore_type>]
```

例:

```
OSBHOST1> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1
appIdentity1 welcome1 ORACLE_BASE/product/osbmw/certs/OSBHOST1_cert.pem
ORACLE_BASE/product/osbmw/certs/OSBHOST1_key.pem
OSBHOST1> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1 appIdentity2 welcome1
ORACLE_BASE/product/osbmw/certs/VIPHOST1_cert.pem
ORACLE_BASE/product/osbmw/certs/VIPHOST1_key.pem
```

4.2.3. **keytool**ユーティリティを使用して、信頼キーストアの作成

以下の手順を実行して、OSBHOST1.mycompany.comで信頼キーストアを作成します。

1. 標準のJavaキーストアには必要なルートCA証明書が含まれているので、標準のJavaキーストアをコピーして新しい信頼キーストアを作成します。直接標準のJava信頼キーストアを編集することは推奨しません。WL_HOME/server/libディレクトリにある標準のJavaキーストアのCA証明書を証明書と同じディレクトリにコピーします。

例:

```
cp WL_HOME/server/lib/cacerts
ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/certs/appTrustKeyStore.jks
```

2. 標準のJavaキーストアのデフォルトのパスワードは「changeit」です。デフォルトのパスワードを常に変更することをお勧めします。**keytool**ユーティリティを使用して、パスワードを変更します。構文は以下になります。

構文:

```
keytool -storepasswd -new <NewPassword> -keystore <TrustKeyStore> -storepass <Original Password>
```

例:

```
keytool -storepasswd -new welcome1 -keystore appTrustKeyStore.jks -storepass changeit
```

3. CA証明書CertGenCA.derはutilsによって生成されたすべての証明書を署名するために使用されます。CertGenツールは、WL_HOME/server/libディレクトリに配置されています。keytoolユーティリティを使用して、CA証明書をappTrustKeyStoreにインポートする必要があります。

構文:

```
keytool -import -v -noprompt -trustcacerts -alias <AliasName> -file <CAFileLocation> -keystore  
<KeyStoreLocation> -storepass <KeyStorePassword>
```

例:

```
keytool -import -v -noprompt -trustcacerts -alias clientCACert -file $WL_HOME/server/lib/CertGenCA.der  
-keystore appTrustKeyStore.jks -storepass welcome1
```

4.2.4. カスタム・キーストアを使用するためのノードマネージャの構成

カスタム・キーストアを使用するために、ORACLE_BASE/product/osbmw/wlserver_10.3/common/nodemanagerにあるnodemanager.propertiesファイルの最後に以下の行を追加します。

```
KeyStores=CustomIdentityAndCustomTrust  
CustomIdentityKeyStoreFileName=<Identity KeyStore>  
CustomIdentityKeyStorePassPhrase=<Identity KeyStore Passwd>  
CustomIdentityAlias=<Identity Key Store Alias>  
CustomIdentityPrivateKeyPassPhrase=<Private Key used when creating Certificate>
```

各ノードでCustomIdentityAliasの正しい値を使用します。例えば、OSBHOST1では「appIdentity1」を使用し、VIPHOST1では「appIdentity2」を使用します。

ノード1の例:

```
KeyStores=CustomIdentityAndCustomTrust  
CustomIdentityKeyStoreFileName=ORACLE_BASE/product/osbmw/certs/  
appIdentityKeyStore.jks  
CustomIdentityKeyStorePassPhrase=welcome1  
CustomIdentityAlias=appIdentity1  
CustomIdentityPrivateKeyPassPhrase=welcome1
```

項4.3「OSBHOST1でのノードマネージャの起動」で説明したようにノードマネージャを起動すると、nodemanager.propertiesファイルのパスフレーズのエントリが暗号化されます。セキュリティのために、nodemanager.propertiesファイル内のエントリが暗号化されていない時間を最小限にします。ファイルを編集した後、エントリを暗号化するために、できるだけ早くサーバーを起動します。

共通または共有ストレージを使用してOSBMW_HOMEをインストールした場合、ノードマネージャは、同じ基本構成(nodemanager.properties)を使用して、異なるノードから開始されます。この場合、バイナリを共有するすべてのノードの証明書をappIdentityKeyStore.jksキーストアに追加する必要があります。これを行うには、新しいノードの証明書を作成し、上記のようにappIdentityKeyStore.jksにインポートします。証明書がキーストアで使用可能になると、各ノードマネージャが管理サーバーに正しい証明書を送信するために、個別のアイデンティティ別名を指定する必要があります。これを行うには、ノードマネージャを起動する前に、以下のように環境変数を設定します。

```
OSBHOST1>cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOST1>export JAVA_OPTIONS=-DCustomIdentityAlias=appIdentityOSBHOST1
OSBHOSTn>cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOSTn>export JAVA_OPTIONS=-DCustomIdentityAlias=appIdentityOSBHOSTn
```

4.3. OSBHOST1でのノードマネージャの起動

OSBHOST1で以下のコマンドを実行して、ノードマネージャを起動します。

```
OSBHOST1> cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOST1> ./startNodeManager.sh
```

4.4. OSBHOST2でのノードマネージャ用のホスト検証証明書の有効化

以下の手順を実行して、ノードマネージャと管理サーバー間の通信に必要なSSLを設定します。

- 手順1: `utils.CertGen`ユーティリティを使用して、自己署名証明書の生成
- 手順2: `utils.ImportPrivateKey`ユーティリティを使用して、IDキースタアの作成
- 手順3: `keytool`ユーティリティを使用して、信頼キースタアの作成
- 手順4: カスタム・キースタアを使用するためのノードマネージャの構成

4.4.1. `utils.CertGen`ユーティリティを使用して、自己署名証明書の生成

以下の手順を実行して、OSBHOST2.mycompany.comで自己署名証明書を作成します。証明書はネットワークを使用して作成されます。

1. `ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin/setWLSEnv.sh`スクリプトを実行して、環境を設定します。

Bourneシェルでは、以下のコマンドを実行します。

```
OSBHOST2> ./setWLSEnv.sh
```

`CLASSPATH`環境変数が設定されていることを確認します。

```
OSBHOST2> echo $CLASSPATH
```

2. 証明書用のユーザー定義のディレクトリを作成します。例えば、`ORACLE_BASE/product/osbmw`ディレクトリの下に`certs`というディレクトリを作成します。証明書はWLSドメイン間で共有できることに注意してください。

```
OSBHOST2> cd ORACLE_BASE/product/osbmw
OSBHOST2> mkdir certs
```

3. ユーザー定義のディレクトリに移動します。

```
OSBHOST2> cd certs
```

4. ユーザー定義のディレクトリからutils.CertGenツールを実行して、OSBHOST2とVIPHOST1両方の証明書を作成します。

構文:

```
java utils.CertGen <key_passphrase> <cert_file_name> <key_file_name> [export | domestic] [hostname]
```

例:

```
OSBHOST2> java utils.CertGen welcome1 OSBHOST2_cert OSBHOST2_key domestic
OSBHOST2.mycompany.com

OSBHOST2> java utils.CertGen welcome1 VIPHOST1_cert VIPHOST1_key domestic
OSBHOST1VHN1.mycompany.com
```

4.4.2. utils.ImportPrivateKeyユーティリティを使用して、IDキーストアの作成

1. utils.ImportPrivateKeyユーティリティを使用して、appIdentityKeyStoreという新しいアイデンティティ・キーストアを作成します。

証明書と同じディレクトリ(ORACLE_BASE/product/osbmw/certs)の下でこのキーストアを作成します。

utils.ImportPrivateKeyユーティリティを使用して、証明書および対応するキーをキーストアにインポートします。

OSBHOST2とVIPHOST1両方の証明書および秘密鍵をアイデンティティ・ストアにインポートします。各証明書/キーペアは異なる別名を使用することを確認します。

構文:

```
java utils.ImportPrivateKey <keystore_file> <keystore_password> <certificate_alias_to_use>
<private_key_passphrase> <certificate_file> <private_key_file> [<keystore_type>]
```

例:

```
OSBHOST2> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1 appIdentity1 welcome1
ORACLE_BASE/product/osbmw/certs/OSBHOST2_cert.pem
ORACLE_BASE/product/osbmw/certs/OSBHOST2_key.pem

OSBHOST2> java utils.ImportPrivateKey appIdentityKeyStore.jks welcome1 appIdentity2 welcome1
ORACLE_BASE/product/osbmw/certs/VIPHOST1_cert.pem
ORACLE_BASE/product/osbmw/certs/VIPHOST1_key.pem
```

4.4.3. keytoolユーティリティを使用して、信頼キーストアの作成

以下の手順を実行して、OSBHOST2.mycompany.comで信頼キーストアを作成します。

1. 標準のJavaキーストアには必要なルートCA証明書がほとんど含まれているので、標準のJavaキーストアをコピーして新しい信頼キーストアを作成します。直接標準のJava信頼キーストアを編集することを推奨しません。WL_HOME/server/libディレクトリにある標準のJavaキーストアのCA証明書を証明書と同じディレクトリにコピーします。

例:

```
cp WL_HOME/server/lib/cacerts
ORACLE_BASE/admin/<domain_name>/aserver/<domain_name>/certs/appTrustKeyStore.jks
```

2. 標準のJavaキーストアのデフォルトのパスワードは「changeit」です。デフォルトのパスワードを常に変更することをお勧めします。keytoolユーティリティを使用して、パスワードを変更します。

構文:

```
keytool -storepasswd -new <NewPassword> -keystore <TrustKeyStore> -storepass <Original Password>
```

例:

```
keytool -storepasswd -new welcome1 -keystore appTrustKeyStore.jks -storepass changeit
```

3. CA証明書CertGenCA.derはutilsによって生成されたすべての証明書を署名するために使用されます。CertGenツールは、WL_HOME/server/libディレクトリに配置されています。keytoolユーティリティを使用して、CA証明書をappTrustKeyStoreにインポートする必要があります。

構文:

```
keytool -import -v -noprompt -trustcacerts -alias <AliasName> -file <CAFileLocation> -keystore  
<KeyStoreLocation> -storepass <KeyStorePassword>
```

例:

```
keytool -import -v -noprompt -trustcacerts -alias clientCACert -file $WL_HOME/server/lib/CertGenCA.der  
-keystore appTrustKeyStore.jks -storepass welcome1
```

4.4.4. カスタム・キーストアを使用するためのノードマネージャの構成

以下の手順を実行して、ノードマネージャがカスタム・キーストアを使用するように設定します。

1. ORACLE_BASE/product/osbmw/wlserver_10.3/common/nodemanagerにあるnodemanager.propertiesファイルの最後に以下の行を追加します。

```
KeyStores=CustomIdentityAndCustomTrust  
CustomIdentityKeyStoreFileName=<Identity KeyStore>  
CustomIdentityKeyStorePassPhrase=<Identity KeyStore Passwd>  
CustomIdentityAlias=<Identity Key Store Alias>  
CustomIdentityPrivateKeyPassPhrase=<Private Key used when creating Certificate>
```

各ノードでCustomIdentityAliasの正しい値を使用します。例えば、OSBHOST2では「appIdentity2」を使用し、VIPHOST1では「appIdentity2」を使用します。

ノード1の例:

```
KeyStores=CustomIdentityAndCustomTrust  
CustomIdentityKeyStoreFileName=ORACLE_BASE/product/osbmw/certs/appIdentityKeyStore.jks  
CustomIdentityKeyStorePassPhrase=welcome1  
CustomIdentityAlias=appIdentity1  
CustomIdentityPrivateKeyPassPhrase=welcome1
```

注: 項4.5「OSBHOST2でのノードマネージャの起動」で説明したようにノードマネージャを起動すると、nodemanager.propertiesファイルのパスフレーズのエンTRIESが暗号化されます。

セキュリティのために、nodemanager.propertiesファイル内のENTRIESが暗号化されていない時間を最小限にします。ファイルを編集した後、ENTRIESを暗号化するために、できるだけ早くサーバーを起動します。

共通または共有ストレージを使用してOSBMW_HOMEをインストールした場合、ノードマネージャは、同じ基本構成(nodemanager.properties)を使用して、異なるノードから開始されます。この場合、バイナリを共有するすべてのノードの証明書をappIdentityKeyStore.jksキーストアに追加する必要があります。これを行うには、新しいノードの証明書

を作成し、上記のようにappIdentityKeyStore.jksにインポートします。証明書がキーストアで使用可能になると、各ノードマネージャが管理サーバーに正しい証明書を送信するために、個別のアイデンティティ別名を指定する必要があります。これを行うには、ノードマネージャを起動する前に、環境変数を設定します。

4.5. OSBHOST2でのノードマネージャの起動

OSBHOST2で以下のコマンドを実行して、ノードマネージャを起動します。

```
OSBHOST2> cd ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin
OSBHOST2> ./startNodeManager.sh
```

5. サーバーの移行

エンタープライズ・トポロジでは、管理対象サーバーWLS_OSB1とWLS_OSB2に対して、サーバーの移行を設定する必要があります。管理対象サーバーWLS_OSB1は、障害が発生した場合、OSBHOST2で再起動されるように設定されます。管理対象サーバーWLS_OSB2は、障害が発生した場合、OSBHOST1で再起動されるように設定されます。この構成では、WLS_OSB1とWLS_OSB2は、WLSサーバーの移行によってフェイルオーバーされる特定の浮動IPアドレスをリスニングします。管理対象サーバーWLS_OSBnに対してサーバー移行を構成するには、以下の手順を実行します。

手順1: サーバー移行用のユーザー、表領域およびテーブルの作成

手順2: 「Oracle WebLogic Server Administration Console」を使用して、マルチデータソースの作成

手順3: OSBHOST1とOSBHOST2と管理サーバーの間でホスト名検証証明書の有効化

手順4: ノードマネージャのプロパティ・ファイルの編集

手順5: wlsifconfig.shスクリプトの環境変数およびスーパーユーザーの権限の設定

手順6: サーバー移行ターゲットの構成

手順7: サーバー移行のテスト

5.1. サーバー移行用のユーザー、表領域およびテーブルの作成

まず、サーバー移行用ユーザー、表領域およびテーブルを作成します。

1. leasingと呼ばれる表領域を作成します。ユーザーSYSDBAとしてSQL*Plusにログインし、以下のコマンドを実行します。

```
SQL> create tablespace leasing logging datafile 'DB_HOME/oradata/orcl/leasing.dbf' size 32m autoextend
on next 32m maxsize 2048m extent management local;
```

2. leasingというユーザーを作成し、leasing表領域に割り当てます。

```
SQL> create user leasing identified by welcome1;
SQL> grant create table to leasing;
SQL> grant create session to leasing;
SQL> alter user leasing default tablespace leasing;
SQL> alter user leasing quota unlimited on LEASING;
```

3. leasing.ddlスクリプトを使用して、テーブルleasingを作成します。
以下のディレクトリに存在するleasing.ddlファイルをデータベース・ノードにコピーします。

ORACLE_BASE/product/osbmw/wlserver_10.3/server/db/oracle/817

または

ORACLE_BASE/product/osbmw/wlserver_10.3/server/db/oracle/920

ユーザーleasingとして、データベースに接続します。

SQL*Plusでleasing.ddlスクリプトを実行します。

```
SQL> @copy_location/leasing.ddl;
```

5.2. マルチデータソースの作成

ここでは、「Oracle WebLogic Server Administration Console」から、テーブルleasing用のマルチデータソースを作成します。マルチデータソースを作成し、RACデータベースの各インスタンスを指定します。

以下の設定で、データソースを作成します。

- ◇ 「non-XA driver」を選択します。
- ◇ フォーマット<MultiDS>-rac0,<MultiDS>-rac1などを使用して、マルチデータベースの名前を付けます。
- ◇ 「Oracle's Driver (Thin) Version 9.0.1, 9.2.0, 10, 11」を選択します。
- ◇ 「Supports Global Transactions」を選択し、「One-Phase Commit」を選択します。
- ◇ データベースの「Service Name」を指定します。
- ◇ データソースをOSBクラスタにターゲットします。

以下の手順を実行して、マルチデータソースを作成します。

1. 「Oracle WebLogic Server Administration Console」の「Domain Structure」ウィンドウで、「Services」ノードを展開し、「JDBC」ノードを展開します。
2. 「Multi Data Sources」をクリックします。「Summary of JDBC Multi Data Source」ページが表示されます。
3. 「Lock and Edit」をクリックします。
4. 「New」をクリックします。「Create a New JDBC Multi Data Source」ページが表示されます。
5. 「New」をクリックします。
6. 「Name」フィールドに「leasing」と入力します。
7. 「JNDI name」フィールドに「jdbc/leasing」と入力します。
8. 「algorithm」として「Failover」を選択します。これはデフォルトの設定です。
9. 「Next」をクリックします。
10. 「target」として「OSB_Cluster」を選択します。
11. 「Next」をクリックします。
12. 「non-XA driver」を選択します。これはデフォルトの設定です。
13. 「Next」をクリックします。
14. 「Create New Data Source」をクリックします。
15. 「name」として、「leasing-rac0」と入力します。「JNDI name」として「jdbc/leasing-rac0」と入力します。「database type」として「oracle」と入力します。「driver type」フィールドに「Oracle Driver (Thin) for RAC server-Instance connection Version 10,11」と入力します。
注: テーブルleasing用のマルチデータソースを作成する時、以下のフォーマットで名前を入力します。
<MultiDS>-rac0,<MultiDS>-rac1など
16. 「Next」をクリックします。
17. 「Support Global Transactions」のチェックを外します。
18. 「Next」をクリックします。
19. 「Test Configuration and verify the connection works」をクリックします。
20. データソースを「OSB_Cluster」にターゲットします。
21. データソースを選択し、右側のスクリーンに追加します。
22. 「Create a New Data Source」をクリックして、RAC Databaseの一番目のインスタンスのデータソースを作成します。データソースを「OSB_Cluster」にターゲットします。RACデータベースの二番目のインスタンスに対して手順を繰り返します。
23. 二番目のデータソースをマルチデータソースに追加します。
24. 「Activate Changes」をクリックします。

5.3. OSBHOST1とOSBHOST2と管理サーバーの間でのホスト名検証証明書の有効化

ここでは、ノードマネージャと管理サーバーの間でホスト名検証を設定ために、適切な証明書を作成します。手順は項4.2「OSBHOST1でのノードマネージャ用のホスト検証証明書の有効化」と項4.4「OSBHOST2でのノードマネージャ用のホスト検証証明書の有効化」で説明されています。

5.4. ノードマネージャのプロパティ・ファイルの編集

ここでは、ノードマネージャのプロパティ・ファイルを編集します。このファイルは**nodemanager.properties**と呼ばれ、**ORACLE_BASE/product/osbmw/wlserver_10.3/common/nodemanager**ディレクトリに格納されています。サーバーの移行が正しく動作するために、以下のプロパティを追加する必要があります。

```
Interface=eth0
NetMask=255.255.255.0
UseMACBroadcast=true
```

◇ **Interface**

このプロパティは、浮動IPのインタフェースの名前(eth0など)を指定します。

◇ **NetMask**

このプロパティは、浮動IP用のインタフェースのネットマスクを指定します。

ネットマスクは、インタフェース上のネットマスクと一致する必要があります。このドキュメントでは、例として255.255.255.0を使用しています。

◇ **UseMACBroadcast**

このプロパティは、ARPパケットを送信する際に、ノードのMACアドレスを使用するかどうかを指定します。つまり、**aspring**コマンドで**-b**フラグを使用するかどうかを指定します。

ノードマネージャの出力を確認し、プロパティの使用または移行中に発生する問題を確認します。

ノードマネージャの出力は以下のように表示されます。

...

StateCheckInterval=500

Interface=eth0

NetMask=255.255.255.0

...

注: 以下の手順は、サーバー・プロパティ(起動プロパティ)が適切に設定されており、ノードマネージャがリモートでサーバーを起動できる場合、必要としません。

1. **Nodemanager.properties**ファイルで以下のプロパティを設定します。

◇ **StartScriptEnabled**

このプロパティを「**true**」に設定します。これは、ノードマネージャが管理対象サーバーを起動する時、使用する**shiphome**に必要なになります。

2. **ORACLE_BASE/product/osbmw/wlserver_10.3/server/bin**ディレクトリにある**startNodeManager.sh**スクリプトを実行して、ノード1とノード2でノードマネージャを起動します。

注: 共有ストレージのインストールからノードマネージャを実行する場合、複数のノードが同じ**nodemanager.properties**ファイルを使用して起動されます。しかし、各ノードは異なるネットマスクまたはインタフェースのプロパティが必要になる場合があります。その場合、環境変数を使用して各ノードで個別のパラメータを指定します。

例えば、異なるインタフェースを使用する場合、以下のようにインタフェース環境変数を設定します。環境変数がシェルスクリプトで設定された後、ノードマネージャを起動します。

```
OSBHOSTn> export JAVA_OPTIONS=-DInterface=eth3
```

5.5. wlsifconfig.shスクリプトの環境変数およびスーパーユーザーの権限の設定

ここで、**wlsifconfig**スクリプトの環境変数およびスーパーユーザーの権限を設定します。

1. 環境変数**PATH**が以下のファイルで使用されていることを確認します。

表8-1 **PATH**環境変数が必要なファイル

ファイル	格納先
wlsifconfig.sh	ORACLE_BASE/admin/<domain_name>/msserver/<domain_name>/bin/server_migration
wlscontrol.sh	ORACLE_BASE/product/osbmw/wlserver_10.3/common/bin
nodemanager.domains	ORACLE_BASE/product/osbmw/wlserver_10.3/common

2. wlsifconfig.shスクリプトを実行するために、sudoに権限を付与します。
- ✧ パスワード・プロンプトなしで動作するようにsudoを設定します。
- ✧ セキュリティのために、sudoにwlsifconfig.shを実行するために必要なコマンドのサブセットのみ付与します。以下の手順を実行して、wlsifconfig.shスクリプトの環境変数およびスーパーユーザーの権限を設定します。
 1. パスワードが必要ないsudoコマンドの実行権限および/sbin/ifconfigと/sbin/arpingバイナリの実行権限をWebLogicユーザー(oracle)に付与します。
 2. スクリプトがWebLogicユーザー(oracle)によって実行できることを確認します。以下は、sudo、ifconfig、arpingの実行権限をoracleユーザーに付与するために使用する/etc/sudoers内部のエントリの例です。

```
oracle ALL=NOPASSWD: /sbin/ifconfig,/sbin/arping
```

注: 必要なsudoとシステム権限をシステム管理者に求めます。

5.6. サーバー移行ターゲットの構成

ここでは、サーバー移行のターゲットを設定します。クラスタの移行を設定すると、DataSourceForAutomaticMigrationプロパティがtureに設定されます。以下の手順を実行して、クラスタの移行を設定します。

1. 「Oracle WebLogic Server Administration Console」にログインします。
(http://<host>:<adminPort>/console adminPortはデフォルトで7001です。)
2. 「Domain Structure」ウィンドウで、「Environment」を展開し、「Clusters」を選択します。「Summary of Cluster」ページが表示されます。
3. テーブルの「Name」列で、移行を構成するクラスタ(OSB_Cluster)をクリックします。
4. 「Migration」タブをクリックします。
5. 「Available」フィールドで、移行可能なマシンを選択し、右矢印をクリックします。ここで、「OSBHOST1」と「OSBHOST2」を選択します。
6. 自動移行に使用するデータソースを選択します。ここで、「leasing」データソースを選択します。
7. 「Save」をクリックします。
8. サーバー移行の候補マシンを設定します。すべての管理対象サーバーに対して、以下のタスクを実行する必要があります。
 1. 「Oracle WebLogic Server Administration Console」の「Domain Structure」ウィンドウで、「Environment」を展開し、「Servers」を選択します。
 2. 移行を設定するサーバーを選択します。
 3. 「Migration」タブをクリックします。
 4. 「Available」フィールドで、「Migration Configuration」セクションから移行可能なマシンを選択し、右矢印をクリックします。WLS_OSB1の場合は、OSBHOST2を選択します。WLS_OSB2の場合は、OSBHOST1を選択します。
 5. 「Automatic Server Migration Enabled」を選択します。これにより、ノードマネージャは障害が発生したサーバーをターゲット・ノードで自動的に起動できます。
 6. 「Save」をクリックします。
 7. 管理サーバーを再起動します。

ヒント:

「Summary of Servers」ページで「Customize this table」をクリックして、現在のマシンを「Available Windows」から「Chosen Window」に移動して、サーバーが稼働しているマシンを表示します。これはサーバーが自動的に移行される場合の設定とは異なります。

5.7. サーバー移行のテスト

ここでは、サーバーの移行をテストします。サーバーの移行が正常に動作していることを確認するには、以下の手順を実行します。

ノード1で以下の手順を実行します。

1. 管理対象サーバーWLS_OSB1を停止します。以下のコマンドを実行して、管理対象サーバーを停止します。

```
OSBHOST1> kill -9 <pid>
```

pidには、管理対象サーバーのプロセスIDを指定します。以下のコマンドを実行して、ノード内のpidを識別することができます。

```
OSBHOST1> ps -ef | grep WLS_OSB1
```

2. ノードマネージャのコンソールを確認してください。WLS_OSB1の浮動IPが無効になっていることを示すメッセージが表示されます。
3. ノードマネージャがWLS_OSB1の再起動を試行します。再起動する前に、ノードマネージャは30秒のフェンス時間を待ちます。
4. サーバーが起動されたら、再びサーバーを停止します。ノードマネージャは、再びローカルでサーバーを再起動しないというメッセージをログに記録します。

ノード2で以下の手順を実行します。

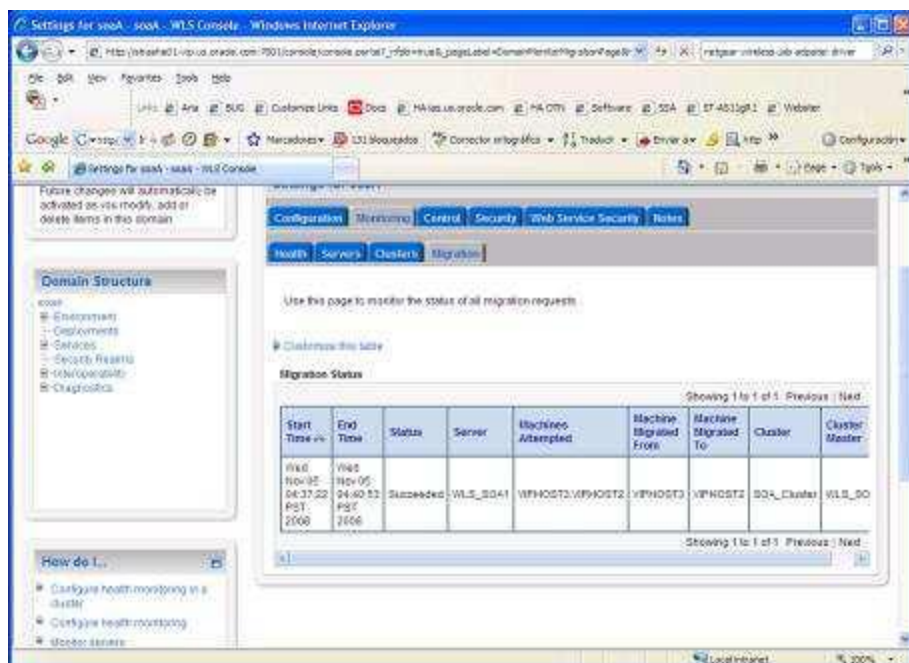
1. ローカルのノードマネージャのコンソールを確認してください。ノード1でWLS_OSB1を再起動してから30秒立つと、ノード2でのノードマネージャは、WLS_OSB1の浮動IPを立ち上げ、サーバーがノード2で起動されるというプロンプトが表示されます。
2. 同じIPを使用して、osb-infraコンソールに接続します。

「Oracle WebLogic Server Administration Console」でサーバーの移行を確認することもできます。

1. 「Oracle WebLogic Server Administration Console」にログインします。
2. 左側で「Domain」をクリックします。
3. 「Monitoring」タブをクリックし、「Migration」サブタブをクリックします。

「Migration Status」テーブルは、移行状態に関する情報を提供します。

図5-1 「Oracle WebLogic Server Administration Console」での「Migration Status」画面



付録A

A.1 用語

ここでは、コンポーネントを記述する用語について、説明します。

- ◇ **Oracleホーム:** Oracleホームには、特定の製品をホストするために必要なインストール・ファイルが含まれています。例えば、OSBのOracleホームには、Oracle Service Bus用のバイナリとライブラリファイルを含むディレクトリが含まれています。Oracleホームは、ミドルウェア・ホームのディレクトリ内に存在します。各Oracleホームは、複数のOracleインスタンスまたはOracle WebLogic Serverドメインと関連付けることができます。
- ◇ **WebLogicサーバー・ホーム:** WebLogicサーバー・ホームには、WebLogicサーバーをホストするために必要なインストール・ファイルが含まれています。WebLogicサーバーのホーム・ディレクトリは、Oracleホーム・ディレクトリのピアで、ミドルウェア・ホームディレクトリ内に存在します。
- ◇ **ミドルウェア・ホーム:** ミドルウェア・ホームは、Oracle WebLogic Serverホームおよび必要に応じて1つ以上のOracleホームで構成されています。ミドルウェア・ホームは、ローカル・ファイルシステムまたはNFSを通してリモート共有ディスク上に存在することができます。
- ◇ **Oracleインスタンス:** Oracleインスタンスには、Oracle Web Cache、Oracle HTTP Server、Oracle Internet Directoryなど1つまたは複数のアクティブなミドルウェア・システム・コンポーネントが含まれています。インストール時、またはインスタンスを作成し、設定する時、どのコンポーネントがインスタンスの一部であるかを決定します。Oracleインスタンスには、構成ファイル、ログ・ファイル、一時ファイルなどの更新可能なファイルが含まれています。
- ◇ **フェイルオーバー:** 高可用性システムのメンバーに予想しない障害が発生した場合（計画外のダウンタイム）、コンシューマへのサービス提供を継続するために、システムはフェイルオーバーを行います。アクティブ・パッシブシステムの場合、フェイルオーバーによって、パッシブメンバーがアクティブになり、コンシューマが障害が発生したメンバーからそのメンバーに送られます。フェイルオーバープロセスを手動で実行することができます。また、障害が発生したノードを検出し、クラスタ・リソースを障害が発生したノードからスタンバイノードに移動するためのハードウェア・クラスタ・サービスを設定して、フェイルオーバープロセスの自動化も実現できます。アクティブ・アクティブシステムの場合、フェイルオーバーは、リクエストをアクティブメンバーに発信するロードバランサのエンティティによって実行されます。アクティブメンバーに障害が発生した場合、ロードバランサは障害を検出し、自動的に障害が発生したメンバーへのリクエストを残りのアクティブメンバーにダイレクトします。
- ◇ **ファイルバック:** フェイルオーバーが成功した後、障害が発生したメンバーは時間をかけて修復され、スタンバイメンバーとしてシステムに再導入されます。必要に応じて、フェイルバック・プロセスは、このメンバーをアクティブにし、他のメンバーを非アクティブにすることもできます。フェイルバック・プロセスはシステムを障害発生前の構造に戻します。
- ◇ **共有ストレージ:** 共有ストレージは、ドメイン内のすべてのマシンから接続できるストレージ・サブシステムです。以下のディレクトリ構造は共有ディスクに配置されます。
 - ミドルウェア・ホーム・ソフトウェア
 - AdminServerドメイン・ホーム
 - JMS
 - Tlog(該当する場合)管理対象サーバーのホームは、必要に応じて共有ディスクに配置することができます。共有ストレージは、ネットワーク接続ストレージ(NAS)、ストレージ・エリア・ネットワーク(SAN)または複数のノードが同時にアクセスし、読み書きできる任意のストレージ・システムを使用することができます。
- ◇ **プライマリ・ノード:** プライマリ・ノードはOracle Fusion Middlewareインスタンスを実行し、バックアップとしてのセカンダリ・ノードを持つように構成されています。プライマリ・ノードに障害が発生した場合、Oracle Fusion Middlewareのインスタンスは、セカンダリ・ノードにフェイルオーバーにされます。フェイルオーバーは管理サーバー用のクラスタウェアによって、手動または自動で実行されます。サーバー移行のシナリオでは、自動フェイルオーバーを実現するために、WebLogicサーバーの全体移行を構成します。

- ◇ セカンダリ・ノード:これはOracle Fusion Middlewareインスタンスのバックアップノードです。プライマリ・ノードが使用不可になった場合、アクティブなインスタンスはセカンダリ・ノードにファイルオーバーされます。プライマリ・ノードの定義を参照してください。
- ◇ ネットワーク・ホスト名:ネットワーク・ホスト名は、`/etc/hosts`ファイルまたはDNS解決によって使用するIPアドレスに割り当てられた名前です。この名前は、参照するマシンが接続されているネットワークで参照できます。多くの場合、ネットワーク・ホスト名は物理ホスト名と同じです。しかし、各マシンは1つの物理ホスト名を持っていますが、複数のネットワーク・ホスト名を持つことができます。従って、マシンのネットワーク・ホスト名は常に物理ホスト名と違います。
- ◇ 物理ホスト名:このガイドでは、物理ホスト名とネットワーク・ホスト名を区別します。このガイドでは、マシンの“内部名”を参照するために、物理ホスト名を使用しています。UNIXでは、物理ホスト名は`hostname`コマンドを使用して戻されます。
ローカル・ホストを参照するために、物理ホスト名がOracle Fusion Middlewareによって使用されます。インストール時、物理ホスト名がインストーラにより、マシンから取得され、ディスク上のOracle Fusion Middlewareの構成メタデータに格納されます。
- ◇ 物理IP:物理IPはネットワーク上のマシンのIPを指します。ほとんどの場合、物理IPはマシンの物理ホスト名(物理ホスト名の定義を参照)に関連付けられています。ネットワーク上の仮想IPとは対照的に、物理IPは常に同じマシンに関連付けられています。
- ◇ スイッチオーバー:システムのアクティブなメンバーに対して、メンテナンスまたはアップグレードする必要があります。その場合、スイッチオーバーが開始され、代替メンバーによって実行されているワークロードを引き継ぎます。スイッチオーバーは、システムのコンシューマに継続的なサービスを提供します。
- ◇ スイッチバック:スイッチオーバー操作が実行されると、システムのメンバーは、メンテナンスまたはアップグレードのために非アクティブになっています。メンテナンスまたはアップグレードが完了した後、システムは、アップグレードしたメンバーをアクティブにし、スイッチオーバー前のシステム構造に戻すスイッチバック操作を行います。
- ◇ 仮想ホスト名:仮想ホスト名は、ロードバランサまたはハードウェア・クラスタを経由して1つ以上の物理マシンにマッピングされるネットワーク・アドレスのホスト名です。ロードバランサの場合、「仮想サーバー名」は、このドキュメントの「仮想ホスト名」と同じ意味で使用されます。ロードバランサは複数のサーバーの仮想ホスト名を持つことができ、クライアントが仮想ホスト名を使用して間接マシンと通信します。ハードウェア・クラスタの仮想ホスト名は、クラスタの仮想IPに割り当てられたネットワーク・ホスト名です。クラスタの仮想IPがクラスタの特定のノードに永久に接続されていないため、仮想ホスト名は特定のノードにも永久に接続されていません。
注:用語「仮想ホスト名」がこのドキュメントで使用されるたびに、それが仮想IPアドレスに関連付けられることを想定しています。IPアドレスのみを必要とする場合は、その旨が明示的に記載されています。
- ◇ 仮想IP、クラスタ仮想IP、ロードバランサ仮想IP:通常、仮想IPはハードウェア・クラスタまたはロードバランサに割り当てられることができます。ネットワーク・クライアントにクラスタのシングル・システム・ビューを提供するために、仮想IPがクラスタ・メンバー・グループへのエントリ・ポイントのIPアドレスとして機能します。仮想IPはサーバーのロードバランサまたはハードウェア・クラスタに割り当てられることができます。
ハードウェア・クラスタは、クラスタ仮想IPを使用して、クラスタ(スタンドアロン・マシン上にも設定できる)のエントリ・ポイントを外の世界に提示します。ハードウェア・クラスタのソフトウェアは、クラスタの2つの物理ノード間でこのIPアドレスの移動を管理します。クラスタはこのIPアドレスがどの物理ノードでアクティブであるかを認識する必要がありません。典型的な2つノードのハードウェアのクラスタ構成では、いくつかのクラスタIPアドレスを持つことができ、各マシンは独自の物理IPアドレスと物理ホスト名を持っています。これらのクラスタIPアドレスは、2つのノード間で浮動または移行します。クラスタIPアドレスの所有権を持つノードはそのアドレスに対してアクティブになっています。
ロードバランサも複数のサーバーへのエントリ・ポイントとして仮想IPを使用しています。これらのサーバーは同時にアクティブになる傾向があります。仮想IPアドレスは個々のサーバーではなく、サーバーとクライアント間のプロキシとして機能するロードバランサに割り当てられています。

A.2 共有ストレージの設定

ここでは、共有ストレージの場所を作成し、マウントします。これにより、OSBHOST1およびOSBHOST2は、2つのボリュームで同じ場所を参照できます。

「nasfiler」は共有ストレージのファイラです。
OSBHOST1で、以下のコマンドを実行します。

```
OSBHOST1> mount nasfiler:/vol/vol1/u01/app/oracle/product/osbmw/u01/app/oracle/product/osbmw -t  
nfs
```

OSBHOST2で、以下のコマンドを実行します。

```
OSBHOST2> mount nasfiler:/vol/vol2/u01/app/oracle/product/osbmw/u01/app/oracle/product/osbmw -t  
nfs
```

1つのボリュームのみが使用できる場合、ユーザーは共有ストレージの2つ異なるディレクトリを使用して、OSBサーバーの同じディレクトリにマウントして、バイナリの冗長性を提供します。

OSBHOST1で、以下のコマンドを実行します。

```
OSBHOST1> mount nasfiler:/vol/vol1/u01/app/oracle/product/osbmw1 u01/app/oracle/product/osbmw -t  
nfs
```

OSBHOST2で、以下のコマンドを実行します。

```
OSBHOST2> mount nasfiler:/vol/vol2/u01/app/oracle/product/osbmw2 u01/app/oracle/product/osbmw -t  
nfs
```

以下のコマンドは、異なるノードでOSB TXログの場所を共有する方法を示します。

```
OSBHOST1> mount nasfiler:/vol/vol1/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs  
/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs -t nfs  
OSBHOST2> nasfiler:/vol/vol1/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs  
/u01/app/oracle/stores/osbdomain/osb_cluster/tlogs -t nfs
```

注: 共有ストレージはNASまたはSANデバイスを使用できます。以下OSBHOST1からNASデバイスを使用する例です。オプションが異なる可能性があります。

```
OSBHOST1> mount nasfiler:/vol/vol1/osbmw11shared ORACLE_BASE/wls -t nfs -o  
rw,bg,hard,nointr,tcp,vers=3,timeo=300,rsize=32768,wsize=32768
```

A.3 ロードバランサの要件

このエンタープライズ・トポロジは外部のロードバランサを使用しています。外部ロードバランサは以下の機能を提供する必要があります。

- ✧ 仮想ホスト名を通して、サーバー・プールのトラフィックのロードバランシング能力
仮想ホスト名 (実際のホスト名ではなく) を使用して、クライアントからサービスへ接続します。ロードバランサは、プール内のサーバーへのリクエストをロードバランシングできます。
- ✧ ポート変換の設定
これにより、仮想ホスト名とポートへの着信要求はバックエンド・サーバーの別のポートに転送されます。
- ✧ プール内のサーバーのポートの監視
これにより、サービスの可用性を判断できます。
- ✧ 使用する外部ロードバランサで仮想サーバー名とポートの設定

仮想サーバー名およびポートは以下の要件を満たす必要があります。

- ロードバランサは、複数の仮想サーバーの構成を許可する必要があります。各仮想サーバーに対して、ロードバランサは、複数のポート上のトラフィック管理の設定を許可する必要があります。例えば、Web層がOracle HTTP Serverである場合、ロードバランサはHTTPおよびHTTPSトラフィック用の仮想サーバーとポートを構成する必要があります。
 - 仮想サーバー名はIPアドレスに関連付けられ、DNSの一部である必要があります。クライアントは、仮想サーバー名を使用して外部ロードバランサに接続できる必要があります。
 - ◇ ノード障害を検出し、即時に障害が発生したノードへのトラフィックのルーティングを停止する機能
 - ◇ フォルト・トレラント・モード:ロードバランサをフォルト・トレラント・モードに設定することをお勧めします。
 - ◇ バックエンド・サービスのトラフィック転送が利用できない場合は、コール元のクライアントに即時に戻るように、ロードバランサの仮想サーバーを構成することをお勧めします。これは、TCP/IP設定に基づいたタイムアウトの後に、クライアントが切断される場合に適しています。
 - ◇ スティックリーなルーティング機能:コンポーネントへのスティッキーな接続を維持する能力
- この例には、クッキー・ベースの永続性、IPベースの永続性などが含まれています。

ロードバランサは、SSLリクエストを終了し、同等の非SSLプロトコル（例えば、HTTPSからHTTPに）を使用して、トラフィックをバックエンド・サーバーに転送することができます。この機能はSSLアクセラレーションと呼び、ロードバランサにとって必要です。

注:Oracle Technology Network(<http://www.oracle.com.technology/index.html>)は、検証済みのロードバランサおよび構成を提供します。詳細については、以下のURLを参照してください

http://www.oracle.com/technology/products/ias/hi_av/Tested_LBR_FW_SSACcel.html

A.4 ユニキャスト要件

推奨トポロジ中のノードはユニキャストを使用して通信することをお勧めします。マルチキャスト通信と異なり、ユニキャストはクロス・ネットワーク構成を必要とせず、マルチキャスト・アドレスの競合から発生するネットワークエラーを減らします。

クラスタ通信を処理するためにユニキャストを使用する場合、以下の考慮事項が適用されます。

- ◇ WebLogicクラスタのすべてのメンバーが同じメッセージタイプを使用する必要があります。マルチキャストとユニキャストメッセージングを混在させることはできません。
- ◇ クラスタ・メンバーがクラスタのメッセージング・タイプをオーバーライドすることはできません。
- ◇ メッセージング・モードを変更するには(ユニキャストからマルチキャストへ、またはユニキャストからマルチキャストへ)、クラスタ全体をシャットダウンし、再起動する必要があります。
- ◇ マルチキャストを使用したJMSトピックは、クラスタ・アドレスに依存しない独自のマルチキャスト・アドレスにメッセージをパブリッシュするため、ユニキャストを使用したWebLogicクラスタに接続できます。ただし、以下の考慮事項が適用されます。
 - ユニキャスト・クラスタを可能にするルートハードウェア構成では、JMSマルチキャスト・サブスライバー機能を許可しない場合があります。
 - JMSマルチキャスト・サブスライバーは、マルチキャスト・アクセスを使用可能なネットワークハードウェア構成に存在する必要があります。(つまり、JMSサブスライバーは、マルチキャスト・トピックに接続するために、マルチキャスト対応可能なネットワークに存在する必要があります。)

A.5 OSB参考資料

1. [Oracle Service Busの高可用性について](#)
2. [Oracle Service Busクラスタについて](#)
3. [WebLogicサーバー全体の移行](#)