

**Oracle® Application Server**

高可用性ガイド

10g リリース 3 (10.1.3.1.0)

部品番号 : B31835-01

2006 年 12 月

Oracle Application Server 高可用性ガイド, 10g リリース 3 (10.1.3.1.0)

部品番号 : B31835-01

原本名 : Oracle Application Server High Availability Guide, 10g Release 3 (10.1.3.1.0)

原本部品番号 : B28941-03

Copyright © 2006, Oracle. All rights reserved.

#### 制限付権利の説明

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会社との契約に記載された制約条件に従うものとします。著作権、特許権およびその他の知的財産権と工業所有権に関する法律により保護されています。

独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバース・エンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントの情報は、予告なしに変更される場合があります。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションへの用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性がありま。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。オラクル社およびその関連会社は、第三者の製品およびサービスの品質、契約の履行（製品またはサービスの提供、保証義務を含む）に関しては責任を負いかねます。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

---

---

# 目次

はじめに .....	xi
対象読者 .....	xii
ドキュメントのアクセシビリティについて .....	xii
関連ドキュメント .....	xii
表記規則 .....	xii
サポートおよびサービス .....	xiii

## 第 I 部 概要

### 1 高可用性の概要

1.1 高可用性とは .....	1-2
1.1.1 高可用性の問題 .....	1-2
1.1.2 高可用性ソリューション .....	1-3
1.2 Oracle Application Server の高可用性の概要 .....	1-5
1.2.1 用語 .....	1-5
1.2.2 Oracle Application Server の基本アーキテクチャ .....	1-8
1.2.3 Oracle Identity Management と Oracle Application Server 10g リリース 3 (10.1.3.1.0) の使用 .....	1-9
1.2.4 Oracle Application Server の高可用性アーキテクチャ .....	1-9
1.2.5 最適な高可用性アーキテクチャの選択 .....	1-10
1.3 他のドキュメントの高可用性に関する情報 .....	1-11

### 2 Oracle Application Server 高可用性フレームワーク

2.1 OPMN でのプロセス管理 .....	2-2
2.2 状態情報のレプリケーション .....	2-2
2.3 Oracle Application Server でのロード・バランシング .....	2-3
2.4 OracleAS Cluster .....	2-5
2.5 外部ロード・バランサ .....	2-5
2.6 バックアップとリカバリ .....	2-8
2.6.1 Oracle Application Server Backup and Recovery Tool .....	2-8
2.7 障害時リカバリ .....	2-8
2.7.1 Oracle Application Server Guard .....	2-9
2.8 高可用性トポロジ：概要 .....	2-9
2.8.1 アクティブ / アクティブ・トポロジ .....	2-9
2.8.2 アクティブ / パッシブ・トポロジ：OracleAS Cold Failover Cluster .....	2-10

## 第 II 部 中間層の高可用性

### 3 アクティブ/アクティブ・トポロジ

3.1	アクティブ/アクティブ・トポロジについて .....	3-2
3.1.1	アクティブ/アクティブ・トポロジの OracleAS Cluster .....	3-3
3.1.2	アクティブ/アクティブ・トポロジにおけるアプリケーションレベルの クラスタリング .....	3-4
3.1.2.1	OracleAS Cluster (OC4J) によるステートフル・セッション EJB の 状態レプリケーション .....	3-5
3.1.3	アクティブ/アクティブ・トポロジの Oracle Application Server インスタンスの プロパティ .....	3-6
3.1.4	グループについて .....	3-6
3.1.4.1	グループの作成 .....	3-7
3.1.4.2	追加の OC4J インスタンスの作成 .....	3-7
3.1.4.3	グループ内のインスタンスの管理 .....	3-9
3.1.4.4	admin_client.jar を使用した、グループへのアプリケーションのデプロイ .....	3-10
3.1.5	Oracle HTTP Server がリクエストを OC4J にルーティングする方法 .....	3-10
3.1.6	アクティブ/アクティブ・トポロジでの Oracle Identity Management の使用 .....	3-11
3.1.7	アクティブ/アクティブ・トポロジでの Oracle HTTP Server 10.1.2 の使用 .....	3-13
3.1.8	アクティブ/アクティブ・トポロジでの OracleAS Web Cache リリース 2 (10.1.2) の使用 .....	3-13
3.2	アクティブ/アクティブ・トポロジの管理 .....	3-13
3.2.1	OracleAS Cluster の設定 .....	3-14
3.2.1.1	動的検出方法 .....	3-14
3.2.1.2	検出サーバー方法 .....	3-14
3.2.2	マルチキャスト・レプリケーションの設定 .....	3-15
3.2.3	peer-to-peer レプリケーションの設定 .....	3-16
3.2.4	データベースへのレプリケーションの設定 .....	3-18
3.2.5	レプリケーション・ポリシーの設定 .....	3-18
3.2.6	レプリケート先のノードの数の指定 .....	3-19
3.2.7	コンポーネントのステータスのチェック .....	3-20
3.2.8	トポロジ内のコンポーネントの起動と停止 .....	3-20
3.2.9	クラスタへのアプリケーションのデプロイ .....	3-20
3.2.10	アクティブ/アクティブ・トポロジへのインスタンスの追加 .....	3-21
3.2.11	アクティブ/アクティブ・トポロジからのインスタンスの削除 .....	3-21
3.2.12	mod_oc4j のロード・バランシング・オプションの設定 .....	3-21
3.2.13	Java Message Service (JMS) での高可用性の構成 .....	3-24
3.3	Oracle HTTP Server および OC4J における高可用性機能の要約 .....	3-24
3.4	その他のトピック .....	3-24
3.4.1	JNDI ネームスペースのレプリケーション .....	3-25
3.4.2	EJB クライアント・ルーティング .....	3-25
3.4.3	Java Object Cache を使用した OC4J の分散キャッシュ .....	3-25

### 4 アクティブ/パッシブ・トポロジ

4.1	アクティブ/パッシブ・トポロジについて .....	4-2
4.2	アクティブ/パッシブ・トポロジの管理 .....	4-5
4.2.1	Application Server Control コンソールを使用した管理 .....	4-5
4.2.2	コンポーネントの起動と停止 .....	4-5

4.2.3	アプリケーションのデプロイ .....	4-5
4.3	アクティブ / パッシブ・トポロジにおける Oracle HTTP Server および OC4J の 高可用性機能の概要 .....	4-5

## 5 Oracle SOA Suite の高可用性

5.1	インストールに関する注意事項 .....	5-2
5.2	Oracle BPEL Process Manager .....	5-2
5.2.1	Oracle BPEL Process Manager について .....	5-2
5.2.2	アクティブ / アクティブ・トポロジの Oracle BPEL Process Manager .....	5-3
5.2.2.1	アクティブ / アクティブ・トポロジでの BPEL プロセスの起動 .....	5-4
5.2.2.2	デハイドレーション・ストアに Real Application Clusters データベースを使用 .....	5-4
5.2.2.3	異なるサブネットにあるマシンでのアクティブ / アクティブ・トポロジの実行 .....	5-5
5.2.3	アクティブ / パッシブ・トポロジの Oracle BPEL Process Manager .....	5-7
5.2.4	アダプタとの Oracle BPEL Process Manager の使用 .....	5-7
5.2.4.1	J2CA ベースのアダプタの概要 .....	5-7
5.2.4.2	同時実行性のサポート .....	5-8
5.2.4.3	アダプタのアクティブ / アクティブ・トポロジ .....	5-8
5.2.4.4	アダプタの変更済アクティブ / アクティブ・トポロジ .....	5-9
5.2.4.5	アダプタのアクティブ / パッシブ・トポロジ .....	5-10
5.3	Oracle Enterprise Service Bus .....	5-12
5.3.1	Oracle Enterprise Service Bus について .....	5-12
5.3.2	アクティブ / アクティブ・トポロジの Oracle Enterprise Service Bus .....	5-12
5.3.2.1	ESB リポジトリ・サーバーが ESB ランタイム・サーバーと異なる ESB クラスタにあることの検証 .....	5-15
5.3.2.2	仮想ホスト名およびポートの ORAESB スキーマへの登録の検証 .....	5-15
5.3.2.3	Oracle Enterprise Service Bus での Real Application Clusters データベースの 使用 .....	5-15
5.3.2.4	OC4J インスタンスの OC4J クラスタへのクラスタ化 .....	5-16
5.3.2.5	Oracle Enterprise Service Bus サービスへのアクセス .....	5-16
5.3.2.6	アプリケーションの登録 .....	5-16
5.3.3	Oracle Enterprise Service Bus での Oracle Application Server アダプタの使用 .....	5-16
5.4	Oracle Business Activity Monitoring .....	5-16
5.4.1	Oracle Business Activity Monitoring について .....	5-16
5.4.2	要件 .....	5-19
5.4.3	インストールに関する重要項目 .....	5-19
5.4.3.1	Active Data Cache、Event Engine および Report Cache のインストール .....	5-19
5.4.3.2	Web アプリケーションのインストール .....	5-20
5.4.3.3	Enterprise Link および Plan Monitor のインストール .....	5-20
5.4.4	Microsoft Cluster Server (MSCS) の構成 .....	5-20
5.4.4.1	Oracle BAM Active Data Cache リソース・タイプの作成 .....	5-20
5.4.4.2	Oracle BAM Active Data Cache リソースの作成 .....	5-21
5.4.4.3	Oracle BAM Report Cache リソースの作成 .....	5-22
5.4.4.4	Oracle BAM Event Engine リソースの作成 .....	5-23
5.4.5	Microsoft IIS 6 の Web ガーデンの設定 .....	5-25
5.4.6	Enterprise Link と Plan Monitor の構成 .....	5-25
5.4.7	既知の問題とトラブルシューティング .....	5-25
5.4.7.1	クラスタ・ノードの障害発生時に Enterprise Link でエラーが発生する .....	5-26
5.4.7.2	Active Data Cache の実行ノードの障害発生時に Active Viewer が別ノードに 自動再接続されない .....	5-26

5.4.7.3	Real Application Clusters データベース内のノードの障害発生時にデータが失われる場合がある .....	5-26
5.4.7.4	Plan Monitor が Active Data Cache または Data Flow Service (DFS) に再接続されない .....	5-27
5.4.7.5	Plan Monitor が Enterprise Link に自動再接続されない .....	5-27
5.4.7.6	ハードウェア・クラスタのスタンバイ・ノードで icommand を実行するとエラーが表示される .....	5-27
5.4.7.7	フェイルオーバー時にアラートが起動されない .....	5-28
5.4.7.8	高速接続フェイルオーバー (FCF) の未サポート .....	5-28
5.4.8	メッセージ整合の設定 .....	5-28
5.4.8.1	「Oracle BAM Enterprise Message Receiver」ダイアログで「Run forever」を選択する .....	5-28
5.4.8.2	サブプランに各レコードの反復処理を設定する .....	5-29
5.4.8.3	グローバル・トランザクションに変換を含める .....	5-30
5.4.8.4	グローバル・トランザクションにメッセージ・トラッカを含める .....	5-31
5.5	Oracle Service Registry .....	5-31
5.6	Oracle Business Rules .....	5-31
5.6.1	リポジトリのタイプ .....	5-32
5.6.2	リポジトリへの WebDAV URL .....	5-32
5.6.3	Real Application Clusters データベースと Oracle Business Rules .....	5-32
5.6.4	高可用性環境の Rule Author .....	5-32
5.7	Oracle Web Services Manager .....	5-32

### 第 III 部 障害時リカバリ

## 6 OracleAS Disaster Recovery

6.1	Oracle Application Server 10g Disaster Recovery ソリューション .....	6-3
6.1.1	OracleAS Disaster Recovery の要件 .....	6-4
6.1.2	サポートされる Oracle Application Server リリースとオペレーティング・システム .....	6-5
6.1.3	サポートされているトポロジ .....	6-5
6.1.3.1	対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による本番サイトの完全なミラー .....	6-6
6.1.3.2	非対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による単純な非対称のスタンバイ・トポロジ .....	6-8
6.1.3.3	OracleAS Portal の OracleAS Metadata Repository は別の場所にあり、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure (部門別トポロジ) .....	6-10
6.1.3.4	アプリケーション OracleAS Metadata Repository が分散され、Oracle Identity Management と OracleAS Metadata Repository が同じ場所でない Infrastructure .....	6-10
6.1.3.5	冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ .....	6-12
6.1.3.6	既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合された冗長な単一 OracleAS 10.1.3 Oracle ホーム J2EE トポロジ .....	6-13
6.2	OracleAS Disaster Recovery 環境の準備 .....	6-14
6.2.1	ホスト名の計画と割当て .....	6-14
6.2.1.1	物理ホスト名 .....	6-16
6.2.1.2	ネットワーク・ホスト名 .....	6-17
6.2.1.3	仮想ホスト名 .....	6-17
6.2.1.4	仮想ホストの別名 .....	6-17
6.2.2	ホスト名解決の構成 .....	6-18
6.2.2.1	ローカル・ホストネーミング・ファイル解決の使用 .....	6-18

6.2.2.2	DNS 解決の使用 .....	6-20
6.2.2.2.1	DNS サーバーへの Oracle Data Guard 用エントリの追加 .....	6-22
6.3	Oracle Application Server のインストールの概要 .....	6-22
6.4	OracleAS Guard と asgctl の概要 .....	6-23
6.4.1	asgctl の概要 .....	6-23
6.4.2	OracleAS Guard クライアント .....	6-23
6.4.3	OracleAS Guard サーバー .....	6-24
6.4.4	asgctl 操作 .....	6-24
6.4.5	OracleAS Guard と OPMN との統合 .....	6-28
6.4.6	サポートされる OracleAS Disaster Recovery 構成 .....	6-28
6.4.7	OracleAS Guard の構成とその他の関連情報 .....	6-28
6.5	データベースの認証 .....	6-30
6.6	トポロジの検出、ダンプおよび検証 .....	6-31
6.7	いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用 .....	6-32
6.8	冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ内の OracleAS 10.1.3 インスタンスの検出 .....	6-33
6.9	既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除 .....	6-34
6.10	OracleAS Guard 操作: スタンバイ・システムへの 1 つ以上の本番インスタンスのスタンバイ・サイト・クローニング .....	6-35
6.10.1	スタンバイ・システムへの 1 つまたは複数の本番インスタンスのクローニング .....	6-37
6.11	OracleAS Guard の操作: スタンバイのインスタンス化とスタンバイの同期化 .....	6-40
6.11.1	スタンバイのインスタンス化 .....	6-40
6.11.2	スタンバイの同期化 .....	6-41
6.12	実行時操作: OracleAS Guard のスイッチオーバーおよびフェイルオーバー操作 .....	6-42
6.12.1	停止 .....	6-42
6.12.1.1	スケジューリングした停止 .....	6-42
6.12.1.2	計画外停止 .....	6-45
6.13	Real Application Clusters データベースを使用しない OracleAS Disaster Recovery の構成 .....	6-47
6.13.1	前提条件 .....	6-47
6.13.2	構成手順 .....	6-48
6.13.3	スイッチオーバー手順 .....	6-48
6.13.4	スイッチバック手順 .....	6-49
6.13.5	フェイルオーバー手順 .....	6-49
6.14	Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery .....	6-49
6.14.1	プライマリ・サイトおよびスタンバイ・サイトの両方で Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の構成 .....	6-49
6.14.1.1	前提条件 .....	6-50
6.14.1.2	構成手順 .....	6-50
6.14.1.3	スイッチオーバー手順 .....	6-53
6.14.1.4	スイッチバック手順 (プライマリ・サイトへのスイッチバック) .....	6-54
6.14.1.5	フェイルオーバー手順 .....	6-56
6.14.2	プライマリ・サイトのみで Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の構成 (スタンバイ・サイトでは Real Application Clusters 以外のデータベースを使用) .....	6-56
6.14.2.1	前提条件 .....	6-56

6.14.2.2	構成手順 .....	6-57
6.14.2.3	スイッチオーバー手順 .....	6-59
6.14.2.4	スイッチバック手順 .....	6-60
6.14.2.5	フェイルオーバー手順 .....	6-61
6.15	OracleAS Guard 操作の監視とトラブルシューティング .....	6-61
6.15.1	トポロジの検証 .....	6-61
6.15.2	現在の操作の表示 .....	6-62
6.15.3	完了した操作のリストの表示 .....	6-63
6.15.4	オペレーションの停止 .....	6-63
6.15.5	タスクのトレース .....	6-63
6.15.6	トポロジに関する情報のファイルへの書込み .....	6-64
6.15.7	エラー・メッセージ .....	6-64
6.16	ワイド・エリア DNS の操作 .....	6-64
6.16.1	ワイド・エリア・ロード・バランサの使用 .....	6-64
6.16.2	DNS 名の手動変更 .....	6-64
6.17	OracleAS Guard コマンドライン・ユーティリティ (asgctl) の使用 .....	6-65
6.17.1	asgctl を使用する一般的な OracleAS Guard セッション .....	6-65
6.17.1.1	ヘルプの参照 .....	6-65
6.17.1.2	プライマリ・データベースの指定 .....	6-66
6.17.1.3	トポロジの検出 .....	6-66
6.17.1.4	asgctl スクリプトの作成と実行 .....	6-67
6.17.2	OracleAS Guard asgctl スクリプトの定期的なスケジュール .....	6-67
6.17.3	Enterprise Manager Job System を使用した OracleAS Guard ジョブの実行 .....	6-67
6.18	いくつかの OracleAS Metadata Repository 構成の特別な考慮事項 .....	6-68
6.18.1	複数の OracleAS Metadata Repository 構成の特別な考慮事項 .....	6-68
6.18.1.1	asgctl 資格証明の設定 .....	6-68
6.18.1.2	プライマリ・データベースの指定 .....	6-69
6.18.1.3	OracleAS Guard ポート番号の設定 .....	6-69
6.18.2	OracleAS Metadata Repository Creation Assistant を使用して作成した OracleAS Metadata Repository 構成の特別な考慮事項 .....	6-69
6.19	OracleAS Disaster Recovery 環境の特別な考慮事項 .....	6-70
6.19.1	いくつかの OracleAS Disaster Recovery サイトを設定する際に注意する必要がある 特別な考慮事項 .....	6-70
6.19.2	非対称トポロジの構成ファイル ons.conf および dsa.conf の処理 .....	6-70
6.19.3	OracleAS Disaster Recovery 環境における他の特別な考慮事項 .....	6-71

## 7 OracleAS Guard asgctl コマンドライン・リファレンス

7.1	OracleAS Guard asgctl コマンドに共通な情報 .....	7-3
7.2	OracleAS Guard コマンドの一部に特有の情報 .....	7-4
7.2.1	CFC 環境の OracleAS Disaster Recovery 構成に関する特別な考慮事項 .....	7-5
7.2.1.1	CFC 環境でインスタンス化およびフェイルオーバー操作を実行するときの特別な 考慮事項 .....	7-5
7.2.1.2	CFC 環境でインスタンス化操作を実行するときの特別な考慮事項と回避方法 .....	7-6
7.2.1.3	CFC 環境でスイッチオーバー操作を実行するときの特別な考慮事項 .....	7-6
7.2.2	OracleAS Disaster Recovery 環境における他の特別な考慮事項 .....	7-6
	add instance .....	7-7
	asgctl .....	7-9
	clone instance .....	7-10

clone topology .....	7-13
connect asg .....	7-16
create standby database .....	7-17
disconnect .....	7-19
discover topology .....	7-20
discover topology within farm .....	7-22
dump policies .....	7-23
dump topology .....	7-24
exit .....	7-26
failover .....	7-27
help .....	7-28
instantiate topology .....	7-29
quit .....	7-31
remove instance .....	7-32
run .....	7-34
set asg credentials .....	7-35
set echo .....	7-37
set new primary database .....	7-38
set noprompt .....	7-39
set primary database .....	7-40
set trace .....	7-42
show env .....	7-43
show operation .....	7-44
shutdown .....	7-46
shutdown topology .....	7-47
startup .....	7-48
startup topology .....	7-49
stop operation .....	7-50
switchover topology .....	7-51
sync topology .....	7-54
verify topology .....	7-56
dump farm (廃止済) .....	7-58
instantiate farm (廃止済) .....	7-59
shutdown farm (廃止済) .....	7-60
startup farm (廃止済) .....	7-61
switchover farm (廃止済) .....	7-62
sync farm (廃止済) .....	7-64
verify farm (廃止済) .....	7-65

## 8 手動同期化操作

8.1 OracleAS Guard の asgctl コマンドライン・ユーティリティを使用しない、 ベースライン・インストールとスタンバイ・サイトの手動による同期化 .....	8-2
8.1.1 本番サイトの手動バックアップ .....	8-2
8.1.1.1 OracleAS Infrastructure データベースのアーカイブ・ログの転送 .....	8-3

8.1.1.2	構成ファイルのバックアップ (OracleAS Infrastructure および中間層) .....	8-4
8.1.2	スタンバイ・サイトの手動リストア .....	8-4
8.1.2.1	構成ファイルのリストア (OracleAS Infrastructure および中間層) .....	8-5
8.1.2.2	OracleAS Infrastructure データベースのリストア: ログ・ファイルの適用 .....	8-6

## 9 OracleAS Disaster Recovery サイトのアップグレード手順

9.1	前提条件 .....	9-2
9.2	Disaster Recovery のトポロジ .....	9-2
9.3	高レベル OracleAS Disaster Recovery のアップグレード手順 .....	9-2
9.4	既存の OracleAS Disaster Recovery 環境へのパッチ適用 .....	9-6

## 10 DNS サーバーの設定

## 11 Secure Shell (SSH) ポート・フォワーディング

11.1	SSH ポート・フォワーディング .....	11-2
------	------------------------	------

## 第 IV 部 付録

### A 高可用性のトラブルシューティング

A.1	OracleAS Disaster Recovery トポロジのトラブルシューティング .....	A-2
A.1.1	スタンバイ・サイトが同期化されない .....	A-2
A.1.2	フェイルオーバーまたはスイッチオーバー後にスタンバイ・インスタンスの起動に失敗する .....	A-2
A.1.3	dcmctl resyncInstance -force -script の手順でスイッチオーバー操作に失敗する .....	A-3
A.1.4	スタンバイ・サイトでスタンドアロンの OracleAS Web Cache インストールを起動できない .....	A-4
A.1.5	スタンバイ・サイトの中間層インストールで間違ったホスト名が使用されている .....	A-4
A.1.6	スタンバイ・ファームのファーム検証操作に失敗する .....	A-4
A.1.7	ファームの同期操作でエラー・メッセージが返される .....	A-5
A.1.8	Windows システムで PATH 環境変数が 1024 文字を超過している場合に asgctl startup コマンドの実行に失敗する可能性がある .....	A-6
A.2	中間層コンポーネントのトラブルシューティング .....	A-7
A.2.1	OracleAS Cluster (OC4J-EJB) での複数の NIC の使用 .....	A-7
A.2.2	「opmn:」 URL 接頭辞を使用するとパフォーマンスが遅くなる .....	A-8
A.3	その他の問題の場合 .....	A-9

### B OracleAS Guard エラー・メッセージ

B.1	DGA エラー・メッセージ .....	B-2
B.1.1	LRO エラー・メッセージ .....	B-2
B.1.2	Undo エラー・メッセージ .....	B-3
B.1.3	テンプレートの作成中のエラー・メッセージ .....	B-3
B.1.4	フィジカル・スタンバイのスイッチオーバーのエラー・メッセージ .....	B-4
B.2	Duf エラー・メッセージ .....	B-4
B.2.1	データベース・エラー・メッセージ .....	B-10
B.2.2	接続とネットワークのエラー・メッセージ .....	B-15
B.2.3	SQL*Plus エラー・メッセージ .....	B-16

B.2.4	JDBC エラー・メッセージ .....	B-16
B.2.5	OPMN エラー・メッセージ .....	B-17
B.2.6	Net Services エラー・メッセージ .....	B-18
B.2.7	LDAP または OID のエラー・メッセージ .....	B-20
B.2.8	システム・エラー・メッセージ .....	B-21
B.2.9	警告エラー・メッセージ .....	B-21
B.2.10	OracleAS データベース・エラー・メッセージ .....	B-21
B.2.11	OracleAS トポロジ・エラー・メッセージ .....	B-22
B.2.12	OracleAS バックアップおよびリストアのエラー・メッセージ .....	B-23
B.2.13	OracleAS Guard 同期化エラー・メッセージ .....	B-25
B.2.14	OracleAS Guard インスタンス化エラー・メッセージ .....	B-26

## **C Oracle Notification Server (ONS) を使用せずに管理する Oracle Application Server Cluster**

C.1	ONS を使用せずに管理する OracleAS Cluster について .....	C-2
C.2	構成作業 .....	C-2

## **索引**



---

---

# はじめに

「はじめに」の項は次のとおりです。

- [対象読者](#)
- [ドキュメントのアクセシビリティについて](#)
- [関連ドキュメント](#)
- [表記規則](#)
- [サポートおよびサービス](#)

## 対象読者

『Oracle Application Server 高可用性ガイド』は、高可用性が必要とされる Oracle Application Server を配置および管理する役割を持つ管理者や開発者を対象としています。

## ドキュメントのアクセシビリティについて

オラクル社は、障害のあるお客様にもオラクル社の製品、サービスおよびサポート・ドキュメントを簡単にご利用いただけることを目標としています。オラクル社のドキュメントには、ユーザーが障害支援技術を使用して情報を利用できる機能が組み込まれています。HTML 形式のドキュメントで用意されており、障害のあるお客様が簡単にアクセスできるようにマークアップされています。標準規格は改善されつつあります。オラクル社はドキュメントをすべてのお客様がご利用できるように、市場をリードする他の技術ベンダーと積極的に連携して技術的な問題に対応しています。オラクル社のアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト <http://www.oracle.com/accessibility/> を参照してください。

### ドキュメント内のサンプル・コードのアクセシビリティについて

スクリーン・リーダーは、ドキュメント内のサンプル・コードを正確に読めない場合があります。コード表記規則では閉じ括弧だけを行に記述する必要があります。しかし JAWS は括弧だけの行を読まない場合があります。

### 外部 Web サイトのドキュメントのアクセシビリティについて

このドキュメントにはオラクル社およびその関連会社が所有または管理しない Web サイトへのリンクが含まれている場合があります。オラクル社およびその関連会社は、それらの Web サイトのアクセシビリティに関しての評価や言及は行っておりません。

### Oracle サポート・サービスへの TTY アクセス

アメリカ国内では、Oracle サポート・サービスへ 24 時間年中無休でテキスト電話 (TTY) アクセスが提供されています。TTY サポートについては、(800)446-2398 にお電話ください。

## 関連ドキュメント

詳細は、次の Oracle ドキュメントを参照してください。

- 『Oracle Application Server 概要』
- Oracle Application Server のインストール・ガイド
- 『Oracle Application Server 管理者ガイド』

## 表記規則

本文では、次の表記規則を使用します。

規則	意味
太字	太字は、操作に関連するグラフィカル・ユーザー・インタフェース要素、または本文中で定義されている用語および用語集に記載されている用語を示します。
イタリック	イタリックは、特定の値を指定するプレースホルダ変数を示します。
固定幅フォント	固定幅フォントは、パラグラフ内のコマンド、URL、例に記載されているコード、画面に表示されるテキスト、または入力するテキストを示します。

# サポートおよびサービス

次の各項に、各サービスに接続するための URL を記載します。

## Oracle サポート・サービス

オラクル製品サポートの購入方法、および Oracle サポート・サービスへの連絡方法の詳細は、次の URL を参照してください。

<http://www.oracle.co.jp/support/>

## 製品マニュアル

製品のマニュアルは、次の URL にあります。

<http://otn.oracle.co.jp/document/>

## 研修およびトレーニング

研修に関する情報とスケジュールは、次の URL で入手できます。

<http://www.oracle.co.jp/education/>

## その他の情報

オラクル製品やサービスに関するその他の情報については、次の URL から参照してください。

<http://www.oracle.co.jp>

<http://otn.oracle.co.jp>

---

---

**注意：** ドキュメント内に記載されている URL や参照ドキュメントには、Oracle Corporation が提供する英語の情報も含まれています。日本語版の情報については、前述の URL を参照してください。

---

---



# 第 I 部

---

## 概要

この部は、Oracle Application Server の高可用性の概要を説明する章で構成されています。

- [第 1 章「高可用性の概要」](#)
- [第 2 章「Oracle Application Server 高可用性フレームワーク」](#)



---

---

## 高可用性の概要

このリリースの Oracle Application Server では、前のリリースで使用可能であった高可用性ソリューションが拡張および強化されています。Oracle Application Server のフレキシブルで自動化された新しい高可用性ソリューションのテストが実施されました。これらのソリューションについては、このマニュアルで説明されています。これらのソリューションはすべて、Oracle Application Server に配置するアプリケーションがビジネス目標の達成に必要な可用性を実現することを目的としています。このマニュアルで説明するソリューションと手順の目的は、Oracle Application Server コンポーネントのシングル・ポイント障害をなくし、サービスをいっさい停止しないか最小限の停止にとどめることにあります。

この章では、高可用性とその重要性について Oracle Application Server の観点から説明します。この章に含まれている項は次のとおりです。

- [第 1.1 項「高可用性とは」](#)
- [第 1.2 項「Oracle Application Server の高可用性の概要」](#)
- [第 1.3 項「他のドキュメントの高可用性に関する情報」](#)

## 1.1 高可用性とは

この項では、問題解決の観点から高可用性の概要を示します。この項の内容は次のとおりです。

- 第 1.1.1 項「高可用性の問題」
- 第 1.1.2 項「高可用性ソリューション」

### 1.1.1 高可用性の問題

ミッション・クリティカルなコンピュータ・システムは、24 時間 365 日利用可能である必要があります。しかし、計画停止または計画外停止では、システムの一部またはすべてを停止することがあります。システムの可用性は、システム導入後の合計時間のうちのサービス提供時間の割合で測定されます。表 1-1 に、例を示します。

**表 1-1 可用性の割合と対応する停止時間の値**

可用性の割合	年間停止時間の概算
95%	18 日
99%	4 日
99.9%	9 時間
99.99%	1 時間
99.999%	5 分

表 1-2 に、コンピュータ・システムで想定される様々な障害のタイプを示します。

**表 1-2 システムの停止および障害のタイプ**

停止のタイプ	障害のタイプ
計画外停止	システム障害
	データ障害
	災害
	人為ミス
計画停止	システム・メンテナンス（オペレーティング・システム、アプリケーション・サーバー、構成、アプリケーションの変更などのハードウェアおよびソフトウェアの変更を含む）
	データ・メンテナンス

停止のこれら 2 つのタイプ（計画停止と計画外停止）は通常、システムの可用性要件の設計時には別々に扱われます。システムの要件は、計画外停止に関しては非常に制限的な場合がありますが、計画停止に関しては非常にフレキシブルな場合があります。これは、営業時間には負荷がピークに達するが、夜間と週末にはほとんど非アクティブなままであるアプリケーションの場合に当てはまります。

## 1.1.2 高可用性ソリューション

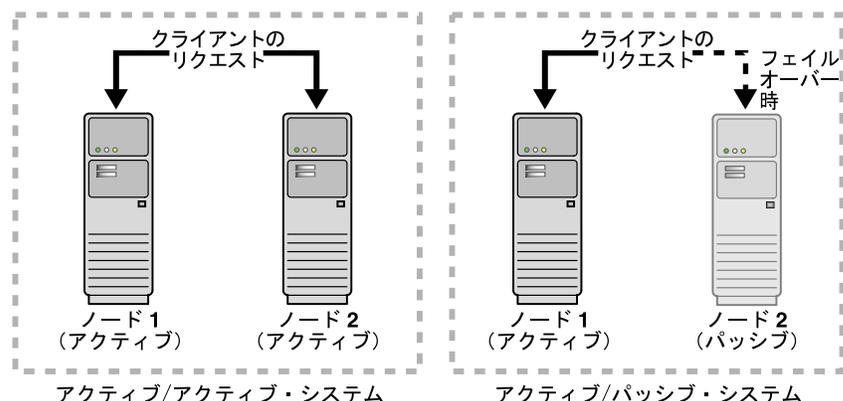
高可用性ソリューションは、1つのデータ・センター配置における高可用性を実現するローカル高可用性ソリューションと、通常は地理的に分散配置されていて、洪水や地域的なネットワークの停止などの災害からアプリケーションを保護する、障害時リカバリ・ソリューションに分類できます。

想定される障害タイプのうち、プロセス障害、ノード障害、メディア障害および人為ミスは、ローカル高可用性ソリューションで保護できます。ローカルの物理的な障害については、地理的に分散された障害時リカバリ・ソリューションで保護できます。

高可用性の問題を解決するには、いくつかのテクノロジーとベスト・プラクティスが必要になります。その中で最も重要なメカニズムは冗長性です。高可用性は、システムとコンポーネントに冗長性を持たせることによって実現されます。ローカル高可用性ソリューションは、冗長性のレベルによってアクティブ/アクティブ・ソリューションとアクティブ/パッシブ・ソリューションに分類されます（図 1-1 を参照）。

- アクティブ/アクティブ・ソリューションでは、複数のアクティブなシステム・インスタンスが配置されます。これにより、スケーラビリティが向上し、高可用性が実現されます。すべてのインスタンスは、同時にリクエストを処理します。
- アクティブ/パッシブ・ソリューションでは、リクエストを処理するアクティブ・インスタンスと、スタンバイ状態のパッシブ・インスタンスを配置します。さらに、この2つのインスタンスの間にハートビート・メカニズムが設定されます。このメカニズムは、オペレーティング・システムのベンダー固有のクラスタウェアで提供および管理されます。通常、ベンダー固有のクラスタ・エージェントでは、クラスタ・ノード間の監視とフェイルオーバーを自動的に行うことも可能なため、アクティブ・インスタンスに障害が発生した場合には、エージェントがそのアクティブ・インスタンスを完全に停止して、パッシブ・インスタンスをアクティブ化します。アプリケーション・サービスは、処理を再開できます。その結果、アクティブ/パッシブの役割が切り替わります。計画停止または計画外停止の場合は、この同じ手順を手動で行えます。アクティブ/パッシブ・ソリューションは一般に、コールド・フェイルオーバー・クラスタとも呼ばれます。

図 1-1 アクティブ/アクティブおよびアクティブ/パッシブの高可用性ソリューション



包括的な高可用性システムでは、アーキテクチャの冗長性に加えて、次のローカル高可用性テクノロジーも必要です。

- プロセス障害の検出と自動再起動

構成やソフトウェアに障害が発生したため、予期しないときにプロセスが停止する場合があります。プロセスの監視と再起動が適切に行われるシステムでは、すべてのシステム・プロセスが常時監視され、障害がある場合にそれらが再起動されます。

システム・プロセスは、特定の時間間隔における再起動の回数も記録している必要があります。短期間に何度も再起動を行うと他の障害を引き起こすことがあるため、これも重要な機能です。したがって、特定の時間間隔における再起動や再試行の回数の上限も指定しておく必要があります。

- クラスタリング

システムのコンポーネントをまとめてクラスタリングすることにより、クライアント側は、ランタイム処理と管理性の観点から、これらのコンポーネントを機能的に1つのエンティティとみなすことができます。クラスタは、1つのコンピュータ、または同じワークロードを共有する複数のコンピュータ上で実行される一連のプロセスです。クラスタリングと冗長性には密接な関係があります。クラスタは、システムの冗長性を実現します。

- 構成管理

類似したコンポーネントを1つのグループにクラスタリングすると、多くの場合、共通の構成を共有する必要があります。適切な構成管理によって、コンポーネントが同じ受信リクエストに対して同一のレスポンスを返したり、これらのコンポーネントの構成を同期化することが可能になります。また、管理目的の停止時間を短縮する可用性の高い構成管理を実現できます。

- 状態のレプリケーションとルーティング

ステートフル・アプリケーションでは、これらのリクエストを処理しているプロセスに障害が発生した場合、リクエストのステートフル・フェイルオーバーを有効にするためにクライアント状態をレプリケートできます。

- サーバーのロード・バランシングとフェイルオーバー

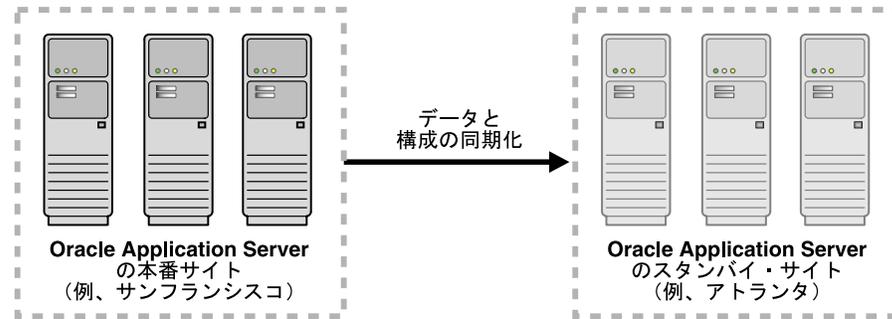
同じサーバー・コンポーネントの複数のインスタンスを使用できる場合、これらのコンポーネントに対するクライアント・リクエストは、各インスタンスのワークロードがほぼ同一になるようにロード・バランシングされます。ロード・バランシング・メカニズムが利用されると、インスタンスは冗長になります。いずれかのインスタンスに障害が発生した場合、そのインスタンスに対するリクエストを、障害の発生していないインスタンスに送信できます。

- バックアップとリカバリ

ユーザー・エラーによってシステムが誤作動することがあります。特定の状況下では、コンポーネントまたはシステムの障害が修復不可能なこともあります。そのため、バックアップとリカバリの機能を使用してシステムを一定の間隔でバックアップし、修復不可能な障害が発生したときにバックアップをリストアできるようにする必要があります。

障害時リカバリ・ソリューションでは通常、2つの同種のサイト（1つはアクティブ・サイト、もう1つはパッシブ・サイト）を設定します。サイトはそれぞれが自己完結型のシステムです。一般的に、アクティブ・サイトを本番サイト、パッシブ・サイトをスタンバイ・サイトと呼びます。通常の運用では、本番サイトがリクエストを処理しますが、サイトでフェイルオーバーやスイッチオーバーが発生すると、スタンバイ・サイトが本番サイトを引き継ぎ、すべてのリクエストはそのサイトにルーティングされます。フェイルオーバーのためにスタンバイ・サイトを保持するには、スタンバイ・サイトに同種のインストールとアプリケーションを含めるだけでなく、データと構成を本番サイトからスタンバイ・サイトに常時同期化する必要もあります。

図 1-2 地理的に分散された障害時リカバリ



## 1.2 Oracle Application Server の高可用性の概要

Oracle Application Server の高可用性の概要については、次の各項で説明します。

- [第 1.2.1 項「用語」](#)
- [第 1.2.2 項「Oracle Application Server の基本アーキテクチャ」](#)
- [第 1.2.3 項「Oracle Identity Management と Oracle Application Server 10g リリース 3 \(10.1.3.1.0\) の使用」](#)
- [第 1.2.4 項「Oracle Application Server の高可用性アーキテクチャ」](#)
- [第 1.2.5 項「最適な高可用性アーキテクチャの選択」](#)

### 1.2.1 用語

次に示す用語の定義は、このマニュアルで説明されている概念の理解に役立ちます。

- **フェイルオーバー**：高可用性システムのメンバーに予期しない障害（計画外停止）が発生した場合、コンシューマへのサービス提供を継続するために、フェイルオーバー操作が実行されます。アクティブ / パッシブ・システムの場合、フェイルオーバー操作によってパッシブ・メンバーがアクティブになり、コンシューマは障害が発生したメンバーからそのメンバーにリダイレクトされます。フェイルオーバー操作は、手動で実行できます。また、ハードウェア・クラスタ・サービスを設定して自動化できます。これにより、障害を検出したり、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・ノードへ移行させることが可能になります。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ロード・バランサのエントリが、アクティブ・メンバーにリクエストを配信することにより実現されます。アクティブ・メンバーに障害が発生した場合には、ロード・バランサが障害を検出し、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・メンバーに自動的にリダイレクトします。
- **フェイルバック**：システムがフェイルオーバー操作に成功した後、障害が発生した元のメンバーが修復されて、スタンバイ・メンバーとしてシステムに再導入されます。フェイルバック・プロセスを開始してこのメンバーをアクティブにしてから、他のメンバーを非アクティブにすることもできます。このプロセスによって、システムが障害発生前の構成に戻されます。
- **ハードウェア・クラスタ**：ハードウェア・クラスタとはコンピュータの集まりであり、ネットワーク・サービス（例：IP アドレス）またはアプリケーション・サービス（例：データベース、Web サーバー）のシングル・ビューをこれらのサービスのクライアントに提供します。ハードウェア・クラスタ内の各ノードは、それぞれのプロセスを実行するスタンドアロン・サーバーです。これらのプロセスは相互に通信して、アプリケーション、システム・リソースおよびデータを協調してユーザーに提供する単一のシステムのように機能します。

ハードウェア・クラスタは、専用のハードウェア（クラスタ・インターコネクト、共有記憶域）とソフトウェア（ヘルス・モニター、リソース・モニター）を使用して、高可用性とスケーラビリティを実現します（クラスタ・インターコネクトは、ハートビート情報がノードの停止を検出する際にハードウェア・クラスタで使用されるプライベート・リンクです）。専用のハードウェアとソフトウェアが必要であるため、通常、ハードウェア・クラスタは Sun 社、HP 社、IBM 社、Dell 社などのハードウェア・ベンダーによって提供されます。1つのハードウェア・クラスタ内で構成可能なノード数は、ベンダーによって異なりますが、Oracle Application Server の高可用性では2つのノードのみを必要とします。そのため、このドキュメントでは、2つのノードを持つハードウェア・クラスタを高可用性ソリューションで使用することを前提とします。

- **クラスタ・エージェント**：ハードウェア・クラスタのノード・メンバー上で実行されるソフトウェアで、可用性とパフォーマンスの操作を他のノードと協調させます。クラスタウェアは、リソースのグループ化やモニタリング、およびサービスを移行させる機能を提供します。クラスタ・エージェントによって、サービスのフェイルオーバーを自動化できます。
- **クラスタウェア**：クラスタ・メンバーの処理を、1つのシステムとして管理するソフトウェアです。これにより、クラスタ・メンバー間のハートビート・メカニズムを通じて、監視対象のリソースとサービスのセットを定義したり、これらのリソースとサービスをクラスタ内の別のメンバーにできるだけ効率的かつ透過的な方法で移動することができます。
- **共有記憶域**：ハードウェア・クラスタ内の各ノードが、各自のプロセス・セットを実行するスタンドアロン・サーバーの場合でも、通常、クラスタ対応のために必要とされる記憶域サブシステムは共有されます。共有記憶域とは、ハードウェア・クラスタ内のいずれかのノードから同じ記憶域（通常はディスク）にアクセスできるクラスタの機能のことです。両ノードはこの記憶域に対して等しくアクセスできますが、一方のノード（1次ノード）が記憶域に対して常に頻繁にアクセスします。1次ノードで障害が発生すると、ハードウェア・クラスタのソフトウェアは、2次ノードがこの記憶域にアクセスすることを許可します。

OracleAS Cold Failover Cluster（中間層）環境では、Oracle ホーム・ディレクトリは共有記憶域システムにインストールするか、ハードウェア・クラスタ内の各ノードのローカル記憶域にインストールできます。

- **1次ノード**：常に Oracle Application Server をアクティブに実行しているノードです。このノードで障害が発生すると、Oracle Application Server は2次ノードにフェイルオーバーされます。1次ノードではアクティブな Oracle Application Server インストールが実行されるため、1次ノードはホット・ノードとみなされます。この項の2次ノードの定義を参照してください。
- **2次ノード**：これは、1次ノードに障害が発生した場合に Oracle Application Server を実行するノードです。それまで2次ノードでは Oracle Application Server が実行されていないため、2次ノードはコールド・ノードとみなされます。アプリケーションがホット・ノード（1次）からコールド・ノード（2次）へフェイルオーバーされるため、このタイプのフェイルオーバーはコールド・フェイルオーバーと呼ばれます。この項の1次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（UNIX の場合）、`C:\WINDOWS\system32\drivers\etc\hosts` ファイル（Windows の場合）または DNS 解決のいずれかで指定される、IP アドレスに割り当てられる名前です。この名前は、マシンが接続しているネットワークで参照可能です。多くの場合、ネットワーク・ホスト名と物理ホスト名は同じものを指します。しかし、各マシンが持つ物理ホスト名は1つのみですが、ネットワーク・ホスト名は複数持つことができます。したがって、マシンのネットワーク・ホスト名が常に物理ホスト名になるわけではありません。
- **物理ホスト名**：このマニュアルでは、物理ホスト名とネットワーク・ホスト名の用語が区別されます。このマニュアルでは、物理ホスト名は、現行マシンの内部名を表すために使用されます。UNIX では、`hostname` コマンドにより返される名前です。

物理ホスト名は、Oracle Application Server により、ローカル・ホストを参照する名前として使用されます。物理ホスト名はインストール時に、インストーラにより現行マシンから取得され、ディスク上の Oracle Application Server 構成メタデータに格納されます。

- **スイッチオーバー**: 通常の運用時に、システムのアクティブ・メンバーのメンテナンスまたはアップグレードが必要になることがあります。スイッチオーバー・プロセスを開始すると、メンテナンスまたはアップグレードが必要な計画停止中のメンバーによって処理されているワークロードを、他のメンバーに引き継がせることができます。スイッチオーバー操作によって、システムのコンシューマにサービスを継続して提供できます。
- **スイッチバック**: スwitchオーバー操作が実行されると、メンテナンスまたはアップグレードが必要なシステムのメンバーが非アクティブになります。メンテナンスまたはアップグレードが完了すると、システムがスイッチバック操作を行い、アップグレードしたメンバーをアクティブにして、スイッチオーバー前のシステム構成に戻します。
- **仮想ホスト名**: 仮想ホスト名はネットワーク・アドレスで参照可能なホスト名で、ロード・バランサまたはハードウェア・クラスタを使用して1つ以上の物理マシンにマッピングされます。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、一連のサーバーを表す仮想ホスト名を持つことができ、クライアントはこの仮想ホスト名を使用してマシンと間接的に通信します。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 に割り当てられるネットワーク・ホスト名です。クラスタの仮想 IP は、クラスタの特定のノードに永続的に割り当てられるものではないため、仮想ホスト名も特定のノードに永続的に割り当てられません。

---

**注意:** このドキュメントで「仮想ホスト名」という用語を使用する場合、それが仮想 IP アドレスに関連付けられることを前提とします。単に IP アドレスを必要とする場合または使用する場合には、その旨を明示的に記載します。

---

- **仮想 IP**: クラスタの仮想 IP、ロード・バランサの仮想 IP とも呼ばれます。一般に、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ることができます。ネットワーク・クライアントに対してクラスタのシングル・システム・ビューを提供する際には、仮想 IP がクラスタのメンバーであるサーバー・グループのエントリ・ポイントとして機能します。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ることができます。

ハードウェア・クラスタは、クラスタの仮想 IP を使用して、外部に対するクラスタへのエントリ・ポイントを示します (スタンドアロン・マシン上にも設定できます)。ハードウェア・クラスタのソフトウェアは、外部クライアントによるこの IP アドレスへの接続中、クラスタの2つの物理ノード間でのこの IP アドレスの移動を管理します。その際、クライアントは、この IP アドレスが現在どの物理ノードでアクティブであるかを知る必要はありません。2つのノードを持つ標準的なハードウェア・クラスタ構成の場合、各マシンは固有の物理 IP アドレスと物理ホスト名を持ちますが、複数のクラスタ IP アドレスを持つことができます。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。現在、クラスタ IP アドレスの所有権を持っているノードが、そのアドレスのアクティブ・ノードになります。

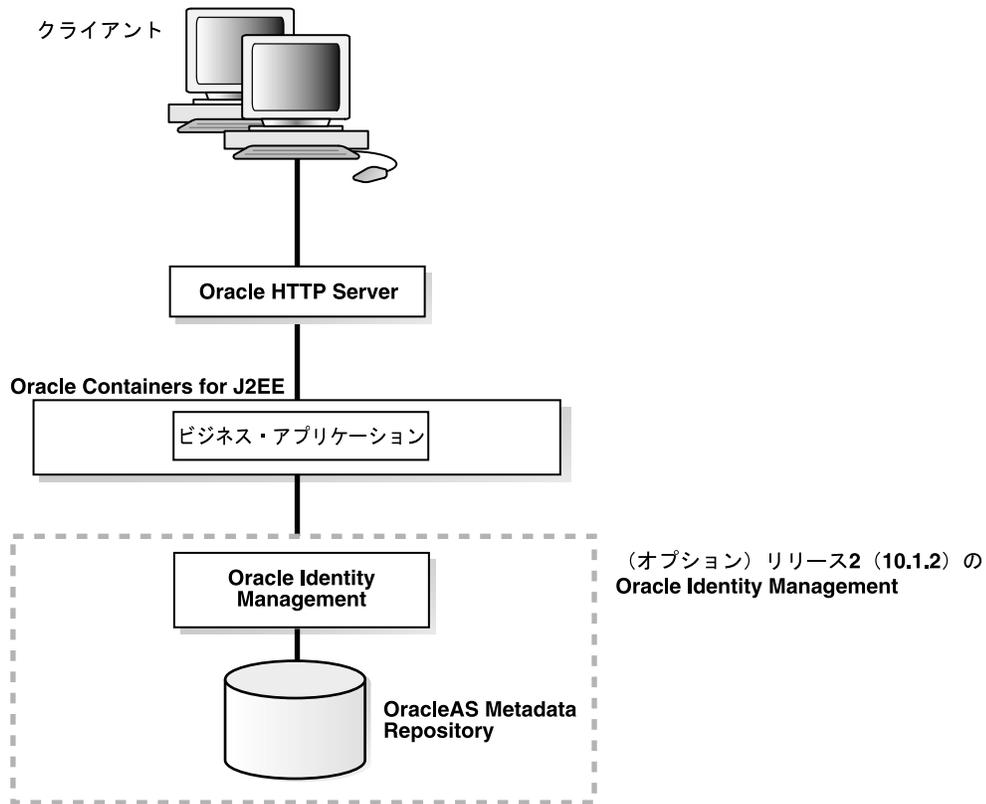
また、ロード・バランサも一連のサーバーのエントリ・ポイントとして仮想 IP を使用します。これらのサーバーは、同時にアクティブになる傾向があります。この仮想 IP アドレスは個々のサーバーに割り当てられるのではなく、サーバーとクライアントの間のプロキシとして機能するロード・バランサに割り当てられます。

## 1.2.2 Oracle Application Server の基本アーキテクチャ

可用性の高い Oracle Application Server のインストールを作成する前に、Oracle Application Server の基本アーキテクチャについて理解する必要があります。次に、Oracle Application Server の高可用性を実現するために、すべてのコンポーネントおよびコンポーネント間の接続経路を調べて、それらの可用性を向上させます。これにより、基本アーキテクチャを冗長にすることで、高可用性アーキテクチャが実現します。

図 1-3 に、Oracle Application Server の基本アーキテクチャを示します。

図 1-3 Oracle Application Server の基本アーキテクチャ



高レベルでは、Oracle Application Server は Oracle HTTP Server と Oracle Containers for J2EE (OC4J) で構成されます。OC4J は、ビジネス・アプリケーションをデプロイする J2EE コンテナを提供します。

必要に応じて、リリース 10.1.4、リリース 2 (10.1.2) またはリリース 9.0.4 の Oracle Identity Management を使用することもできます。詳細は、次の項を参照してください。

## 1.2.3 Oracle Identity Management と Oracle Application Server 10g リリース 3 (10.1.3.1.0) の使用

このリリースの Oracle Application Server には、J2EE 中間層のみが含まれています。Oracle Identity Management または OracleAS Metadata Repository は含まれていません。

ビジネス・アプリケーションで、Oracle Identity Management コンポーネントが提供するサービスが必要な場合は、リリース 3 (10.1.3.1.0) の J2EE 中間層にビジネス・アプリケーションをデプロイし、実行できます。それによって、これらのアプリケーションからリリース 10.1.4、リリース 2 (10.1.2.x) またはリリース 9.0.4 の Oracle Identity Management にアクセスできるようになります。図 1-3 を参照してください。

Oracle Identity Management は、ユーザー認証、許可および ID 情報を管理します。これは、次の主要コンポーネントで構成されます。

- OracleAS Single Sign-On
- Oracle Delegated Administration Services
- Oracle Internet Directory
- Oracle Directory Integration and Provisioning
- OracleAS Certificate Authority

可用性の高い環境では、中間層、Oracle Identity Management および OracleAS Metadata Repository のすべてにおいて高度な可用性を実現する必要があります。このガイドでは、高可用性 J2EE 中間層の設定方法について説明します。高可用性 Oracle Identity Management および OracleAS Metadata Repository の設定方法の詳細は、使用している Oracle Identity Management リリースの『Oracle Application Server 高可用性ガイド』を参照してください。

OracleAS Metadata Repository データベースは Oracle Identity Management リリースに関連付けられていますが、OracleAS Metadata Repository データベースは DCM またはリリース 3 (10.1.3.1.0) のインスタンス構成情報の格納には使用されません。リリース 3 (10.1.3.1.0) には DCM は含まれません。OracleAS Metadata Repository は、Oracle Identity Management コンポーネントのみに使用されます。

## 1.2.4 Oracle Application Server の高可用性アーキテクチャ

Oracle Application Server は、柔軟性の高いインストール、配置およびセキュリティ・オプションとともに、ローカル高可用性、あらゆる障害に対して最大限の保護を提供する障害時リカバリ・ソリューションを備えています。ローカル高可用性ソリューションと障害時リカバリ・ソリューションのアーキテクチャに含まれる冗長ノードおよび冗長プロセスによって、高可用性が実現されます。

高レベルでは、Oracle Application Server のローカル高可用性アーキテクチャには複数のアクティブ/アクティブおよびアクティブ/パッシブ・アーキテクチャが含まれています。どちらのソリューションを使用しても高可用性は実現できますが、一般的にはアクティブ/アクティブ・ソリューションのほうがスケーラビリティに優れ、フェイルオーバーも迅速です。その一方で、より多くのコストがかかるというデメリットもあります。アクティブ/アクティブとアクティブ/パッシブのどちらのカテゴリでも、インストールの容易さ、コスト、スケーラビリティおよびセキュリティが異なる複数のソリューションがあります。

ローカル高可用性ソリューションの上部には、Oracle Application Server Disaster Recovery ソリューションである Oracle Application Server Guard が構築されます。このユニークなソリューションでは、Oracle Database の実績のある Oracle Data Guard テクノロジがアプリケーション分野の高度な障害時リカバリ・テクノロジと統合されます。これにより、アプリケーション・システム全体における包括的な障害時リカバリ・ソリューションが提供されます。このソリューションには、同種の本番サイトとスタンバイ・サイトが必要ですが、障害時リカバリ設定のインスタンスに影響しない範囲で、それぞれのサイトに他の Oracle Application Server インスタンスをインストールすることもできます。これら 2 つのサイトの同質性を確保するには、両サイト間で構成とデータを定期的に同期化する必要があります。

## 1.2.5 最適な高可用性アーキテクチャの選択

世界のすべてのシステムに通用する、ただ1つの最適な高可用性ソリューションは存在しませんが、ご使用のシステムに最適なソリューションを選択することはできます。高可用性システムを設計するうえで最も重要な決定は、ビジネスまたはアプリケーションのサービス・レベル要件に基づく最適な高可用性アーキテクチャまたは冗長性タイプの選択であるといえます。高可用性の実装にかかるコストは、そのレベルによって異なるため、ビジネスの可用性の要件を理解することは非常に重要です。

Oracle Application Server には、様々なサービス・レベル要件に合せた数多くの高可用性ソリューションが用意されています。ご使用のアプリケーションにとって最も包括的なソリューションが最適であるとはかぎりません。最適な高可用性アーキテクチャの選択には、ビジネスのサービス・レベル要件をまず理解することが不可欠です。

次に、高可用性アーキテクチャを決定するのに役立つ高レベルの質問を示します。

1. ローカル高可用性: 本番システムは 24 時間 365 日利用する必要がありますか。
2. スケーラビリティ: 複数のアクティブな Oracle Application Server インスタンスでスケーラビリティを実現する必要がありますか。
3. サイト間の障害時リカバリ: この機能は必要ですか。

これらの質問に対する回答に基づいて、次の2つの項目で選択を行う必要があります。

1. インスタンスの冗長性: 基本、アクティブ / アクティブまたはアクティブ / パッシブ
2. サイト間の障害時リカバリ対応アーキテクチャ: Y または N

表 1-3 に、ビジネス要件に基づくアーキテクチャの選択を示します。

表 1-3 サービス・レベル要件とアーキテクチャの選択

ビジネス要件			選択するアーキテクチャ	
ローカル高可用性	スケーラビリティ	障害時リカバリ	インスタンスの冗長性	障害時リカバリ
N	N	N	基本	N
Y	N	N	アクティブ / パッシブ	N
N	Y	N	アクティブ / アクティブ	N
N	N	Y	基本	Y
Y	Y	N	アクティブ / アクティブ	N
Y	N	Y	アクティブ / パッシブ	Y
N	Y	Y	アクティブ / アクティブ (中間層) 基本 (Oracle Identity Management) <sup>1</sup>	Y
Y	Y	Y	アクティブ / アクティブ (中間層) アクティブ / パッシブおよびアクティブ / アクティブ (Oracle Identity Management) <sup>1</sup>	Y

<sup>1</sup> Oracle Identity Management は、リリース 10.1.4 またはリリース 2 (10.1.2) で提供されているものです。OracleAS Disaster Recovery は、Oracle Identity Management の基本、アクティブ / パッシブおよびアクティブ / アクティブのアーキテクチャをサポートしています。基本、アクティブ / パッシブまたはアクティブ / アクティブ・アーキテクチャのスケーラビリティを向上させるには、高性能の CPU やメモリーの追加によって Infrastructure ハードウェアの処理能力を上げてください。

次の段落では、リリース 10.1.4 またはリリース 2 (10.1.2) の Oracle Identity Management について説明します。

中間層と Oracle Identity Management には、それぞれ別の高可用性アーキテクチャを選択できますが、ローカル高可用性および障害時リカバリの要件は同一である必要があります。また、スケーラビリティの要件は、中間層と Oracle Identity Management では別に評価する必要があります。Oracle Identity Management では通常、処理するリクエストが少ないため、中間層ほどのスケーラビリティは必要とされません。

このようなスケーラビリティ要件の違いのため、配布するアーキテクチャは中間層と Oracle Identity Management とで異なる場合があります。たとえば、ローカル高可用性、サイト間の障害時リカバリ、スケーラブルな中間層と Oracle Identity Management の基本的なスケーラビリティを組み合わせた配置が必要であるとします。この場合は、アクティブ / アクティブの中間層、アクティブ / パッシブの Oracle Identity Management、および本番サイトのすべての中間層と Oracle Identity Management 構成をミラー化するスタンバイ障害時リカバリ・サイトを配置することになります。

## 1.3 他のドキュメントの高可用性に関する情報

表 1-4 に、高可用性情報を含む Oracle Application Server のガイド（このガイド以外）を示します。この情報は、Oracle Application Server の各種コンポーネントの高可用性に関連するものです。

**表 1-4 Oracle Application Server ドキュメントの高可用性に関する情報**

コンポーネント	情報の場所
Oracle Installer	Oracle Application Server のインストール・ガイドの高可用性環境におけるインストールの章
OracleAS Backup and Recovery Tool	『Oracle Application Server 管理者ガイド』のバックアップとリストアの部
Oracle Process Manager and Notification Server (OPMN)	『Oracle Process Manager and Notification Server 管理者ガイド』
OC4J	『Oracle Containers for J2EE 構成および管理ガイド』 『Oracle Containers for J2EE サービス・ガイド』 『Oracle Containers for J2EE Enterprise JavaBeans 開発者ガイド』
Oracle HTTP Server (OC4J プロセスのロード・バランシング)	『Oracle HTTP Server 管理者ガイド』



---

---

# Oracle Application Server 高可用性フレームワーク

第1章では一般的な高可用性の概要について説明しましたが、この章では高可用性トポロジで重要な Oracle Application Server の機能について説明します。この章の項は次のとおりです。

- 第2.1項「OPMNでのプロセス管理」
- 第2.2項「状態情報のレプリケーション」
- 第2.3項「Oracle Application Serverでのロード・バランシング」
- 第2.4項「OracleAS Cluster」
- 第2.5項「外部ロード・バランサ」
- 第2.6項「バックアップとリカバリ」
- 第2.7項「障害時リカバリ」
- 第2.8項「高可用性トポロジ：概要」

## 2.1 OPMN でのプロセス管理

Oracle Application Server インスタンスは、クライアント・リクエストを処理する多数の様々な実行中のプロセスで構成されています。高可用性の確保とは、これらのプロセスがすべてスムーズに実行され、リクエストが処理されると同時に、予期しないハングや障害が発生しないことを意味します。

Oracle Application Server の Oracle Process Manager and Notification Server (OPMN) コンポーネントは、次のプロセス管理サービスを提供します。

- OPMN は、停止しているか、レスポンスがないか、切断されている Oracle Application Server プロセスを検出すると、そのプロセスを自動的に再起動します。OPMN は、ping または通知メソッドによってプロセスを監視します。
- OPMN では適切な順番でプロセスが開始されます。他のプロセスに依存しているプロセスは、依存先のプロセスが起動するまで起動されません。

OPMN を使用して、プロセスの起動や停止、プロセスのステータスのチェックなどのタスクを実行することもできます。詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

OPMN は、次の Oracle Application Server プロセスを管理します。

- Oracle HTTP Server
- Oracle Containers for J2EE (OC4J)
- OracleAS ログ・ローダー
- OracleAS Guard (障害時リカバリ用)
- OracleAS Port Tunnel

さらに、OPMN によって、前述のコンポーネントに依存するどのアプリケーションも暗黙的に管理されます。たとえば、J2EE アプリケーションは OPMN によって管理される OC4J の下で実行されるため、OPMN によって管理されます。

OPMN を使用して Oracle Application Server インスタンスをクラスタリングし、Oracle HTTP Server がリクエストをクラスタ内の任意の OC4J インスタンスに送信できるようにすることもできます。このクラスタリングはアクティブ / アクティブ・トポロジが必要です。詳細は、[第 3 章「アクティブ / アクティブ・トポロジ」](#)を参照してください。

OracleAS Cluster では、クラスタ内のすべての Oracle Application Server インスタンスを管理することもできます。たとえば、OPMN コマンドを 1 台のマシンで発行し、クラスタ内のすべてのローカルおよびリモートの Oracle Application Server インスタンスですべてのプロセスまたは特定のプロセス・タイプを起動できます。

OPMN は拡張可能で、カスタム・プロセス、システムの負荷、カスタム停止プロシージャ、および障害の検出と再起動のメソッドに関する情報を追加する機能を提供します。

OPMN の詳細は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

## 2.2 状態情報のレプリケーション

アプリケーションを分散する 1 つの利点は、複数の冗長プロセスすべてがクライアントからのリクエストを処理できることです。これらのプロセスの 1 つが使用不可になった場合、別のプロセスでリクエストを処理できます。

アプリケーションによっては、Oracle Application Server が、連続したリクエスト間のステータス情報を保持することが必要な場合があります。これらのリクエストの透過的なフェイルオーバーを実現するには、複数のプロセス間でこのアプリケーション状態をレプリケートする必要があります。Oracle Application Server では、アプリケーションレベルのクラスタリングによって、J2EE アプリケーションでの状態のレプリケーションが可能です。アプリケーション・クラスタでは、いくつかのプロセスが連動して、同じ J2EE アプリケーションを実行し、それによって作成された状態がレプリケートされます。これによって、クラスタ内のインスタンス間でのリクエストの透過的なフェイルオーバーが可能になります。

J2EE アプリケーションは、次の 2 つのタイプの状態情報を保持できます。

- HTTP セッション状態（サーブレットと JSP によって更新）
- ステートフル・セッション EJB 状態（ステートフル・セッション EJB インスタンスによって更新）

OracleAS Cluster (OC4J) 内では、アプリケーションレベルのクラスタリングによって、両方のタイプの状態情報のレプリケーションが実現されます。状態情報のレプリケート方法、状態情報が OC4J インスタンス間でレプリケートされる頻度およびレプリケートする状態情報は制御が可能です。表 2-1 を参照してください。

**表 2-1 状態情報のレプリケート手順**

作業内容	参照先
状態情報のレプリケート方法の設定	状態情報は、マルチキャスト、peer-to-peer またはデータベースを使用してレプリケートできます。 <ul style="list-style-type: none"> <li>■ 第 3.2.2 項「マルチキャスト・レプリケーションの設定」</li> <li>■ 第 3.2.3 項「peer-to-peer レプリケーションの設定」</li> <li>■ 第 3.2.4 項「データベースへのレプリケーションの設定」</li> </ul>
状態情報をレプリケートする頻度の指定 またはレプリケートする情報の指定	第 3.2.5 項「レプリケーション・ポリシーの設定」を参照してください。

詳細は、『Oracle Containers for J2EE 構成および管理ガイド』の「OC4J」でのアプリケーション・クラスタリング」も参照してください。

## 2.3 Oracle Application Server でのロード・バランシング

ロード・バランシングでは、複数のプロセスにリクエストを分散します。ソフトウェアまたはハードウェアのロード・バランサの機能は次のとおりです。

### ■ ロード・バランシング・アルゴリズム

アルゴリズムは、リクエストを各プロセスに割り当てる方法を指定します。最も一般的なロード・バランシング・アルゴリズムとしては、他のインスタンスと比較した場合のレスポンス時間や許容量など、そのインスタンスの重み付けされたプロパティに基づいた単純なラウンドロビンまたは割り当てがあります。

通常、アルゴリズムにはローカルのインスタンスとリモートのインスタンスのプロセスが含まれています。これによって、あるコンポーネントからリモート・マシンで実行されている別のコンポーネントにリクエストを送信できます。

### ■ 障害の検出

ロード・バランサは、1 つ以上のプロセスに対するリクエストに障害が発生すれば、そのリクエストを検出でき、それ以降はそれらのプロセスにリクエストが転送されないように、該当プロセスを非アクティブとしてマークできることが必要です。

Oracle Application Server では、一部のコンポーネント間のリクエストのルーティングでロード・バランシング・メカニズムが使用されます。表 2-2 に、例を示します。

表 2-2 コンポーネント間のリクエストのルーティング

ルーティング元	ルーティング先	説明
OracleAS Web Cache (リリース 2 (10.1.2))	Oracle HTTP Server	<p>OracleAS Web Cache では、Oracle HTTP Server を含む任意の Oracle Application Server インスタンスにリクエストをルーティングできます。</p> <p>OracleAS Web Cache は、Oracle HTTP Server から返されたレスポンスで障害を検出した場合、使用可能な Oracle HTTP Server に新しいリクエストをルーティングします。</p> <p>ルーティング・アルゴリズムは OracleAS Web Cache コンポーネントで構成します。詳細は、リリース 2 (10.1.2) の『Oracle Application Server Web Cache 管理者ガイド』を参照してください。</p>
Oracle HTTP Server	OC4J	<p>Oracle HTTP Server は、J2EE アプリケーションに対するリクエストを OC4J プロセスにルーティングします。ルーティングは Oracle HTTP Server の mod_oc4j モジュールによって実行されます。</p> <p>mod_oc4j では、OracleAS Cluster の任意の OC4J プロセスにリクエストをルーティングできます。</p> <p>Oracle HTTP Server は、使用可能な OC4J プロセスのルーティング・テーブルを維持し、実行中の OC4J プロセスにのみ新しいリクエストをルーティングします。</p> <p>ルーティング・アルゴリズムは Oracle HTTP Server 構成ファイルのディレクティブを使用して構成します。詳細は、第 3.2.12 項「<a href="#">mod_oc4j のロード・バランシング・オプションの設定</a>」を参照してください。</p>
Oracle HTTP Server	データベース	<p>Oracle HTTP Server は、PL/SQL アプリケーションに対するリクエストをデータベースにルーティングします。ルーティングを実行するのは Oracle HTTP Server の mod_plsql モジュールです。mod_plsql は、データベースの障害を検出でき、使用可能なデータベース・ノードにのみリクエストをルーティングします。</p>
OC4J	OC4J	<p>OC4J 内では、OC4J がプレゼンテーション・レイヤー・コンポーネント (サーブレットおよび JSP) からビジネス・レイヤー・コンポーネント (EJB) にリクエストをルーティングします。</p> <p>OC4J が EJB 層に対する RMI 呼出しで障害を検出した場合は、使用可能な EJB ノードに通信をフェイルオーバーします。</p>
OC4J	データベース	<p>OC4J ドライバは、データベース・ノードの障害を検出し、使用可能なノードにリクエストを再ルーティングできます。</p>

## 2.4 OracleAS Cluster

Oracle Application Server は、様々なタイプのクラスタリングをサポートしています。この項では、OracleAS Cluster について説明します。アプリケーションレベルのクラスタリングの詳細は、[第 2.2 項「状態情報のレプリケーション」](#)を参照してください。

OracleAS Cluster 内の Oracle Application Server インスタンスはグループ化できます。OracleAS Cluster には次のような利点があります。

- アクティブ / アクティブ・トポロジでは、1 つのクラスタ内の Oracle Application Server インスタンスをグループ化し、1 つのトポロジの Oracle HTTP Server インスタンスが、同じトポロジ内にある他の OC4J インスタンスを認識できるようにする必要があります。これらのインスタンスは、Oracle HTTP Server がアプリケーション・リクエストの転送先として選択できる OC4J インスタンスです。

Oracle HTTP Server が mod\_oc4j モジュールに指定されているロード・バランシング・アルゴリズムを適用するとき、アルゴリズムはクラスタ内の OC4J インスタンスに適用されます。mod\_oc4j ロード・バランシング・アルゴリズムを設定するには、[第 3.2.12 項「mod\\_oc4j のロード・バランシング・オプションの設定」](#)を参照してください。

- Oracle Application Server インスタンスは、opmnctl コマンドの @cluster パラメータを使用して一括管理できます。たとえば、1 つのクラスタに属しているすべてのインスタンスの Oracle HTTP Server を停止するには、次のコマンドを使用します。

```
> opmnctl @cluster stopproc ias-component=HTTP_Server
```

@cluster パラメータを受け入れる opmnctl コマンドの一覧は、『Oracle Process Manager and Notification Server 管理者ガイド』を参照してください。

- 動的な peer-to-peer レプリケーションでアプリケーションレベルのクラスタリングを使用する場合は、OracleAS Cluster によってクラスタのメンバーが判断されるため、OracleAS Cluster を設定する必要があります。

peer-to-peer レプリケーションの詳細は、[第 3.2.3 項「peer-to-peer レプリケーションの設定」](#)を参照してください。

## 2.5 外部ロード・バランサ

アクティブ / アクティブ・トポロジの多数の Oracle Application Server インスタンス間でリクエストをロード・バランシングするには、外部ロード・バランサを使用する必要があります。

複数の Oracle Application Server インスタンスがクラスタ化されており、それらの前にロード・バランサが配置されている場合、ロード・バランサはシステムのエントリ・ポイントとなることで、複数のインスタンス構成を隠します。リクエストはクラスタ内のどのインスタンスでも処理できるので、外部ロード・バランサは、クラスタ内の任意の Oracle Application Server インスタンスにリクエストを送信できます。システムの能力は、追加の Oracle Application Server インスタンスを導入することにより、高めることができます。これらのインスタンスを別々のノードにインストールすれば、ノードの障害に備えて冗長性を確保できます。

Oracle Application Server とともに使用可能な外部ロード・バランサには様々なタイプがあります。[表 2-3](#) に、これらのタイプの要約を示します。

**表 2-3 外部ロード・バランサのタイプ**

ロード・バランサのタイプ	説明
ハードウェア・ロード・バランサ	ハードウェア・ロード・バランシングでは、Oracle Application Server インスタンスのグループまたは OracleAS Web Cache (リリース 2 (10.1.2)) の前にハードウェア・ロード・バランサを配置します。ハードウェア・ロード・バランサは、クライアントには透過的な方法で、Oracle HTTP Server または OracleAS Web Cache のインスタンスにリクエストをルーティングします。
ソフトウェア・ロード・バランサ	ソフトウェア・ロード・バランサでは、1 つのアプリケーション・サーバーに対する複数のコールをインターセプトして、これらのリクエストを冗長なコンポーネントにルーティングするプロセスが使用されます。

表 2-3 外部ロード・バランサのタイプ (続き)

ロード・バランサのタイプ	説明
Linux 用 LVS ネットワーク・ロード・バランサ	一部の Linux オペレーティング・システムでは、オペレーティング・システムを使用してネットワーク・ロード・バランシングを実行できます。
Windows ネットワーク・ロード・バランサ (Oracle Application Server の Windows バージョンに適用)	一部の Windows オペレーティング・システムでは、オペレーティング・システムを使用してネットワーク・ロード・バランシングを実行できます。たとえば、Microsoft Advanced Server では、NLB 機能を使用して、同じ仮想 IP アドレスまたは MAC アドレスを共有している異なるマシンにリクエストを送信できます。これらのサーバー自体をオペレーティング・システム・レベルでクラスタリングする必要はありません。

#### 外部ロード・バランサの要件

オラクル社は、外部ロード・バランサを提供していません。他社の外部ロード・バランサを使用できます。

ご使用の外部ロード・バランサが Oracle Application Server と連動することを確認するには、外部ロード・バランサが表 2-4 に示されている要件を満たすことをチェックしてください。

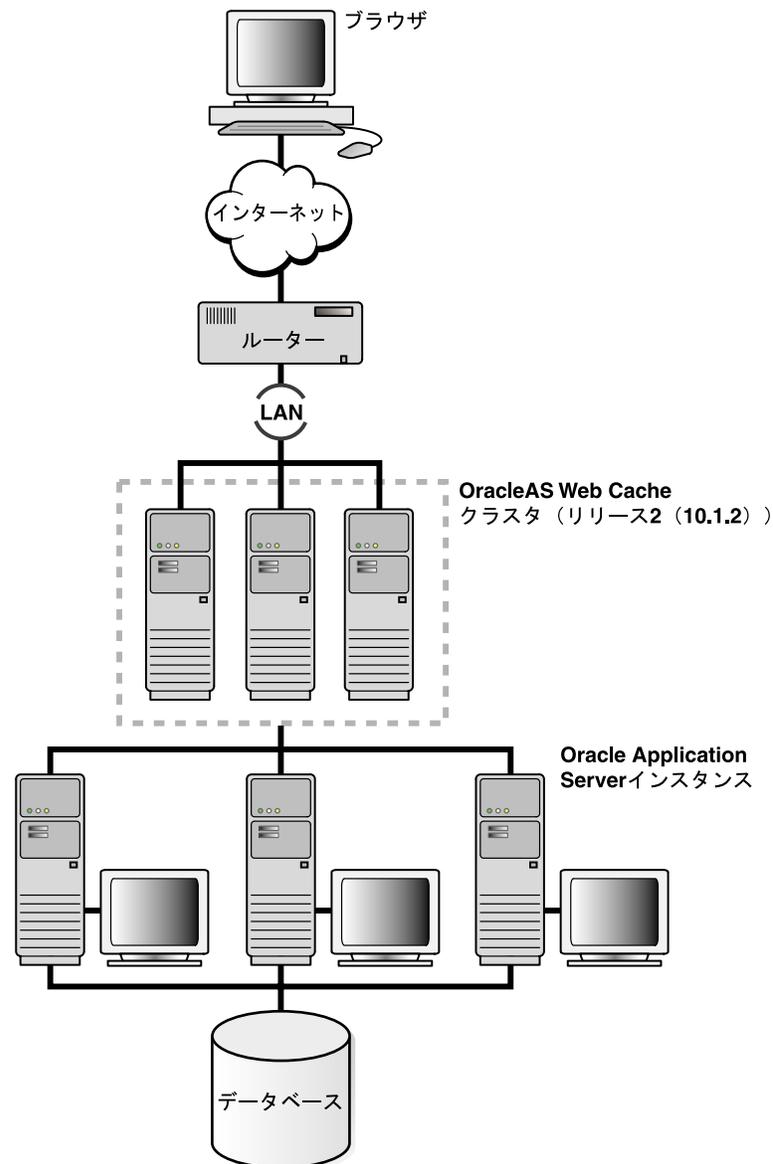
この表に示されている要件の一部を満たす必要がない場合があります。外部ロード・バランサの要件は、考慮されるトポロジと、ロード・バランシングが実行される Oracle Application Server コンポーネントによって異なります。

表 2-4 外部ロード・バランサの要件

外部ロード・バランサの要件	説明
仮想サーバーおよびポートの構成	外部ロード・バランサで仮想サーバー名とポートを構成できる必要があります。また、仮想サーバー名とポートは次の要件を満たす必要があります。 <ul style="list-style-type: none"> <li>ロード・バランサで複数の仮想サーバーの構成を許可する必要があります。各仮想サーバーについて、ロード・バランサでは複数のポート上のトラフィック管理を構成できることが必要です。たとえば、OracleAS Cluster では、ロード・バランサは HTTP および HTTPS のトラフィックに対する仮想サーバーとポートを使用して構成する必要があります。</li> <li>仮想サーバー名は IP アドレスに関連付けられていて、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・バランサにアクセスできることが必要です。</li> </ul>
リソースの監視、ポートの監視およびプロセス障害の検出	通知などの方法を使用してサービスおよびノードの障害を検出するように、また Oracle Net 以外のトラフィックを障害の発生したノードに送ることを停止するように、外部ロード・バランサを設定する必要があります。ご使用の外部ロード・バランサに、障害の自動検出機能がある場合は、それを使用してください。
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・モードになるように構成することを強くお勧めします。
その他	トラフィックの転送先のバックエンド・サービスが使用できない場合は、コールド元のクライアントにただちに戻すようロード・バランサの仮想サーバーを構成することを強くお勧めします。この構成は、クライアント・マシンの TCP/IP 設定に基づいて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。

図 2-1 は、Oracle Application Server でのハードウェア・ロード・バランシング・ルーターの配置例を示しています。

図 2-1 Oracle Application Server でのロード・バランシング・ルーターの配置例



ロード・バランシングでは、アクセス・ポイントの提供によってスケーラビリティが向上します。このアクセス・ポイントを通じて、使用可能な多数のインスタンスのいずれかにリクエストがルーティングされます。外部ロード・バランサが機能しているグループにインスタンスを追加すれば、ユーザーの追加に対応できます。

ロード・バランシングでは、最も使用可能度が高いインスタンスにリクエストをルーティングすることにより、可用性が向上します。あるインスタンスが停止した場合、または著しくビジーである場合、外部ロード・バランサは別のアクティブなインスタンスにリクエストを送信できます。

## 2.6 バックアップとリカバリ

システム・コンポーネントのデータ損失からの保護は、高可用性環境の保持に不可欠です。環境内のすべての Oracle Application Server インスタンスを定期的にバックアップしてください。

Oracle Application Server 環境の完全なバックアップには、次のものが含まれます。

- 中間層の Oracle ホームにあるすべてのファイルの全体バックアップ (Oracle ソフトウェア・ファイルおよび構成ファイルを含む)
- 環境内の各ホストにある Oracle システム・ファイルの全体バックアップ
- 以前のリリースの Oracle Identity Management を使用している場合は、次の全体バックアップも実行する必要があります。
  - Oracle Identity Management の Oracle ホームのすべてのファイル (Oracle ソフトウェア・ファイルと構成ファイルを含む)
  - Oracle Identity Management によって使用される OracleAS Metadata Repository

### 2.6.1 Oracle Application Server Backup and Recovery Tool

Oracle インストールで最も頻繁に変更される重要なファイルは、構成ファイルとデータ・ファイルです。Oracle Application Server には、これらのファイルをバックアップするための OracleAS Backup and Recovery Tool が用意されています。

OracleAS Backup and Recovery Tool を使用すると、次のインストール・タイプをバックアップおよびリカバリできます。

- J2EE サーバー
- Oracle HTTP Server
- 統合サーバー (J2EE サーバーと Oracle HTTP Server)

OracleAS Backup and Recovery Tool は、Oracle Application Server をインストールしたときにデフォルトでインストールされます。これは、ORACLE\_HOME/backup\_restore ディレクトリにインストールされます。

OracleAS Backup and Recovery Tool の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。

## 2.7 障害時リカバリ

障害時リカバリとは、自然災害またはそれ以外の災害によるサイトの壊滅的な障害からシステムをどのように回復できるのかを指します。さらに、システムの計画停止の管理方法も障害時リカバリに含まれます。障害時リカバリを必要とするほとんどの状況において、障害の解決には個々のハードウェアやサブコンポーネントのレプリケーションだけでなく、サイト全体を対象としたレプリケーションを行う必要があります。これは、OracleAS Disaster Recovery ソリューションについても同様です。

最も一般的な構成では、本番サイトをミラー化するスタンバイ・サイトを作成します。通常の運用では、本番サイトがクライアント・リクエストを積極的に処理します。スタンバイ・サイトは、本番サイトがホスティングするアプリケーションおよびコンテンツをミラー化するために維持されています。

## 2.7.1 Oracle Application Server Guard

OracleAS Guard によって、対応するスタンバイ・サイトで本番サイトのリストアが自動的に行われます。完全な Oracle Application Server 環境を災害から保護するために、OracleAS Guard によって次の操作が実行されます。

- スタンバイ・サイトのインスタンス化: プライマリ・ファームをミラー化する Oracle Application Server スタンバイ・ファームをインスタンス化します。
- 構成の検証: ファームが、対応するプライマリ・ファームのスタンバイ・ファームとして使用されるための要件を満たしていることを検証します。
- サイトの同期化: 本番サイトとスタンバイ・サイトを同期化します。

**関連項目:** 第 6.4 項「OracleAS Guard と asgctl の概要」

## 2.8 高可用性トポロジ：概要

Oracle Application Server は、同じワークロードに対して複数のインスタンスをサポートすることによって冗長性を実現します。これらの冗長構成では、分散ワークロードまたはフェイルオーバー設定、あるいはその両方によって高可用性を実現します。

Oracle Application Server では、Oracle Application Server システムのエントリ・ポイント（コンテンツ・キャッシュ）からバック・エンド・レイヤー（データソース）までの、リクエストが通過するすべての層を冗長に構成できます。構成は、OracleAS Cluster を使用したアクティブ / アクティブ構成にすることも、OracleAS Cold Failover Cluster を使用したアクティブ / パッシブ構成にすることもできます。

次の項では、これらの構成の基礎について説明します。

- 第 2.8.1 項「アクティブ / アクティブ・トポロジ」
- 第 2.8.2 項「アクティブ / パッシブ・トポロジ: OracleAS Cold Failover Cluster」

### 2.8.1 アクティブ / アクティブ・トポロジ

Oracle Application Server は、そのすべてのコンポーネントに対してアクティブ / アクティブ冗長モデルを提供します。アクティブ / アクティブ・トポロジでは、複数の Oracle Application Server インスタンスが同じワークロードを処理するように構成されます。これらのインスタンスは、同一マシンまたは異なるマシンで実行できます。

これらのインスタンスの前には外部ロード・バランサが配置され、それによって任意のアクティブなインスタンスにリクエストが送信されます。外部ロード・バランサのかわりにソフトウェア・ロード・バランサを実行して、リクエストを分散することもできます。ただし、本番環境ではハードウェア・ロード・バランサの使用をお勧めします。

アクティブ / アクティブ・トポロジの一般的なプロパティは次のとおりです。

- 類似のインスタンス構成

複数のインスタンスで、同じワークロードまたはアプリケーションを処理する必要があります。一部の構成プロパティにはインスタンス間で類似の値を指定して、これらのインスタンスが同じリクエストに対して同一のレスポンスを返すことができるようにする必要があります。他の構成プロパティには、ローカル・ホスト名情報などのようにインスタンス固有のものもあります。

1 つのインスタンスの構成を変更する場合は、アクティブ / アクティブ・トポロジ内の他のインスタンスにも同一の変更を行ってください。『Oracle Containers for J2EE 構成および管理ガイド』の「クラスタの構成と管理」では、レプリケートするプロパティを含むファイルの一覧を示しています。

- 独立操作

アクティブ / アクティブ・トポロジ内の 1 つの Oracle Application Server インスタンスに障害が発生した場合は、クラスタ内の他のインスタンスが引き続きリクエストを処理します。ロード・バランサは、動作中のインスタンスのみにリクエストを送信します。

アクティブ / アクティブ・トポロジの利点は次のとおりです。

- 可用性の向上
 

アクティブ / アクティブ・トポロジは、冗長構成です。他のインスタンスが同じリクエストの処理を続行できるので、1つのインスタンスの停止に対する耐性があります。
- スケーラビリティとパフォーマンスの向上
 

複数の同一に構成されたインスタンスによって、異なるマシン間およびプロセス間でワークロードを共有できます。トポロジは、リクエスト数が増加するにつれて新しいインスタンスを追加することで、拡大できます。

## 2.8.2 アクティブ / パッシブ・トポロジ : OracleAS Cold Failover Cluster

Oracle Application Server は、OracleAS Cold Failover Cluster のすべてのコンポーネントに対してアクティブ / パッシブ・モデルを提供します。OracleAS Cold Failover Cluster トポロジでは、2つの Oracle Application Server インスタンスが同じアプリケーション・ワークロードを処理しますが、アクティブなインスタンスは常に1つになるように構成されます。パッシブなインスタンスはアクティブなインスタンスに障害が発生した場合にのみ実行されます（アクティブになります）。これらのインスタンスはハードウェア・クラスタ内のノードで実行されません。

OracleAS Cold Failover Cluster トポロジの一般的なプロパティは次のとおりです。

- ハードウェア・クラスタ
 

OracleAS Cold Failover Cluster トポロジでは、ハードウェア・クラスタ内の、ベンダー・クラスタウェアが実行されているマシンで Oracle Application Server を実行します。
- 共有記憶域
 

Oracle Application Server インスタンス用の Oracle ホームは、ハードウェア・クラスタ内のマシンが共有する記憶域にインストールします。

OracleAS Cold Failover Cluster トポロジのアクティブ・ノードは、Oracle ホームにアクセスできるように共有記憶域をマウントします。このノードに障害が発生した場合は、パッシブなインスタンスが共有記憶域をマウントし、同じ Oracle ホームにアクセスします。
- 仮想ホスト名
 

仮想ホスト名は、Oracle Application Server 中間層のシングル・システム・ビューをクライアントに提供します。クライアントは、仮想ホスト名を使用して Oracle Application Server 中間層にアクセスします。

仮想ホスト名は、仮想 IP に関連付けられています。この名前と IP のエントリーは、サイトが使用する DNS に追加する必要があります。たとえば、ハードウェア・クラスタの2つの物理ホスト名が `node1.mycompany.com` と `node2.mycompany.com` であるとき、このクラスタのシングル・ビューを `apps.mycompany.com` という仮想ホスト名で示すことができます。DNS では、`apps` はハードウェア・クラスタを経由して `node1` と `node2` 間で浮動する仮想 IP アドレスにマップします。クライアントは `apps.mycompany.com` を使用して Oracle Application Server にアクセスします。どの物理ノードがアクティブになっていて、特定のリクエストを実際に処理しているかはクライアントにはわかりません。

インストール時には仮想ホスト名を指定できます。Oracle Application Server のインストール・ガイドを参照してください。
- フェイルオーバー・プロシージャ
 

アクティブ / パッシブ構成には、アクティブ・インスタンスの障害を検出し、停止時間を最小限に抑えてパッシブ・インスタンスにフェイルオーバーするための一連のスク립トとプロシージャも含まれています。

OracleAS Cold Failover Cluster トポロジの利点は次のとおりです。

- 可用性の向上

アクティブ・インスタンスになんらかの理由で障害が発生したか、オフラインにする必要がある場合はいつでも、同一に構成されたパッシブ・インスタンスが引き継ぐことができます。

- 運用コストの削減

アクティブ / パッシブ・トポロジでは、起動していてリクエストを処理しているのは1つのセットのプロセスのみです。通常、1つのアクティブ・インスタンスの管理には、一連のアクティブ・インスタンスの管理ほど手間がかかりません。

- アプリケーションの独立性

アクティブ / アクティブ・トポロジに適していないアプリケーションもあります。アプリケーションの状態や、ローカルに保存されている情報に対する依存度が高いアプリケーションなどです。アクティブ / パッシブ・トポロジでは、リクエストを処理しているインスタンスは常に1つのみです。



# 第 II 部

---

## 中間層の高可用性

この部は、中間層の高可用性について説明する章で構成されています。これらの章は次のとおりです。

- 第3章「アクティブ / アクティブ・トポロジ」
- 第4章「アクティブ / パッシブ・トポロジ」
- 第5章「Oracle SOA Suite の高可用性」



---

---

## アクティブ / アクティブ・トポロジ

この章では、アクティブ / アクティブ・トポロジについて説明します。この章の項は次のとおりです。

- 第 3.1 項「アクティブ / アクティブ・トポロジについて」
- 第 3.2 項「アクティブ / アクティブ・トポロジの管理」
- 第 3.3 項「Oracle HTTP Server および OC4J における高可用性機能の要約」
- 第 3.4 項「その他のトピック」

## 3.1 アクティブ/アクティブ・トポロジについて

アクティブ/アクティブ・トポロジは、単一インスタンスよりも優れたスケーラビリティと可用性を実現する冗長な Oracle Application Server インスタンスで構成されています。アクティブ/アクティブ・トポロジを使用すると、単一インスタンスで発生するシングル・ポイント障害を回避できます。単一の Oracle Application Server インスタンスでは、単一ホストのリソースを利用しますが、Oracle Application Server インスタンスのクラスタでは複数のホストにわたって、多数の CPU にアプリケーションの実行を分散できます。単一の Oracle Application Server インスタンスは、ホストおよびオペレーティング・システムの障害に対して脆弱ですが、アクティブ/アクティブ・トポロジではオペレーティング・システムまたはホストの損失が発生しても引き続き機能します。クライアントがこれらの障害の影響を受けることはありません。

アクティブ/アクティブ・トポロジでは、すべてのインスタンスが同時にアクティブになります。これは、一度に1つのみのインスタンスがアクティブになるアクティブ/パッシブ・トポロジとは異なります。

アクティブ/アクティブ・トポロジのノードは、ハードウェア・クラスタには含まれていません。

### ロード・バランサの要件

アクティブ/アクティブ・トポロジではロード・バランサを使用して、トポロジ内のいずれかの Oracle Application Server インスタンスにリクエストを送信します。つまり、Oracle Application Server インスタンスの前にはロード・バランサが配置されています。

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名を使用して構成します。クライアントはリクエストの中でこれらの仮想サーバー名を使用します。ロード・バランサは、使用可能な Oracle Application Server インスタンスにリクエストを送信します。

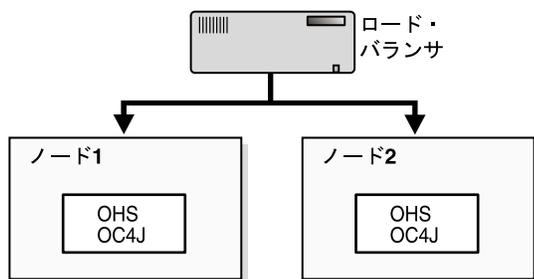
ロード・バランサに必要な機能の一覧は、[第 2.5 項「外部ロード・バランサ」](#)を参照してください。

### アクティブ/アクティブ・トポロジの図

次の図は、2つのアクティブ/アクティブ・トポロジを示しています。これらのトポロジの違いは、Oracle HTTP Server と OC4J を同じ Oracle ホームにインストールするか、個別の Oracle ホームにインストールするかです。

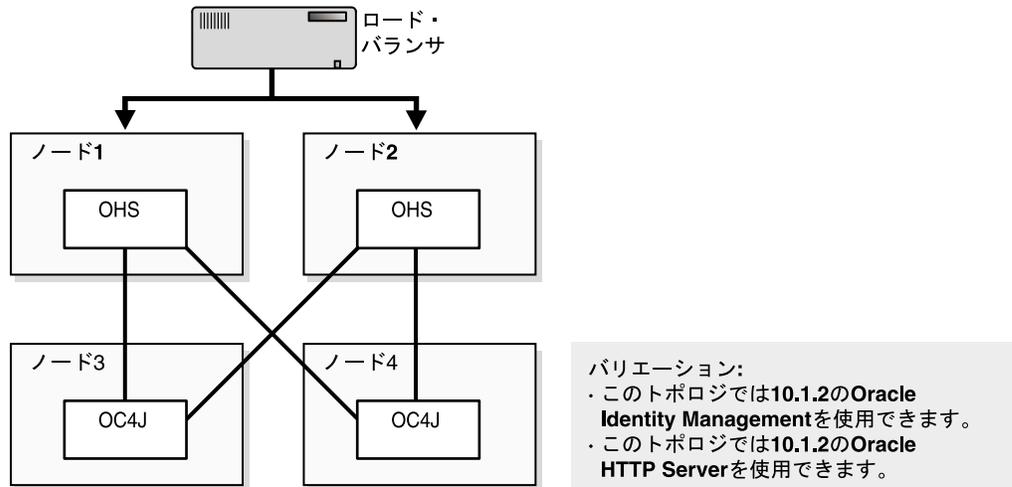
[図 3-1](#) は、Oracle HTTP Server と OC4J が同じ Oracle ホームにインストールされているアクティブ/アクティブ・トポロジを示しています。[図 3-2](#) は、Oracle HTTP Server と OC4J が個別の Oracle ホームにインストールされているアクティブ/アクティブ・トポロジを示しています。

図 3-1 同じ Oracle ホームに Oracle HTTP Server と OC4J がインストールされているアクティブ/アクティブ・トポロジ



バリエーション: このトポロジでは10.1.2の Oracle Identity Managementを使用できます。

図 3-2 個別の Oracle ホームに Oracle HTTP Server と OC4J がインストールされているアクティブ/アクティブ・トポロジ



### 3.1.1 アクティブ/アクティブ・トポロジの OracleAS Cluster

アクティブ/アクティブ・トポロジ内のすべての Oracle Application Server インスタンスは、同じクラスタに属しています。Oracle HTTP Server は、アプリケーションのリクエストを、同じクラスタに属している OC4J インスタンスのみに転送します。

クラスタ内のインスタンスは、次の方法のいずれかを使用してグループ化できます。

- すべてのインスタンスで同一のマルチキャスト IP アドレスとポートを使用します。
- すべてのインスタンスを同一の検出サーバーに連鎖します。
- 各インスタンスで、opmn.xml 構成ファイル内のその他すべてのインスタンスを指定します。
- 異なるサブネットにあるノードでインスタンスが実行される場合は、ゲートウェイ・サーバーとなるノードを指定する必要があり、このサーバーによって異なるサブネット上のインスタンスをブリッジします。

また、OracleAS Cluster を使用すると、一部の opmnctl コマンドで @cluster パラメータを使用できるようになります。@cluster パラメータを使用するコマンドは、クラスタ内のすべてのインスタンスに適用されます。たとえば、@cluster パラメータを使用して、クラスタ内のすべてのインスタンスのコンポーネントをすべて起動できます。

クラスタ内の OC4J インスタンスには、次の特徴があります。

- OC4J インスタンスには、クラスタレベルのプロパティと、インスタンス固有のプロパティがあります。クラスタレベルのプロパティとは、クラスタ内のすべての OC4J インスタンスに対して値が同様となるプロパティです。インスタンス固有のプロパティとは、OC4J インスタンスごとに値が異なるプロパティです。クラスタレベルのプロパティの一覧は、『Oracle Containers for J2EE 構成および管理ガイド』の「クラスタの構成と管理」を参照してください。
- 1つの OC4J インスタンスでクラスタレベルのプロパティを変更した場合は、クラスタ内のその他すべての OC4J インスタンスにも変更を伝播する必要があります。
- アプリケーションを OC4J インスタンスにデプロイするときは、クラスタ内のその他すべての OC4J インスタンスにもデプロイする必要があります。

- OC4J プロセスの数は、インスタンス固有のプロパティです。これは OC4J インスタンスごとに異なる場合があります。これは、クラスタ内の Oracle Application Server インスタンスごとに構成する必要があります。OC4J プロセス構成は、柔軟性があり、ホスト固有のハードウェア性能に応じて調整できます。デフォルトでは、各 OC4J インスタンスは単一の OC4J プロセスを持つようにインスタンス化されます。

詳細は、『Oracle Containers for J2EE 構成および管理ガイド』の「クラスタの構成と管理」を参照してください。

### 3.1.2 アクティブ/アクティブ・トポロジにおけるアプリケーションレベルのクラスタリング

OC4J で実行されているステートフル Web アプリケーションとステートフル・セッション EJB に対し、クライアントは一連の HTTP リクエストとレスポンスにおいて同じ OC4J プロセスを使用して通信します。ただし、アプリケーションを実行している OC4J プロセスが停止またはハングするか、ノードに障害が発生した場合、クライアントのリクエストに関連付けられている状態は失われます。

このようなソフトウェアやハードウェアの障害から保護するには、次の手順を実行する必要があります。

- 複数のノードで OC4J インスタンスを実行します。
- 同じ OracleAS Cluster (OC4J) 内の OC4J インスタンスをクラスタリングします。
- アプリケーションレベルでアプリケーションをクラスタリングします。

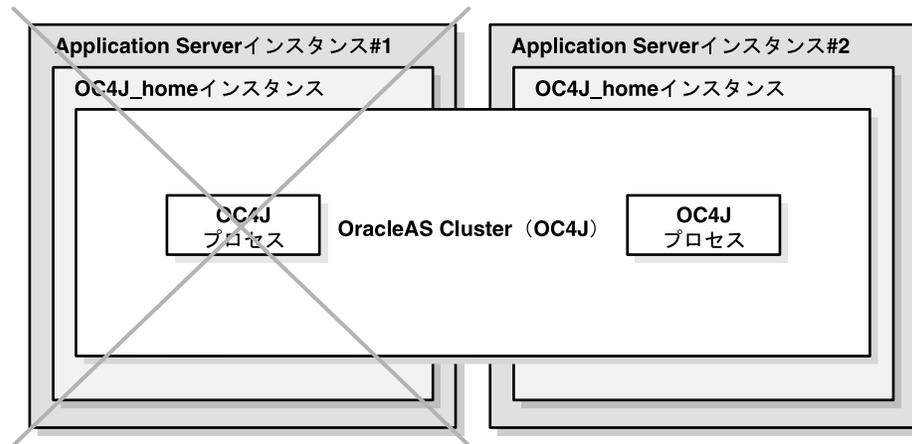
OracleAS Cluster (OC4J) のアプリケーションレベルのクラスタリングでは、OC4J プロセスがセッション状態を互いにレプリケートします。この構成では、異なる Oracle Application Server インスタンスで実行している複数の OC4J プロセス間で状態をレプリケートすることにより、フェイルオーバーと高可用性が実現します。障害が発生した場合、Oracle HTTP Server は、OracleAS Cluster (OC4J) 内のアクティブな OC4J プロセスにリクエストを転送します。

ノードの障害などのハードウェア障害から保護するには、1つの OracleAS Cluster (OC4J) 内の各ノード上の OC4J インスタンスをクラスタリングします。1つの OracleAS Cluster (OC4J) 内の異なるノードで実行される OC4J プロセスは、セッションの状態情報を共有できます。OC4J インスタンスに障害が発生するか、使用不可になると、Oracle HTTP Server は使用可能な OC4J プロセスにリクエストを転送します。Oracle HTTP Server は、クラスタ内のアクティブな（動作中の）OC4J プロセスにのみリクエストを転送します。

OracleAS Cluster (OC4J) を mod\_oc4j リクエスト・ルーティングとともに使用すると、ソフトウェアまたはハードウェアに問題が発生したときにステートフル・フェイルオーバーを利用できます。たとえば、OracleAS Cluster (OC4J) の一部である OC4J プロセスで障害が発生すると、OPMN によってこの障害が mod\_oc4j に通知され、リクエストは mod\_oc4j によって同じクラスタ内の別の OC4J プロセスにルーティングされます。クライアントのセッション状態は維持されます。サービスの損失はクライアントからは見えません。

図 3-3 は、2つの Oracle Application Server インスタンスに構成されている OracleAS Cluster (OC4J) を示しています。OC4J プロセスは各インスタンスで実行されています。Oracle Application Server のインスタンス 1 に障害が発生した場合、インスタンス 2 の OC4J プロセスはセッション状態情報を含んでいるので、リクエストを処理できます。この構成により、OracleAS Cluster (OC4J) における Web アプリケーションのセッション状態レプリケーションのフェイルオーバーが可能になります。

図 3-3 OracleAS Cluster (OC4J) における Web アプリケーションのセッション状態のフェイルオーバー



#### 必要なインスタンスとプロセスの最小数

最小の OC4J プロセス数を維持しつつ、ソフトウェアまたはハードウェア障害から保護するには、同じクラスタ内に少なくとも 2 つの OC4J プロセスを構成する必要があります。たとえば、インスタンス 1 とインスタンス 2 の 2 つの Oracle Application Server インスタンスがある場合は、それぞれのインスタンスに 2 つの OC4J プロセスを構成できます。この構成では、ステートフル・セッションのアプリケーションがハードウェアおよびソフトウェア障害から保護され、次のいずれかのタイプの障害が発生しても、クライアントの状態は維持されます。

- どちらかの OC4J プロセスに障害が発生した場合、クライアントのリクエストは同じ Oracle Application Server インスタンス内の他の OC4J プロセスにリダイレクトされます。状態は維持され、クライアントが異常に気付くことはありません。
- Oracle Application Server のインスタンス 1 が異常終了した場合、クライアントは Oracle Application Server のインスタンス 2 の OC4J プロセスにリダイレクトされます。状態は維持され、クライアントが異常に気づくことはありません。

#### 3.1.2.1 OracleAS Cluster (OC4J) によるステートフル・セッション EJB の状態レプリケーション

---

**注意：** 高可用性を目的とする EJB レプリケーション (OracleAS Cluster (OC4J-EJB)) の使用は、OracleAS Cluster (OC4J) に依存せず、OracleAS Cluster (OC4J) の内部または外部にノード間でインストールされた複数の Oracle Application Server インスタンスを使用できます。

---

OracleAS Cluster (OC4J-EJB) では、ステートフル・セッション EJB の高可用性が実現します。EJB クラスタでは、同じマルチキャスト・アドレス経由で通信する複数の OC4J プロセス間でこれらの EJB をフェイルオーバーできます。このように、ステートフル・セッション EJB でレプリケーションを使用すると、プロセスとノードの障害から保護し、Oracle Application Server で実行されているステートフル・セッション EJB の高可用性を実現できます。

詳細は、『Oracle Containers for J2EE Enterprise JavaBeans 開発者ガイド』の第 24 章「OC4J の EJB アプリケーション・クラスタリング・サービスの構成」を参照してください。

### 3.1.3 アクティブ/アクティブ・トポロジの Oracle Application Server インスタンスのプロパティ

ロード・バランサはトポロジ内の任意の Oracle Application Server インスタンスにリクエストを送信できるため、どのインスタンスがリクエストを処理してもクライアントが同じレスポンスを取得するように、各インスタンスの構成を同じにする必要があります。これには次の作業が含まれます。

- トポロジ内の各 OC4J インスタンスに同じアプリケーションをデプロイします。
- 1つの OC4J インスタンスに障害が発生した場合でも、別の OC4J インスタンスに状態情報が含まれ、その状態情報を含む OC4J インスタンスでセッションを継続できるように、状態およびステートフル・セッション Bean 情報をすべての OC4J インスタンスにレプリケートします。
- トポロジ内のすべての OC4J インスタンスの構成プロパティが同一になるようにします。これらの構成プロパティは、『Oracle Containers for J2EE 構成および管理ガイド』の第 8 章「クラスタの構成と管理」、クラスタにおける変更のレプリケーションに関する項に記載されています。

### 3.1.4 グループについて

OC4J インスタンスをグループに配置することで、いくつかの一般的な管理タスクをグループ・レベルで実行できます。たとえば、Application Server Control の「グループ」ページで、グループ内のすべてのメンバーに対して次の作業を実行できます。

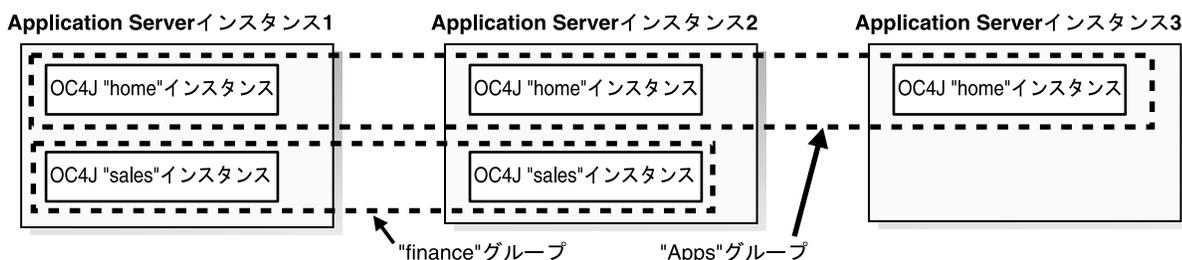
- プロセス管理操作（起動、停止、再起動など）
- デプロイ操作（デプロイ、アンデプロイ、再デプロイなど）
- JDBC 管理操作（JDBC データソースと接続プールの作成、変更、削除など）
- JMS プロバイダ操作（JMS 宛先の作成と削除、JMS 接続ファクトリの作成、変更、削除など）

「グループ」ページを表示するには、「クラスタ・トポロジ」ページの「グループ」セクションでグループ名をクリックします。

各 OC4J インスタンスは、グループに属している必要があります。Oracle Application Server をインストールすると、「default\_group」というグループが自動的に作成されます。

図 3-4 に、Oracle Application Server の 3 つのインスタンスを示します。これら Oracle Application Server の各インスタンスには、「home」という OC4J インスタンスがあり、「Apps」というグループに属しています。この図は、3 つの Oracle Application Server インスタンスのうち 2 つに「sales」という OC4J インスタンスが存在することも示しており、これらの OC4J インスタンスは「finance」というグループに属しています。

図 3-4 OC4J インスタンスのグループ



グループには、クラスタ内のすべての Oracle Application Server インスタンスの OC4J インスタンスが含まれている必要はありません。たとえば、[図 3-4](#) では、"sales" OC4J インスタンスは 2 つの Oracle Application Server インスタンスにのみ存在します。これは有効です。

Oracle Application Server では、グループ内のインスタンスを同一に構成する必要はありません。ただし、一部の構成（データソース、JMS リソース、セキュリティ・プロバイダ設定など）は同一にして、どのインスタンスがリクエストを処理しても、同じレスポンスがアプリケーションから返されるようにしてください。

グループの詳細は、Application Server Control のオンライン・ヘルプにある OC4J インスタンスとグループの作成のガイドラインに関する項を参照してください。

### 3.1.4.1 グループの作成

グループは、Application Server Control を使用して作成できます。「クラスタ・トポロジ」ページには「グループ」セクションがあり、グループの起動、停止、削除、作成などの操作を実行できます。

OC4J インスタンスの作成時にグループを作成することもできます。Application Server Control を使用して OC4J インスタンスを作成する場合、OC4J インスタンス名の入力に加えて、この新しい OC4J インスタンスがどのグループに属しているかを指定する必要があります。既存のグループまたは新しいグループのいずれかを指定できます。[第 3.1.4.2 項「追加の OC4J インスタンスの作成」](#)を参照してください。

createinstance コマンドを使用して OC4J インスタンスを作成する場合は、groupName パラメータでグループの名前を指定します。createinstance コマンドについては、[第 3.1.4.2 項「追加の OC4J インスタンスの作成」](#)を参照してください。

グループの作成手順、OC4J インスタンスの作成手順や削除手順など、グループの詳細は、『Oracle Containers for J2EE 構成および管理ガイド』の第 8 章「クラスタおよび OC4J グループの構成と管理」、Oracle Application Server クラスタ内の OC4J グループの作成と管理に関する項を参照してください。

### 3.1.4.2 追加の OC4J インスタンスの作成

追加の OC4J インスタンスを作成するには、Application Server Control または ORACLE\_HOME/bin/createinstance コマンドを使用します。

#### Application Server Control の使用

「クラスタ・トポロジ」ページで Oracle Application Server インスタンスの名前をクリックすると、「Application Server: instance\_name」ページが表示されます。「OC4J インスタンスの作成」ボタンをクリックして、「OC4J インスタンスの作成」ページを表示します ([図 3-5](#))。

「OC4J インスタンスの作成」ページでは、作成する OC4J インスタンスの名前と、その OC4J インスタンスが属するグループを入力します。既存のグループまたは新しいグループのいずれかを指定できます。

図 3-5 「OC4J インスタンスの作成」 ページ

ORACLE Enterprise Manager 10g  
Application Server Control

Cluster Topology > Application Server: basic.iasdocs-pc3.us.oracle.com >  
Create OC4J Instance

Enter name of the OC4J instance you want to create.

\* OC4J instance name

Every OC4J instance must be in a group. Select one of the following to add this OC4J instance to a group.

Add to an existing group with name  
Existing Group Name

Add to a new group with name  
New Group Name

Start this OC4J instance after creation.

**TIP** OC4J instances created with Enterprise Manager use the AJP protocol by default. After you create an OC4J instance, use the Server Properties page to verify or change the protocol and listener port for the newly created instance. For more information, see [Setting OC4J Server Properties](#)

### createinstance コマンドの使用

追加の OC4J インスタンスを作成するには、ORACLE\_HOME/bin/createinstance コマンドを使用します。構文は次のとおりです。

```
createinstance -instanceName name [-port httpPort] [-groupName groupName]
```

name には、作成する OC4J インスタンスの名前を指定します。

オプションの port パラメータは、Oracle Application Server インスタンスに Oracle HTTP Server が含まれていない場合に役立ちます。HTTP ポートを設定すると、新しい OC4J インスタンスのホームページに直接アクセスできるようになります。

オプションの groupName パラメータを使用すると、指定したグループに新しい OC4J インスタンスを追加できます。このパラメータを指定しない場合、インスタンスは default\_group グループに追加されます。指定したグループが存在しない場合は、そのグループが作成されません。

たとえば、OC4J インスタンス "sales" を作成して "finance" グループに追加するには、次のコマンドを使用します。

```
createinstance -instanceName sales -groupName finance
```

コマンドによって、"sales" インスタンスの oc4jadmin ユーザー・パスワードを設定するように求められます。このパスワードは、"home" インスタンスのパスワードとは異なるものに設定できますが、お薦めできません。パスワードを "home" インスタンスの oc4jadmin ユーザー・パスワードと同じものに設定して、OPMN の問題を回避してください。

パスワードの詳細は、『Oracle Containers for J2EE 構成および管理ガイド』の第 8 章「クラスタの構成と管理」、createinstance ユーティリティによる OC4J インスタンスの作成に関する項を参照してください。

このコマンドによってインスタンスが作成されますが、起動はされません。起動は Application Server Control コンソールまたは opmnctl コマンドで行います。

インスタンスを作成したら、OPMN をリロードし、新しいインスタンスが認識されるようにします。

```
opmnctl reload
```

createinstance コマンドの詳細は、『Oracle Containers for J2EE 構成および管理ガイド』の第8章「クラスタの構成と管理」、追加の OC4J インスタンスの作成と管理に関する項を参照してください。

### 3.1.4.3 グループ内のインスタンスの管理

OC4J インスタンスのグループは、一括管理することも、個別に管理することもできます。すべてのアプリケーションとインスタンスは、一度にまたは個別に起動および停止でき、またアプリケーションは、グループ内のすべてのインスタンスまたは一部のインスタンスにデプロイ、再デプロイ、アンデプロイすることもできます。

グループ内の一部のインスタンスのみにアプリケーションをデプロイすることはお薦めできません。グループ内のすべてのインスタンスに同じアプリケーションをデプロイするようにしてください。

グループは、Application Server Control コンソールまたは admin\_client.jar を使用して管理できます。Application Server Control コンソールでは、グループに対して次のような操作を実行できます。

表 3-1 Application Server Control コンソールを使用した、グループに対する操作の実行

操作	手順
グループ内のアプリケーションの起動、停止、デプロイ、アンデプロイ、再デプロイ	<ol style="list-style-type: none"> <li>1. Application Server Control コンソールの「クラスタ・トポロジ」ページで、下にスクロールして「グループ」セクションを表示します。</li> <li>2. 管理するグループをクリックします。「グループ:&lt;グループ名&gt;」ページが表示されます。</li> <li>3. 「アプリケーション」タブを選択します。</li> <li>4. 起動、停止、アンデプロイ、または再デプロイするアプリケーションを選択します。</li> <li>5. 「起動」、「停止」、「デプロイ」、「アンデプロイ」または「再デプロイ」ボタンをクリックします。</li> </ol>
JDBC リソースと JMS プロバイダの構成	<ol style="list-style-type: none"> <li>1. Application Server Control コンソールの「クラスタ・トポロジ」ページで、下にスクロールして「グループ」セクションを表示します。</li> <li>2. 管理するグループをクリックします。「グループ:&lt;グループ名&gt;」ページが表示されます。</li> <li>3. 「管理」タブを選択します。</li> <li>4. JDBC リソースを構成するには、JDBC Resources 行の「タスクに移動」アイコンをクリックします。「JDBC リソース」ページが表示されます。JDBC の詳細は、『Oracle Containers for J2EE サービス・ガイド』の「データソース」を参照してください。 JMS 宛先を構成するには、JMS 宛先行の「タスクに移動」アイコンをクリックします。「OracleAS JMS」ページが表示されます。JMS の詳細は、『Oracle Containers for J2EE サービス・ガイド』の「Java Message Service (JMS)」を参照してください。</li> </ol>
OC4J インスタンスの個別管理	<ol style="list-style-type: none"> <li>1. Application Server Control コンソールの「クラスタ・トポロジ」ページで、下にスクロールして「グループ」セクションを表示します。</li> <li>2. 管理するグループをクリックします。「グループ:&lt;グループ名&gt;」ページが表示されます。</li> <li>3. 「OC4J インスタンス」タブを選択します。グループ内のインスタンスがすべて表示されます。いずれかのインスタンスをクリックすると、そのインスタンスを管理できます。 <b>注意:</b> インスタンスで実行する操作は、そのインスタンスのみに影響します。操作はグループ内のすべてのインスタンスには適用されません。</li> </ol>

表 3-1 Application Server Control コンソールを使用した、グループに対する操作の実行（続き）

操作	手順
グループとしての OC4J インスタンスの管理	<ol style="list-style-type: none"> <li>1. Application Server Control コンソールの「クラスタ・トポロジ」ページで、「クラスタ MBean ブラウザ」リンクをクリックします。</li> <li>2. 左側の「J2EEServerGroup」を開きます。</li> <li>3. 「J2EEServerGroup」で、管理するグループをクリックします。</li> <li>4. 右側の「操作」タブをクリックします。</li> <li>5. グループ内の OC4J インスタンスを起動または停止するには、右側の「起動」または「停止」をクリックします。</li> <li>6. 「起動操作」をクリックして操作を実行します。</li> </ol>

### 3.1.4.4 admin\_client.jar を使用した、グループへのアプリケーションのデプロイ

admin\_client.jar ユーティリティを使用して、アプリケーションをグループにデプロイできます。構文は次のとおりです。

```
> cd ORACLE_HOME/j2ee/home
> java admin_client.jar
    deployer:cluster:opmn://<host>:<opmnPort>/<groupName>
    <adminID> <adminPassword>
    -deploy -file <pathToArchiveFile> -deploymentName <appName>
```

<host>には、グループ内の任意のホストを指定できます。

<opmnPort>には、OPMN がリスニングするポートを指定します。このポートは、opmn.xml ファイルに記載されています。

<groupName>には、グループの名前を指定します。これは、OC4J インスタンスの名前 (home など) です。

<adminID> と <adminPassword>には、管理者の ID とパスワードを指定します。通常、adminID は oc4jadmin です。

**注意：** グループ内のすべてのインスタンスに対してデプロイを有効にするには、管理者のパスワードがグループ内のすべてのインスタンスに対して同じである必要があります。

<pathToArchiveFile>には、デプロイする EAR、WAR、または JAR ファイルへのフルパスを指定します。

<appName>には、アプリケーション名を指定します。

デプロイについては、コマンドラインで指定できるオプションが他にもあります。詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

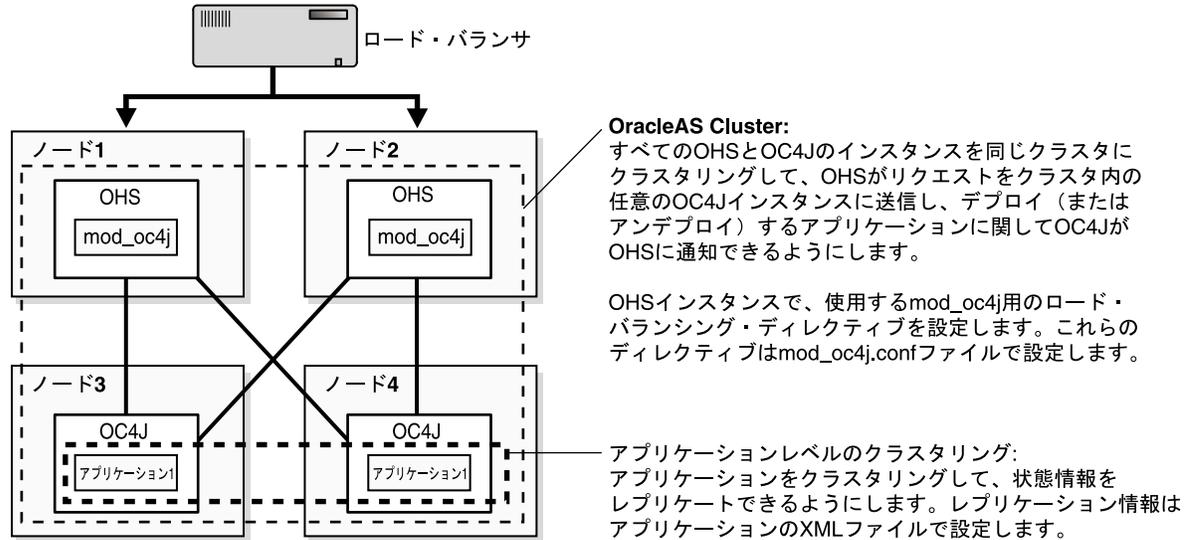
## 3.1.5 Oracle HTTP Server がリクエストを OC4J にルーティングする方法

Oracle HTTP Server では、J2EE アプリケーションに対するリクエストを受信すると、そのリクエストを Oracle HTTP Server 内にある mod\_oc4j モジュールに転送します。ロード・バランシング・アルゴリズムに従って、mod\_oc4j ではリクエストを同じクラスタ内の OC4J インスタンスに転送します。図 3-6 を参照してください。

リクエストの分散に mod\_oc4j が使用するデフォルトのロード・バランシング・アルゴリズムは、単純なラウンド・ロビン・アルゴリズムです。別のアルゴリズムを使用する場合は、mod\_oc4j.conf ファイルにディレクティブを設定します。詳細は、第 3.2.12 項「mod\_oc4j のロード・バランシング・オプションの設定」を参照してください。

セッションの一部であるリクエストは、同じ OC4J インスタンスに送信されます。最初のリクエスト後に OC4J インスタンスが使用不可になった場合は、リクエストを処理する別のインスタンスが mod\_oc4j によって検索され、同じセッション内の後続リクエストがそのインスタンスに送信されます。

図 3-6 OracleAS Cluster とロード・バランシング・ディレクティブ



### 3.1.6 アクティブ/アクティブ・トポロジでの Oracle Identity Management の使用

Oracle Application Server 10g リリース 3 (10.1.3.1.0) には Oracle Identity Management は含まれていませんが、Oracle Application Server 10g リリース 3 (10.1.3.1.0) を、リリース 9.0.4、リリース 2 (10.1.2) またはリリース 10.1.4 の Oracle Identity Management とともに使用することはできます。

Oracle Internet Directory、OracleAS Single Sign-On、OracleAS Certificate Authority、Oracle Directory Integration and Provisioning などの Oracle Identity Management サービスが必要な場合は、アクティブ/アクティブ・トポロジを Oracle Identity Management に関連付けることができます。

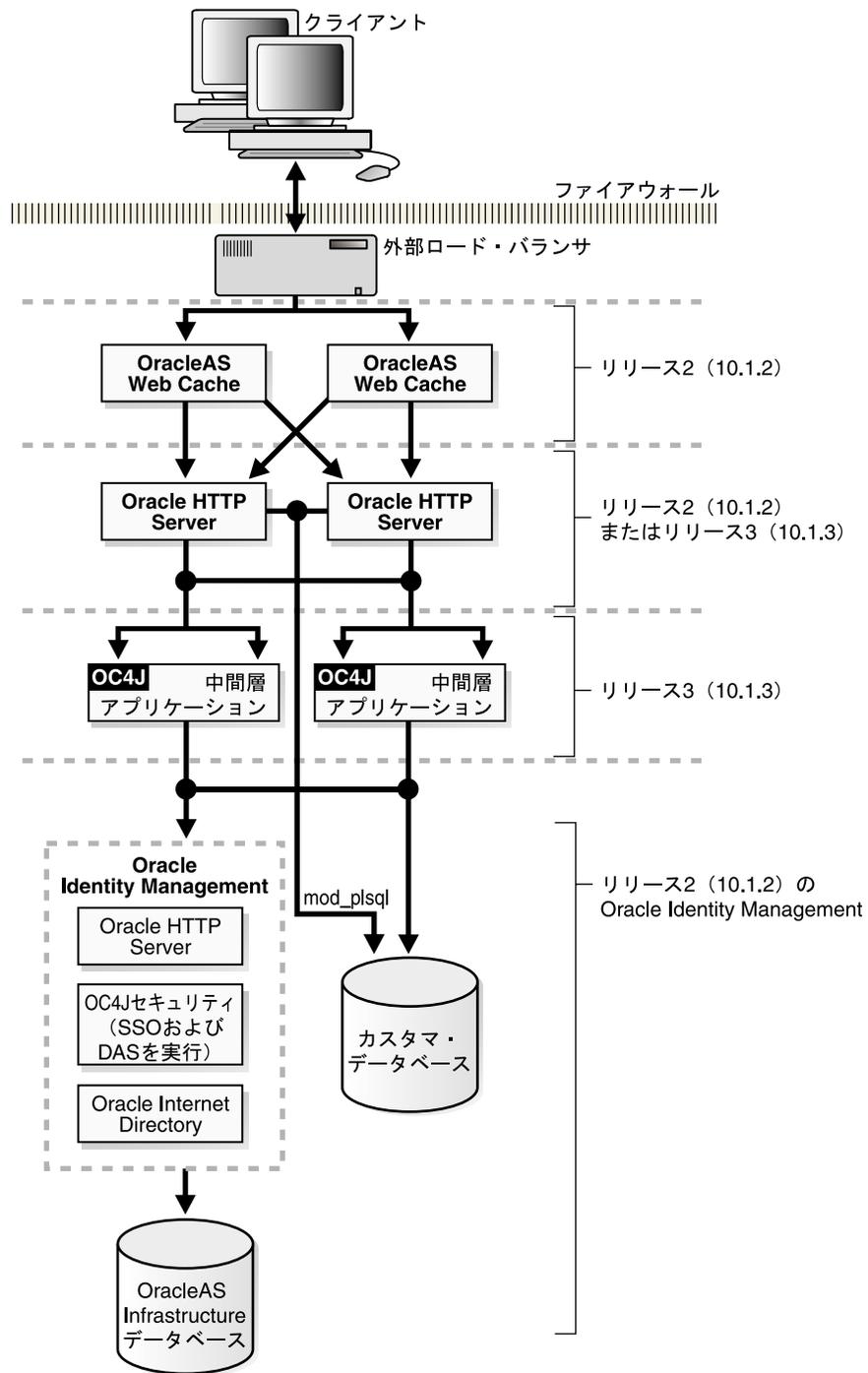
リリース 3 (10.1.3.1.0) のアクティブ/アクティブ・トポロジと Oracle Identity Management インスタンスは個別にインストールします。インストール後に、Application Server Control コンソールを使用してリリース 3 (10.1.3.1.0) のインスタンスを Oracle Identity Management の Oracle Internet Directory に関連付けます。

Oracle Identity Management をリリース 3 (10.1.3.1.0) のインスタンスに関連付ける方法の詳細は、『Oracle Application Server 管理者ガイド』の「10.1.4 および 10.1.2 の Oracle Identity Management を使用するためのインスタンスの構成」を参照してください。

高可用性環境では、高可用性を実現するために、リリース 3 (10.1.3) のインスタンスと Oracle Identity Management インスタンスの両方が必要になります。このガイドでは、リリース 3 (10.1.3) のインスタンス用の高可用性についてのみ説明します。Oracle Identity Management の高可用性の詳細は、使用している Oracle Identity Management リリースの『Oracle Application Server 高可用性ガイド』を参照してください。

図 3-7 は、リリース 2 (10.1.2) の Oracle Identity Management を使用して実行されているアクティブ/アクティブ・トポロジの Oracle Application Server 10g リリース 3 (10.1.3) のインスタンスを示しています。

図 3-7 Oracle Identity Management を使用した Oracle Application Server 中間層



### 3.1.7 アクティブ/アクティブ・トポロジでの Oracle HTTP Server 10.1.2 の使用

アクティブ/アクティブ・トポロジでは、リリース 3 (10.1.3.1.0) の Oracle HTTP Server を使用するかわりに、リリース 2 (10.1.2) の Oracle HTTP Server を使用できます。これは次のような理由で行います。

- リリース 2 (10.1.2) の Oracle HTTP Server とリリース 2 (10.1.2) の Oracle Application Server 中間層をすでに使用している場合。
- Application Server Control コンソールを使用して Oracle HTTP Server を管理する場合。リリース 2 (10.1.2) の Application Server Control コンソールを使用すると、リリース 3 (10.1.3) の Application Server Control コンソールよりも優れた Oracle HTTP Server の管理機能が提供されます。

リリース 2 (10.1.2) の Oracle HTTP Server は、分散アクティブ/アクティブ・トポロジ (図 3-2) に対してのみ使用できます。

### 3.1.8 アクティブ/アクティブ・トポロジでの OracleAS Web Cache リリース 2 (10.1.2) の使用

図 3-7 に示すように、Oracle Application Server 10g リリース 3 (10.1.3) で、リリース 2 (10.1.2) の OracleAS Web Cache を使用できます。

OracleAS Web Cache は、リバース・プロキシ・サーバーとして使用します。リバース・プロキシ・サーバーは、OracleAS Web Cache の 1 つのインスタンスまたはクラスタ (OracleAS Web Cache クラスタ) として構成されている複数のインスタンスで構成できます。

リバース・プロキシ・サーバーとしての OracleAS Web Cache (単一インスタンスまたはクラスタ) の構成方法の詳細は、『Oracle Application Server 管理者ガイド』の「リバース・プロキシとしての 10.1.2 OracleAS Web Cache の構成」を参照してください。

## 3.2 アクティブ/アクティブ・トポロジの管理

この項では、アクティブ/アクティブ・トポロジの管理に必要な一般的な手順について説明します。

- 第 3.2.1 項「OracleAS Cluster の設定」
- 第 3.2.2 項「マルチキャスト・レプリケーションの設定」
- 第 3.2.3 項「peer-to-peer レプリケーションの設定」
- 第 3.2.4 項「データベースへのレプリケーションの設定」
- 第 3.2.5 項「レプリケーション・ポリシーの設定」
- 第 3.2.6 項「レプリケート先のノードの数の指定」
- 第 3.2.7 項「コンポーネントのステータスのチェック」
- 第 3.2.8 項「トポロジ内のコンポーネントの起動と停止」
- 第 3.2.9 項「クラスタへのアプリケーションのデプロイ」
- 第 3.2.10 項「アクティブ/アクティブ・トポロジへのインスタンスの追加」
- 第 3.2.11 項「アクティブ/アクティブ・トポロジからのインスタンスの削除」
- 第 3.2.12 項「mod\_oc4j のロード・バランシング・オプションの設定」
- 第 3.2.13 項「Java Message Service (JMS) での高可用性の構成」

## 3.2.1 OracleAS Cluster の設定

OracleAS Cluster の作成には、いくつかの方法があります。この項では、2つの方法についてのみ説明します。

- 第 3.2.1.1 項「動的検出方法」
- 第 3.2.1.2 項「検出サーバー方法」

その他の方法については、『Oracle Containers for J2EE 構成および管理ガイド』の「クラスタの構成と管理」を参照してください。

### 3.2.1.1 動的検出方法

この方法では、クラスタ内の各 Oracle Application Server インスタンスに対して同じマルチキャスト・アドレスとポートを定義します。この方法の利点は、クラスタ内の各 Oracle Application Server インスタンスの名前を指定する必要がないことです。クラスタのインスタンスの追加または削除は、マルチキャスト・アドレスおよびポートを編集することで実行できます。

1. 同じクラスタにグループ化する Oracle Application Server インスタンスごとに、次のコマンドを実行します。

```
opmnctl config topology update discover="*<multicastAddress>:<multicastPort>"
```

*multicastAddress* は、このクラスタに使用するマルチキャスト・アドレスを指定します。マルチキャスト・アドレスは、224.0.1.0 ~ 239.255.255.255 の有効なアドレス範囲内で指定する必要があります。コマンドでは、マルチキャスト・アドレスの前に \* 文字を付けます。

*multicastPort* には、未使用のポート番号を指定できます。

例：

```
> ORACLE_HOME/opmn/bin/opmnctl config topology update  
discover="*225.0.0.20:8001"
```

分散インストール（異なる Oracle ホームに Oracle HTTP Server および OC4J がインストールされている場合）では、すべての Oracle Application Server インスタンスを同じクラスタにクラスタリングする必要があります。すべてのインスタンスには同じマルチキャスト IP とポートを使用する必要があります。

2. `opmnctl config topology update` コマンドを実行する Oracle Application Server インスタンスごとに `opmnctl reload` コマンドを実行し、更新された `opmn.xml` ファイルの OPMN による読取りを強制します。

```
> ORACLE_HOME/opmn/bin/opmnctl reload
```

### 3.2.1.2 検出サーバー方法

マルチキャスト方法を使用しない場合は、Oracle Application Server インスタンスを実行するノードの名前を各インスタンスの `opmn.xml` ファイルで指定して、クラスタを定義できます。

例：4つのインスタンス（`inst1.node1.mycompany.com`、`inst2.node2.mycompany.com`、`inst3.node3.mycompany.com`、`inst4.node4.mycompany.com`）をクラスタリングする場合は、次の手順を実行します。

1. 検出サーバーとして機能するインスタンスを少なくとも1つ指定します。検出サーバーは、クラスタのトポロジを保持します。

この例では、`inst1.node1.mycompany.com` と `inst2.node2.mycompany.com` がクラスタの検出サーバーになることを前提としています。

分散インストール（異なる Oracle ホームに Oracle HTTP Server および OC4J がインストールされている場合）では、Oracle HTTP Server または OC4J のどちらかを実行するインスタンスでも検出サーバーとして機能できます。

2. クラスタ内のすべてのインスタンス用の `opmn.xml` ファイルで、検出サーバーを実行しているノード（この例では `node1.mycompany.com` と `node2.mycompany.com`）を指定します。

この例では、次の行を含むように `opmn.xml` ファイルを変更します。

```
<notification-server>
  <topology>
    <discover
      list="node1.mycompany.com:6201,node2.mycompany.com:6201"/>
    </topology>
    ...
</notification-server>
```

6201 は、通知サーバーがリスニングするポート番号を示します。この値はそのインスタンスの `opmn.xml` ファイルに含まれています。

複数の検出サーバーを使用する場合は、これらをカンマで区切ります。

3. すべてのインスタンスで `opmnctl reload` を実行して、更新された `opmn.xml` ファイルの OPMN による読取りを強制します。

```
> ORACLE_HOME/opmn/bin/opmnctl reload
```

### 3.2.2 マルチキャスト・レプリケーションの設定

マルチキャスト・レプリケーションは、デフォルトのレプリケーション・タイプです。マルチキャスト・レプリケーションを使用するようにアプリケーションを設定するには、空の `<cluster/>` タグをアプリケーションの `orion-application.xml` ファイルまたはグローバルの `ORACLE_HOME/j2ee/home/config/application.xml` ファイルに追加します。次に、例を示します。

```
<orion-application ... >
  ...
  <cluster/>
</orion-application>
```

`<cluster/>` タグは、アプリケーションのデプロイ先のノードすべてに追加する必要があります。

デフォルトでは、マルチキャスト・レプリケーションには `230.230.0.1` のマルチキャスト・アドレスと `45566` のポートが使用されます。これらの値を変更する場合は、`multicast` 要素の `ip` 属性と `port` 属性に必要な値を指定します。たとえば次のコードでは、`ip` 属性と `port` 属性に、カスタマイズされた値が設定されています。

```
<orion-application ... >
  ...
  <cluster allow-colocation="false">
    <replication-policy trigger="onShutdown" scope="allAttributes"/>
    <protocol>
      <multicast ip="225.130.0.0" port="45577" bind-addr="226.83.24.10"/>
    </protocol>
  </cluster>
</orion-application>
```

マルチキャスト・アドレスには、`224.0.1.0` ～ `239.255.255.255` の範囲の値を指定する必要があります。

上のコードで使用されている他のタグと属性を次に説明します。

- `allow-colocation`: アプリケーション状態が、同じホストで実行されている他の Oracle Application Server インスタンスにレプリケートされるかどうかを指定します。デフォルトは `true` です。
- `trigger` および `scope`: 第 3.2.5 項「レプリケーション・ポリシーの設定」を参照してください。
- `bind-addr`: バインド先のネットワーク・インタフェース・カード (NIC) の IP を指定します。これは、ホスト・マシンに複数の NIC があり、各 NIC に独自の IP アドレスが指定されている場合に役立ちます。

### 3.2.3 peer-to-peer レプリケーションの設定

Oracle Application Server では、動的と静的の 2 種類の peer-to-peer レプリケーションをサポートしています。

- 動的な peer-to-peer レプリケーションでは、OC4J は OPMN を使用して他の OC4J のインスタンスを検出します。構成ファイルにインスタンス名のリストを含める必要はありません。
- 静的な peer-to-peer レプリケーションでは、レプリケーションの対象となるインスタンスの名前をリストに含めます。

#### 動的な peer-to-peer レプリケーション

動的な peer-to-peer レプリケーションを指定するには、空の `<opmn-discovery/>` タグをアプリケーションの `orion-application.xml` ファイルまたはグローバルの `ORACLE_HOME/j2ee/home/config/application.xml` ファイルに追加します。

```
<orion-application ... >
...
<cluster allow-colocation="false">
  <replication-policy trigger="onShutdown" scope="allAttributes"/>
  <protocol>
    <peer>
      <opmn-discovery/>
    </peer>
  </protocol>
</cluster>
</orion-application>
```

OPMN によるクラスタ内のインスタンスの検出方法は、OracleAS Cluster の設定時に定義しました。詳細は、第 3.2.1 項「OracleAS Cluster の設定」を参照してください。

#### 静的な Peer-to-Peer レプリケーション

静的な peer-to-peer レプリケーションを指定するには、ホスト名のリストをアプリケーションの `orion-application.xml` ファイルまたはグローバルの `ORACLE_HOME/j2ee/home/config/application.xml` ファイルの `<node>` 要素に含めます。ノードごとにアクティブ/アクティブ・トポロジー内の別のノードを指定し、トポロジー内のすべてのノードが連鎖の中で接続されるようにします。たとえば、トポロジーに 3 つの Oracle Application Server インスタンスがある場合、ノード 1 はノード 2 を指定し、ノード 2 はノード 3 を指定し、ノード 3 はノード 1 を指定できます。

例:

ノード 1 では、`<node>` タグによってノード 2 が指定されます。

```
<orion-application ... >
...
<cluster allow-colocation="false">
  <replication-policy trigger="onShutdown" scope="allAttributes"/>
  <protocol>
    <peer start-port="7900" range="10" timeout="6000">
      <node host="node2.mycompany.com" port="7900"/>
    </peer>
  </protocol>
</cluster>
</orion-application>
```

```

    </peer>
  </protocol>
</cluster>
</orion-application>

```

ノード2では、<node> タグによってノード3が指定されます。

```

<orion-application ... >
  ...
  <cluster allow-colocation="false">
    <replication-policy trigger="onShutdown" scope="allAttributes"/>
    <protocol>
      <peer start-port="7900" range="10" timeout="6000">
        <node host="node3.mycompany.com" port="7900"/>
      </peer>
    </protocol>
  </cluster>
</orion-application>

```

ノード3では、<node> タグによってノード1が指定されます。

```

<orion-application ... >
  ...
  <cluster allow-colocation="false">
    <replication-policy trigger="onShutdown" scope="allAttributes"/>
    <protocol>
      <peer start-port="7900" range="10" timeout="6000">
        <node host="node1.mycompany.com" port="7900"/>
      </peer>
    </protocol>
  </cluster>
</orion-application>

```

これを実現する別の方法は、すべてのノードで同じノードを指定することです。3つのノードを含む例では、ノード1とノード2がノード3を指定し、ノード3がノード1またはノード2のどちらかを指定することもできます。

上の例で使用されているタグと属性を次に説明します。

- **start-port**: ピア通信用に Oracle Application Server がバインドするローカル・ノード上の最初のポートを指定します。このポートが使用中の場合は、使用可能なポートが見つかるまで Oracle Application Server によってポート番号が増加されます。デフォルトは 7800 です。
- **timeout**: 指定したピア・ノードからのレスポンスを待機する時間をミリ秒で指定します。デフォルトは 3000 ミリ秒です。
- **host**: ピア・ノードの名前を指定します。
- **port**: 指定したホスト (host 属性で指定) でピア通信に使用するポートを指定します。デフォルトは 7800 です。
- **range: port** (start-port ではない) 属性で指定したポートの番号を増加する回数を指定します。デフォルトは 5 です。

次の点に注意してください。

- 静的 peer-to-peer レプリケーションでは、アプリケーションの orion-application.xml ファイルはインスタンスごとに異なります。アプリケーションをデプロイするときは、それに伴って orion-application.xml を更新する必要があります。

### 3.2.4 データベースへのレプリケーションの設定

このレプリケーション・メカニズムでは、レプリケートされたデータはデータベースに保存されます。データベースはアプリケーションの `orion-application.xml` ファイルまたはグローバルの `ORACLE_HOME/j2ee/home/config/application.xml` ファイルの `<database>` タグに指定します。次に、例を示します。

```
<orion-application ... >
...
  <cluster allow-colocation="false">
    <replication-policy trigger="onShutdown" scope="allAttributes"/>
    <protocol>
      <database data-source="jdbc/MyOracleDS"/>
    </protocol>
  </cluster>
</orion-application>
```

`data-source` 属性の値は、`data-sources.xml` ファイルに指定されているデータソースの `jndi-name` に一致する必要があります。データソースの作成と使用の詳細は、『Oracle Containers for J2EE サービス・ガイド』を参照してください。

### 3.2.5 レプリケーション・ポリシーの設定

`<replication-policy>` タグの属性を使用すると、レプリケートするデータおよびデータをレプリケートする頻度を指定できます。

#### trigger 属性

`trigger` 属性は、レプリケーションを行う時期を指定します。表 3-2 では、この属性でサポートされている値について説明しています。

表 3-2 trigger 属性の値

値	HttpSession	ステートフル・セッション Bean
<code>onSetAttribute</code>	HTTP セッション属性に行われた各変更を値の変更時にレプリケートします。プログラミングの観点では、レプリケーションは、 <code>setAttribute()</code> が <code>HttpSession</code> オブジェクトに対して呼び出されるたびに行われます。  このオプションを使用すると、セッションで大きな変更が行われる場合、多くのリソースを消費することがあります。	該当なし。
<code>onRequestEnd</code> (デフォルト)	HTTP セッション属性に行われた変更をすべてキューに格納し、HTTP レスポンスの送信直前にすべての変更をレプリケートします。	各 EJB メソッドの呼出し後に Bean の現在の状態をレプリケートします。状態を頻繁にレプリケートすることで、高い信頼性が得られます。

表 3-2 trigger 属性の値 (続き)

値	HttpSession	ステートフル・セッション Bean
onShutdown	<p>[Ctrl]+[C] などを使用して JVM が正常に終了されるたびに、HTTP セッションの現在の状態をレプリケートします。システム・クラッシュの場合など、ホストが予期せずに終了した場合には、状態はレプリケートされません。</p> <p>セッション状態は以前にレプリケートされていないため、JVM の停止時にすべてのセッション・データがネットワークに一度に送信されます。このため、ネットワークのパフォーマンスが影響を受ける場合があります。このオプションを使用すると、JVM の停止に必要な時間が大幅に長くなる場合があります。</p>	<p>JVM が正常に停止するたびに Bean の現在の状態をレプリケートします。システム・クラッシュの場合など、ホストが予期せずに停止した場合、状態はレプリケートされません。</p> <p>Bean 状態は以前にレプリケートされていないため、JVM の停止時にすべての状態データがネットワークに一度に送信されます。このため、ネットワークのパフォーマンスが影響を受ける場合があります。このオプションを使用すると、JVM の停止に必要な時間が大幅に長くなる場合があります。</p>

**scope 属性**

scope 属性は、レプリケートするデータを指定します。表 3-3 では、この属性でサポートされている値について説明しています。

表 3-3 scope 属性の値

値	HttpSession	ステートフル・セッション Bean
modifiedAttributes	<p>変更された HTTP セッション属性のみをレプリケートします。</p> <p>これは、HttpSession のデフォルトのレプリケーション設定です。</p>	該当なし。
allAttributes	<p>HTTP セッションに設定されている属性値をすべてレプリケートします。</p>	<p>ステートフル・セッション Bean に設定されているすべてのメンバー変数値をレプリケートします。</p> <p>これは、ステートフル・セッション Bean のデフォルトのレプリケーション設定です。</p>

**3.2.6 レプリケート先のノードの数の指定**

レプリケート先のノードの数を指定するには、<cluster> タグの write-quota 属性を使用します。たとえば、次のコードでは、レプリケート・データが 2 つの他のノードにレプリケートされることを指定しています。

```
<orion-application ... >
...
<cluster allow-colocation="false" write-quota="2">
  <replication-policy trigger="onShutdown" scope="allAttributes"/>
  <protocol>
    <peer>
      <opmn-discovery/>
    </peer>
  </protocol>
</cluster>
</orion-application>
```

デフォルトは 1 です。

推奨事項: 2 つのノードを含むアクティブ/アクティブ・トポロジでは、write-quota を 1 に設定し、データが他方のノードにレプリケートされるようにします。

3つ以上のノードを含むトポロジでは、`write-quota` を2以上に設定し、データが少なくともその他2つのノードにレプリケートされるようにします。

トポロジ内のノードすべてにデータをレプリケートするには、`write-quota` をトポロジ内のノードの合計数に設定します。そのノードで別のインスタンスが実行されていると、同じノードに書き込みが行われる場合があります。

データベースにレプリケートする場合、`write-quota` 属性は使用されません。

### 3.2.7 コンポーネントのステータスのチェック

アクティブ/アクティブ・トポロジのインスタンスのステータスをチェックするには、トポロジ内の任意のインスタンスから次のコマンドを実行します。

```
> cd ORACLE_HOME/opmn/bin
> opmnctl @cluster status
```

### 3.2.8 トポロジ内のコンポーネントの起動と停止

`opmnctl` コマンドを使用すると、トポロジ内のコンポーネントを起動および停止できます。トポロジ内のすべての Oracle Application Server インスタンスのコンポーネントを起動および停止するには、`opmnctl` で `@cluster` パラメータを使用する必要があります。`opmnctl` コマンドは、トポロジ内の任意のインスタンスから実行できます。

たとえば、トポロジ内のすべてのインスタンスの Oracle HTTP Server コンポーネントを起動するには、トポロジ内の任意のインスタンスから次のコマンドを実行します。

```
> cd ORACLE_HOME/opmn/bin
> opmnctl @cluster startproc ias-component=HTTP_Server
```

### 3.2.9 クラスタへのアプリケーションのデプロイ

アプリケーションは、Application Server Control コンソールを使用するか、コマンドラインから実行するコマンドを使用してデプロイできます。

アプリケーションをクラスタ内のすべてのインスタンスにデプロイする場合は、次のように `admin_client.jar` を使用できます。

```
> cd ORACLE_HOME/j2ee/home
> java admin_client.jar
    deployer:cluster:opmn://<host>:<opmnPort>/<oc4jInstanceName>
    <adminID> <adminPassword>
    -deploy -file <pathToArchiveFile> -deploymentName <appName>
```

<host> には、クラスタ内の任意のホストを指定できます。

<opmnPort> には、OPMN がリスニングするポートを指定します。このポートは、`opmn.xml` ファイルに記載されています。

<oc4jInstanceName> には、アプリケーションのデプロイ先の OC4J インスタンスを指定します。たとえば、"home" インスタンスにデプロイするには、`home` を指定します。

<adminID> と <adminPassword> には、管理者の ID とパスワードを指定します。通常、`adminID` は `oc4jadmin` です。

---

**注意：** クラスタ内のすべてのインスタンスに対してデプロイを有効にするには、管理者のパスワードがクラスタ内のすべてのインスタンスに対して同じである必要があります。

---

<pathToArchiveFile> には、デプロイする EAR、WAR、または JAR ファイルへのフルパスを指定します。

<appName>には、アプリケーション名を指定します。

デプロイについては、コマンドラインで指定できるオプションが他にもあります。詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

### 3.2.10 アクティブ/アクティブ・トポロジへのインスタンスの追加

インスタンスを既存のトポロジに追加する手順は次のとおりです。

- トポロジが使用している既存のクラスタ方法を使用します。たとえば、トポロジで動的な検出方法を使用している場合は、同じマルチキャスト IP とポートを使用するように新しいインスタンスを構成する必要があります。
- OC4J コンポーネントをクラスタリングするには、トポロジですでに使用されている既存のモデルに従います。
- 新しいインスタンスに同じアプリケーションをデプロイします。
- リクエストを新しいノードに送信するようにロード・バランサを構成します。

### 3.2.11 アクティブ/アクティブ・トポロジからのインスタンスの削除

インスタンスをトポロジから削除する手順は次のとおりです。

- ロード・バランサを再構成し、削除したインスタンスにリクエストが送信されないようにします。
- インスタンスに追加したタグを削除して、インスタンスを OracleAS Cluster から削除します。これらのタグは、クラスタの設定時に追加したものです。詳細は、[第 3.2.1 項「OracleAS Cluster の設定」](#)を参照してください。
- 次のタグを削除して、アプリケーションレベルのクラスタリングからインスタンスを削除します。
  - クラスタリングが構成されているアプリケーションの一部であるすべての Web モジュール用の web.xml ファイルの、<distributed/> タグ。
  - アプリケーションの orion-application.xml ファイルまたはグローバルの ORACLE\_HOME/j2ee/home/config/application.xml ファイルに追加した <cluster> タグ。

### 3.2.12 mod\_oc4j のロード・バランシング・オプションの設定

Oracle HTTP Server 内の mod\_oc4j モジュールは、リクエストを OC4J プロセスに委任します。Oracle HTTP Server が OC4J に対する URL のリクエストを受信すると、Oracle HTTP Server ではそのリクエストを mod\_oc4j モジュールにルーティングします。mod\_oc4j モジュールでは、このリクエストを OC4J プロセスにルーティングします。OC4J プロセスで障害が発生すると、OPMN はその障害を検出し、mod\_oc4j は、障害の発生した OC4J プロセスが再起動されるまでその OC4J プロセスにリクエストを送信しません。

mod\_oc4j は、OC4J プロセスに対するリクエストをロード・バランシングするように構成できます。Oracle HTTP Server では、mod\_oc4j を経由して、各種のロード・バランシング・ポリシーをサポートしています。ロード・バランシング・ポリシーは、ネットワーク・トポロジやホスト・マシンの性能に応じて、パフォーマンス上のメリットだけでなく、フェイルオーバーや高可用性も実現します。

mod\_oc4j には、必要なルーティングのタイプと複雑さに応じて、異なるロード・バランシング・ルーティング・アルゴリズムを指定できます。ステートレス・リクエストは、mod\_oc4j.conf で指定されたアルゴリズムに基づいて使用可能な宛先にルーティングされます。ステートフル HTTP リクエストは、セッション ID を使用して前回のリクエストを処理した OC4J プロセスに転送されます。ただし、mod\_oc4j が OPMN との通信によってそのプロセスが使用不可であると判断した場合を除きます。その場合、mod\_oc4j は、指定されたロード・バランシング・プロトコルに従って使用可能な OC4J プロセスにそのリクエストを転送します。

デフォルトでは、すべての OC4J インスタンスに同じ重みが付けられており（すべてのインスタンスに 1 の重みが付けられている）、`mod_oc4j` ではラウンド・ロビン方法を使用して、リクエストの転送先の OC4J インスタンスを選択します。OC4J インスタンスの重みは、トポロジー内の他の使用可能な OC4J インスタンスの重みに対する比率として扱われ、インスタンスが処理するリクエストの数が定義されます。リクエストが確立済のセッションに属する場合、`mod_oc4j` では、そのセッションを開始したのと同じ OC4J インスタンスおよび同じ OC4J プロセスにそのリクエストを転送します。

`mod_oc4j` のロード・バランシング・オプションでは、リクエストの送信先 OC4J インスタンスを判断するとき、OC4J インスタンス上で実行されている OC4J プロセスの数を考慮しません。OC4J インスタンスの選択は、インスタンスに対して構成済の重みと、その可用性に基づいて行われます。

`mod_oc4j` ロード・バランシング・ポリシーを変更するには、`ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf` ファイル内の `Oc4jSelectMethod` ディレクティブと `Oc4jRoutingWeight` ディレクティブを設定します。

1. 各 Oracle Application Server インスタンスの `mod_oc4j.conf` ファイルの `<IfModule mod_oc4j.c>` セクションで、`Oc4jSelectMethod` ディレクティブを表 3-4 に示す値のいずれかに設定します。

`Oc4jSelectMethod` ディレクティブを `roundrobin:weighted` または `random:weighted` に設定した場合は、`Oc4jRoutingWeight` ディレクティブも設定して重みを指定します（次の手順を参照）。

ルーティング・アルゴリズムを選択する際のヒントについては、3-23 ページの「[mod\\_oc4j のロード・バランシング・アルゴリズムの選択](#)」を参照してください。

**表 3-4 Oc4jSelectMethod の値**

値	説明
<code>roundrobin</code> (デフォルト)	<code>mod_oc4j</code> では、トポロジー内のすべての OC4J プロセスをリストに配置し、リスト内の順序に従ってプロセスを選択します。
<code>roundrobin:local</code>	<code>roundrobin</code> と似ていますが、リストにはローカルの OC4J プロセスのみが含まれています。利用可能なローカル OC4J プロセスがない場合は、リモート OC4J プロセスを選択します。
<code>roundrobin:weighted</code>	<code>mod_oc4j</code> では、各インスタンスに構成されているルーティングの重みに基づいて、各 OC4J インスタンスへのリクエストの合計ロードを分散します。次に、ラウンド・ロビン方式でローカルのインスタンスから OC4J プロセスを選択します。  この重みは、 <code>Oc4jRoutingWeight</code> ディレクティブを使用して構成します。
<code>random</code>	<code>mod_oc4j</code> では、トポロジー内のすべての OC4J プロセスのリストから OC4J プロセスをランダムに選択します。
<code>random:local</code>	<code>random</code> と似ていますが、 <code>mod_oc4j</code> ではローカルの OC4J プロセスが優先されます。利用可能なローカル OC4J プロセスがない場合は、リモート OC4J プロセスを選択します。
<code>random:weighted</code>	<code>mod_oc4j</code> では、トポロジー内の各インスタンスに構成されている重みに基づいて、OC4J プロセスを選択します。  この重みは、 <code>Oc4jRoutingWeight</code> ディレクティブを使用して構成します。
<code>metric</code>	<code>mod_oc4j</code> では、プロセスのビジュー状態を示す実行時メトリックに基づいて、リクエストをルーティングします。
<code>metric:local</code>	<code>metric</code> と似ていますが、 <code>mod_oc4j</code> ではローカルの OC4J プロセスが優先されます。使用可能なローカルの OC4J プロセスがない場合は、リモートの OC4J プロセスにルーティングされます。

例:

```
Oc4jSelectMethod random:local
```

メトリックベースのロード・バランシングの設定方法の詳細は、『Oracle HTTP Server 管理者ガイド』の付録「mod\_oc4jを使用したロード・バランシング」を参照してください。

2. Oc4jSelectMethod ディレクティブを重みベースの方法（つまり roundrobin:weighted または random:weighted）に設定した場合は、Oc4jRoutingWeight ディレクティブも設定して重みを指定します。

Oc4jRoutingWeight の構文は次のとおりです。

```
Oc4jRoutingWeight hostname weight
```

Oc4jRoutingWeight ディレクティブを設定しない場合は、デフォルトの重みの 1 が使用されます。

例: 3つのインスタンス (A、B および C) で構成されるトポロジがあり、B と C が A の 2 倍のリクエストを受信するように設定するには、すべてのインスタンスに対して次のディレクティブを設定します。

```
Oc4jSelectMethod roundrobin:weighted
Oc4jRoutingWeight hostB 2
Oc4jRoutingWeight hostC 2
```

hostA の Oc4jRoutingWeight の設定はオプションです。指定しない場合はデフォルト値の 1 が使用されるためです。

3. トポロジ内のすべてのインスタンスに対して Oracle HTTP Server を再起動し、変更を反映させます。

```
> opmnctl @cluster restartproc ias-component=HTTP_Server
```

### mod\_oc4j のロード・バランシング・アルゴリズムの選択

mod\_oc4j で使用するロード・バランシング・オプションを決定する際は、次のガイドラインが役に立ちます。

- Oracle HTTP Server および OC4J を同一の Oracle ホームで実行している、まったく同じマシンで構成されるトポロジでは、ローカル・アフィニティ・アルゴリズムによるラウンド・ロビンが適しています。この場合、同じマシン上の OC4J プロセスがすべて使用できないような極端なケースを除けば、mod\_oc4j を使用して他のマシンにリクエストをルーティングすることから得られる Oracle HTTP Server のメリットはほとんどありません。
- あるマシンのセットでは Oracle HTTP Server を実行し、別のセットではリクエストを処理する OC4J インスタンスを実行しているような、分散されたデプロイでは、単純なラウンド・ロビンおよびメトリックベースのアルゴリズムが適しています。特定の設定で 2 つのアルゴリズムのどちらがより適しているかを決定するには、それぞれを試してみて結果を比較することが必要な場合もあります。結果はシステムの動作および受信リクエストの分散によって異なるため、こうした試行が必要になります。
- 異なる特性を持つ複数のノードで、異なる Oracle Application Server インスタンスを実行している異機種間の配置では、重み付けされたラウンド・ロビンのアルゴリズムが適しています。各インスタンスの重みを設定するだけでなく、各 Oracle Application Server インスタンスで実行される OC4J プロセスの数も調整して、最大限の効果をあげるようにしてください。たとえば、重みが 4 のマシンは重みが 1 のマシンの 4 倍のリクエストを処理しますが、重みが 4 のシステムは 4 倍の数の OC4J プロセスを実行する必要があります。
- メトリックベースのロード・バランシングは、たとえば CPU、データベース接続の数など、アプリケーションのパフォーマンスを決定するメトリックが数個しかない場合に便利です。

### 3.2.13 Java Message Service (JMS) での高可用性の構成

JMS の高可用性の構成方法の詳細は、『Oracle Containers for J2EE サービス・ガイド』の第 4 章「Oracle Enterprise Messaging Service の使用」を参照してください。

## 3.3 Oracle HTTP Server および OC4J における高可用性機能の要約

表 3-5 は、Oracle HTTP Server および OC4J における高可用性機能の要約を示しています。

表 3-5 Oracle HTTP Server および OC4J における高可用性機能の要約

項目	説明
ノード障害からの保護	<p><b>Oracle HTTP Server:</b> Oracle HTTP Server インスタンスの前にデプロイされているロード・バランサは、ノードの障害からの保護を提供します。ロード・バランサには、外部ロード・バランサまたは OracleAS Web Cache (リリース 2 (10.1.2) の OracleAS Web Cache) を使用できます。</p> <p><b>OC4J:</b> mod_oc4j は、動作中の OC4J インスタンスのみにリクエストをルーティングします。OC4J インスタンスは複数のノードにインストールして実行し、常に少なくとも 1 つのノードで OC4J が動作している確率を高めるようにします。</p>
サービス障害からの保護	<p><b>Oracle HTTP Server:</b> Oracle HTTP Server インスタンスの前にデプロイされているロード・バランサは、最初の Oracle HTTP Server からレスポンスがない場合や、URL の ping の結果から Oracle HTTP Server に障害が発生したと思われる場合、別の Oracle HTTP Server にリクエストを送信します。ロード・バランサには、外部ロード・バランサまたは OracleAS Web Cache を使用できます。</p> <p><b>OC4J:</b> OPMN は OC4J プロセスを監視し、障害発生時にはそのプロセスを再起動します。また OPMN は、この再起動が成功しなかった場合、動作している OC4J プロセスにのみリクエストを送信するよう mod_oc4j に通知します。</p>
プロセス障害からの保護	<p><b>Oracle HTTP Server:</b> OPMN は Oracle HTTP Server プロセスを監視し、障害発生時にそのプロセスを再起動します。さらに、トポロジ内の別の Oracle HTTP Server プロセスで障害が発生した場合、OPMN はそれぞれの Oracle HTTP Server に障害を通知します。</p> <p><b>OC4J:</b> OPMN は OC4J プロセスを監視し、障害発生時にはそのプロセスを再起動します。また OPMN は、この再起動が成功しなかった場合、動作している OC4J プロセスにのみリクエストを送信するよう mod_oc4j に通知します。</p>
自動再ルーティング	<p><b>Oracle HTTP Server:</b> Oracle HTTP Server インスタンスの前にデプロイされているロード・バランサは、最初の Oracle HTTP Server からレスポンスがない場合、別の Oracle HTTP Server にリクエストを自動的に再ルーティングします。</p> <p><b>OC4J:</b> mod_oc4j は、最初の OC4J プロセスからレスポンスがない場合、別の OC4J プロセスにリクエストを自動的に再ルーティングします。</p>

**関連項目：**

- 第 2.1 項「OPMN でのプロセス管理」

## 3.4 その他のトピック

- 第 3.4.1 項「JNDI ネームスペースのレプリケーション」
- 第 3.4.2 項「EJB クライアント・ルーティング」
- 第 3.4.3 項「Java Object Cache を使用した OC4J の分散キャッシュ」

### 3.4.1 JNDI ネームスペースのレプリケーション

EJB クラスタリングを有効にすると、中間層 OracleAS Cluster の OC4J インスタンス間の JNDI ネームスペースのレプリケーションも有効になります。1 つの OC4J インスタンス内の JNDI ネームスペースへの新規バインドは、中間層 OracleAS Cluster 内の他の OC4J インスタンスに伝播されます。再バインドとアンバインドはレプリケートされません。

このレプリケーションは、各 OracleAS Cluster (OC4J) の有効範囲外で行われます。つまり、OC4J インスタンス内の複数の OracleAS Cluster (OC4J) は、同じレプリケート済 JNDI ネームスペースへの可視性を持ちます。

JNDI の詳細は、『Oracle Containers for J2EE サービス・ガイド』を参照してください。

### 3.4.2 EJB クライアント・ルーティング

EJB クライアント・ルーティングでは、Oracle HTTP Server とサーブレットおよび JSP 間で `mod_oc4j` が提供するルーティング機能は、EJB クラスによって実行されます。クライアントは、Remote Method Invocation (RMI) プロトコルを使用して EJB を起動します。RMI プロトコル・リスナーは、RMI 構成ファイル `rmi.xml` によって OC4J インスタンスごとに設定されます。これは Web サイト構成とは別です。EJB クライアントと OC4J ツールは、構成された RMI ポートを使用して OC4J サーバーにアクセスします。OPMN によって、RMI リスナーが使用できる一連のポートが指定されます。

EJB ルックアップで「`opmn:orimi://`」接頭辞文字列を使用すると、クライアントは割り当てられた RMI ポートを自動的に取得します。ロード・バランシングとクライアント・リクエスト・ルーティングは、使用可能な様々な OC4J プロセスを選択する OPMN によって提供されます。このロード・バランシングに使用されるアルゴリズムは、ランダム・アルゴリズムです。高可用性を確保するために、カンマで区切られた複数の OPMN URL を使用できます。

### 3.4.3 Java Object Cache を使用した OC4J の分散キャッシュ

Oracle Application Server Java Object Cache では、OC4J にデプロイされたアプリケーションに対する高可用性ソリューションとして機能する分散キャッシュを提供します。Java Object Cache は、あらゆる Java プラットフォーム上のあらゆる Java アプリケーションで使用可能な Java オブジェクトのインプロセス・キャッシュです。これにより、アプリケーションで、複数のリクエストおよびユーザー間でオブジェクトを共有し、複数のプロセス間でオブジェクトのライフサイクルを調整することが可能になります。

Java Object Cache は、同じ OracleAS Cluster (OC4J)、Oracle Application Server インスタンスまたは全般的な Oracle Application Server Cluster に属していないプロセスも含めた、OC4J プロセス間でのデータ・レプリケーションを可能にします。

Java Object Cache を使用すると、オブジェクトの生成元のアプリケーションがどれであるかわからず、共有 Java オブジェクトがローカルにキャッシュされるため、パフォーマンスが向上します。これにより、可用性も向上します。オブジェクトのソースが使用できなくなっても、ローカルにキャッシュされたバージョンは引き続き使用できます。

Java Object Cache の使用方法の詳細は、『Oracle Containers for J2EE サービス・ガイド』の「Java Object Cache」を参照してください。



---

---

## アクティブ/パッシブ・トポロジ

この章では、アクティブ/パッシブ・トポロジの構成方法と管理方法について説明します。この章の項は次のとおりです。

- 第 4.1 項「アクティブ/パッシブ・トポロジについて」
- 第 4.2 項「アクティブ/パッシブ・トポロジの管理」
- 第 4.3 項「アクティブ/パッシブ・トポロジにおける Oracle HTTP Server および OC4J の高可用性機能の概要」

## 4.1 アクティブ/パッシブ・トポロジについて

アクティブ/パッシブ・トポロジは、次のもので構成されています。

- ハードウェア・クラスタに配置されている2つのノード
- 仮想ホスト名とIPアドレス
- 2つのノード間で共有される共有記憶域

Oracle ホームは共有記憶域にインストールします。アクティブ・パッシブ・トポロジでは、実行時に1つのノードのみがアクティブになります。もう一方のノードはパッシブになります。アクティブ・ノードは共有記憶域をマウントするため、ファイルにアクセスし、すべてのプロセスを実行し、すべてのリクエストを処理できます。クライアントは、仮想ホスト名を使用してアクティブ・ノードにアクセスします。クライアントがトポロジ内のノードの物理ホスト名を認識している必要はありません。

なんらかの理由でアクティブ・ノードに障害が発生すると、フェイルオーバー・イベントが実行され、パッシブ・ノードが引き継いでアクティブ・ノードになります。パッシブ・ノードは共有記憶域をマウントし、すべてのプロセスを実行し、すべてのリクエストを処理します。仮想ホスト名およびIPは、パッシブ・ノードを指すようになります。クライアントは、仮想ホスト名を使用してノードにアクセスするため、リクエストを処理しているのがパッシブ・ノードであることを意識することはありません。

フェイルオーバーを有効にするには、ノードがハードウェア・クラスタに存在する必要があります。

---

---

**注意：** OracleAS Cold Failover Cluster トポロジにある各ノードのローカル記憶域への Oracle ホームのインストールはサポートされていません。Oracle ホームは共有記憶域にインストールする必要があります。

---

---

### ベンダー・クラスタウェア

アクティブ/パッシブ・トポロジの2つのノードはハードウェア・クラスタにあります。このクラスタには通常、ベンダー・クラスタウェアが含まれています。認証済クラスタウェアの一覧は、Oracle Technology Network の Web サイト (<http://www.oracle.com/technology>) を参照してください。

Windows で実行している場合は、クラスタに次の製品が必要です。

- Oracle Fail Safe
- Microsoft Cluster Server

これらの製品は、トポロジ内の両方のノード（アクティブとパッシブ）にインストールする必要があります。

### アクティブ/パッシブ・トポロジ：利点

- アクティブ/アクティブ・トポロジに必要なロード・バランサが不要なため、実装が簡単です。
- ロード・バランシング・アルゴリズム、クラスタリング、レプリケーションなどのオプションを構成する必要がないため、アクティブ/アクティブ・トポロジよりも構成が簡単です。
- 1 インスタンスのトポロジのシミュレーションがアクティブ/アクティブ・トポロジよりも容易です。

**アクティブ/パッシブ・トポロジ: 欠点**

- アクティブ/アクティブ・トポロジほどのスケーラビリティがありません。ノードをトポロジに追加して容量を増加することができません。
- (HTTPセッション状態とEJBステートフル・セッションBeanからの)状態情報は、レプリケートされないため、データベースなどの永続ストレージに保存しないかぎり、ノードが予期せずに終了したときに失われます。

**アクティブ/パッシブ・トポロジの図**

図 4-1 は、共有記憶域に Oracle Application Server の Oracle ホームがインストールされているアクティブ/パッシブ・トポロジの図を示しています。Oracle ホームには Oracle HTTP Server と OC4J の両方が含まれています。図 4-2 は、Oracle HTTP Server と OC4J が個別の Oracle ホームにインストールされている分散アクティブ/パッシブ・トポロジを示しています。

図 4-1 同じ Oracle ホームに Oracle HTTP Server と OC4J がインストールされているアクティブ/パッシブ・トポロジ

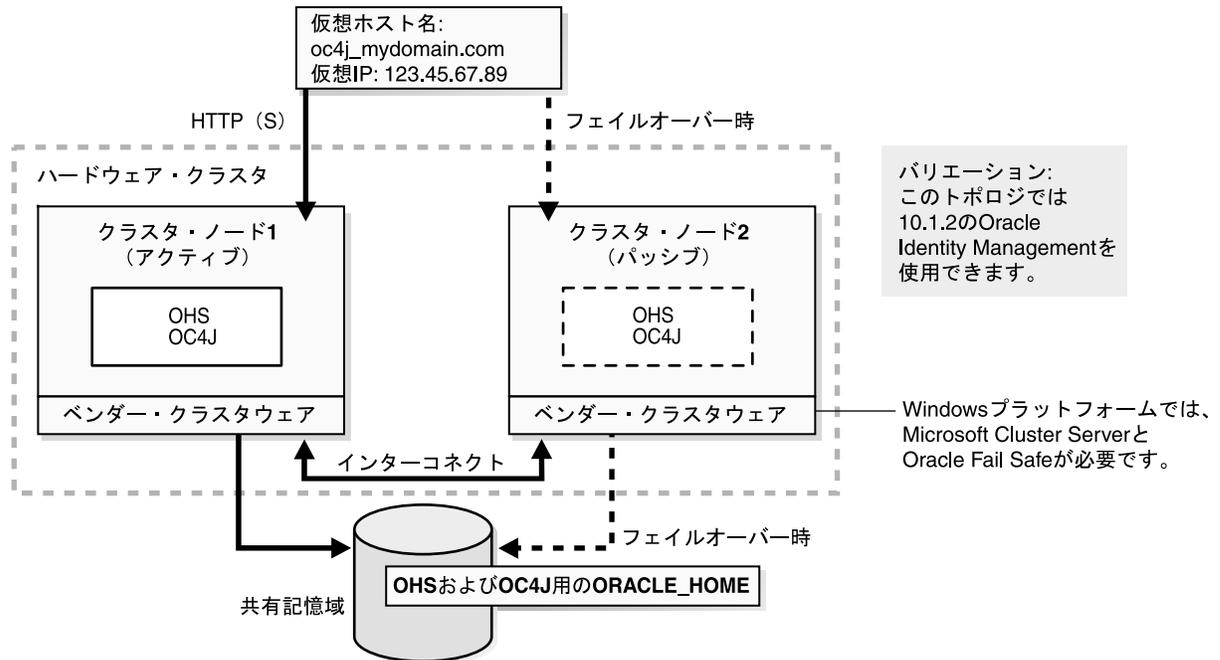
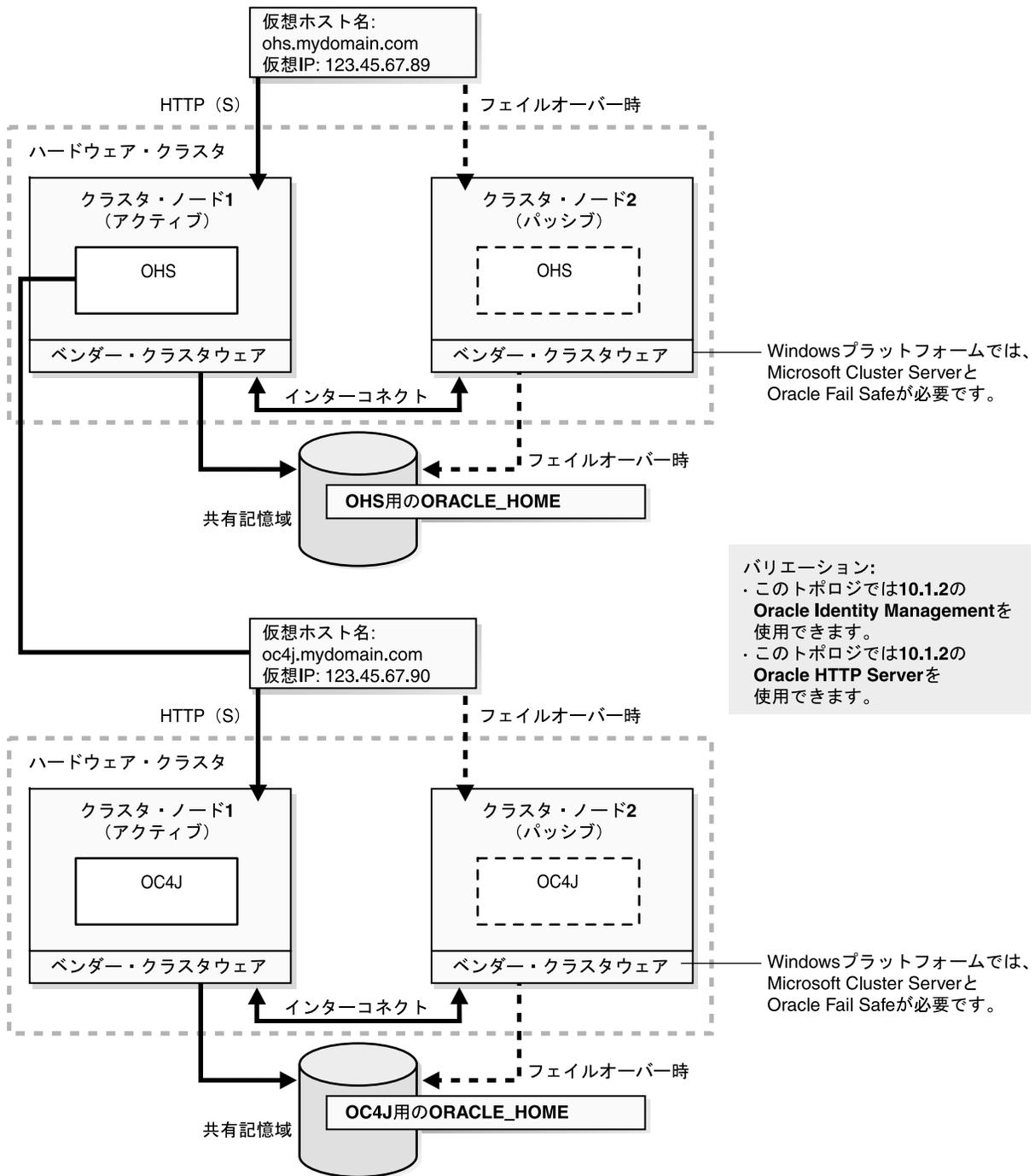


図 4-2 個別の Oracle ホームに Oracle HTTP Server と OC4J がインストールされているアクティブ/パッシブ・トポロジ



## 4.2 アクティブ/パッシブ・トポロジの管理

アクティブ/パッシブ・トポロジの管理では、1つのみの Oracle ホームを管理すればよいいため、単一インスタンスの Oracle Application Server トポロジの管理と非常によく似ています。

この項の項目は次のとおりです。

- 第 4.2.1 項「Application Server Control コンソールを使用した管理」
- 第 4.2.2 項「コンポーネントの起動と停止」
- 第 4.2.3 項「アプリケーションのデプロイ」

### 4.2.1 Application Server Control コンソールを使用した管理

Application Server Control コンソールを使用して、アクティブ/パッシブ・トポロジを管理できます。Application Server Control コンソールにアクセスするには、物理ホスト名のかわりに仮想ホスト名を使用します。

### 4.2.2 コンポーネントの起動と停止

opmnctl コマンドを使用すると、トポロジ内のコンポーネントを起動および停止できます。次に、例を示します。

```
opmnctl startall
```

Application Server Control コンソールを使用して起動と停止の操作を実行することもできます。

### 4.2.3 アプリケーションのデプロイ

アプリケーションは通常の方法でデプロイします。アプリケーションは、アクティブ・ノードのみにデプロイします。

アプリケーションのデプロイ方法の詳細は、『Oracle Containers for J2EE デプロイメント・ガイド』を参照してください。

## 4.3 アクティブ/パッシブ・トポロジにおける Oracle HTTP Server および OC4J の高可用性機能の概要

表 4-1 は、アクティブ/パッシブ・トポロジにおける Oracle HTTP Server と OC4J の高可用性機能の概要を示しています。

**表 4-1 アクティブ/パッシブ・トポロジにおける Oracle HTTP Server および OC4J の高可用性機能の概要**

項目	説明
ノード障害からの保護	<p><b>Oracle HTTP Server:</b> アクティブ・ノードに障害が発生した場合は、パッシブ・ノードが引き継いで Oracle HTTP Server プロセスを実行します。</p> <p><b>OC4J:</b> アクティブ・ノードに障害が発生した場合は、パッシブ・ノードが引き継いで OC4J プロセスを実行します。</p>
サービス障害からの保護	<p><b>Oracle HTTP Server および OC4J:</b> Windows では、Oracle Fail Safe がアクティブ・ノードのサービスを監視します。サービスが停止した場合は、Oracle Fail Safe がそのサービスの再開を試みます。サービスを開始できない場合は、Oracle Fail Safe がインスタンスをノード 2 (パッシブ・ノード) にフェイルオーバーし、サービスを開始します。</p> <p>UNIX では、ベンダー・クラスタウェアが Oracle Fail Safe と同様のサービスを提供します。</p>

**表 4-1 アクティブ/パッシブ・トポロジにおける Oracle HTTP Server および OC4J の高可用性機能の概要（続き）**

項目	説明
プロセス障害からの保護	<p data-bbox="451 260 1365 344"><b>Oracle HTTP Server:</b> OPMN は Oracle HTTP Server プロセスを監視し、障害発生時にそのプロセスを再起動します。さらに、トポロジ内の別の Oracle HTTP Server プロセスで障害が発生した場合、OPMN はそれぞれの Oracle HTTP Server に障害を通知します。</p> <p data-bbox="451 359 1365 443"><b>OC4J:</b> OPMN は OC4J プロセスを監視し、障害発生時にそのプロセスを再起動します。また OPMN は、この再起動が成功しなかった場合、動作している OC4J プロセスにのみリクエストを送信するよう <code>mod_oc4j</code> に通知します。</p> <p data-bbox="451 457 1235 483">OPMN の詳細は、<a href="#">第 2.1 項「OPMN でのプロセス管理」</a>を参照してください。</p>

---

---

## Oracle SOA Suite の高可用性

この章では、Oracle SOA Suite の高可用性ソリューションについて説明します。Oracle SOA Suite は、サービス指向アーキテクチャ・システムのデプロイおよび管理に使用する一連のコンポーネントです。

この章では、次の Oracle SOA Suite コンポーネントの高可用性について説明します。

- [第 5.1 項 「インストールに関する注意事項」](#)
- [第 5.2 項 「Oracle BPEL Process Manager」](#)
- [第 5.3 項 「Oracle Enterprise Service Bus」](#)
- [第 5.4 項 「Oracle Business Activity Monitoring」](#)
- [第 5.5 項 「Oracle Service Registry」](#)
- [第 5.6 項 「Oracle Business Rules」](#)
- [第 5.7 項 「Oracle Web Services Manager」](#)

## 5.1 インストールに関する注意事項

高可用性トポロジに Oracle SOA Suite コンポーネントをインストールするには、次の2つの手順を実行する必要があります。

- メインの Oracle Application Server インストール CD-ROM を使用して、Oracle HTTP Server および OC4J コンポーネントをインストールします。

Oracle SOA Suite は、メインの Oracle Application Server CD-ROM からインストールしないでください。

**拡張**インストール・オプションを必ず選択してください。Oracle SOA Suite コンポーネントが自動インストールされる**基本**インストール・オプションは使用しないでください。このオプションでは、インストールするコンポーネントを選択できません。**基本**インストール・オプションは、非高可用性トポロジを構築する場合にのみ使用します。

- コンポーネント CD-ROM から、必要な Oracle SOA Suite コンポーネントをインストールします。コンポーネントは、OC4J の上部にインストールします。

必要な Oracle SOA Suite コンポーネントのインストールには、それぞれのコンポーネント CD-ROM（たとえば、Oracle BPEL Process Manager CD-ROM や Oracle Enterprise Service Bus CD-ROM）を使用します。これによって、目的のコンポーネントのみをインストールできます。Oracle Enterprise Service Bus の高可用性トポロジでは、ESB リポジトリ・サーバーと ESB ランタイム・サーバーを異なる Oracle ホームにインストールする必要があるため、これは Oracle Enterprise Service Bus の場合に特に重要です。

## 5.2 Oracle BPEL Process Manager

この項では、Oracle BPEL Process Manager の高可用性について説明します。この項の項目は次のとおりです。

- 第 5.2.1 項「Oracle BPEL Process Manager について」
- 第 5.2.2 項「アクティブ / アクティブ・トポロジの Oracle BPEL Process Manager」
- 第 5.2.3 項「アクティブ / パッシブ・トポロジの Oracle BPEL Process Manager」
- 第 5.2.4 項「アダプタとの Oracle BPEL Process Manager の使用」

### 5.2.1 Oracle BPEL Process Manager について

BPEL（ビジネス・プロセス実行言語）は XML ベースの言語であり、これを使用すると別個のサービスを1つのエンドツーエンド・プロセス・フローにアセンブルできます。Oracle BPEL Process Manager には、BPEL ビジネス・プロセスの設計、配置および管理を行うためのフレームワークが用意されています。

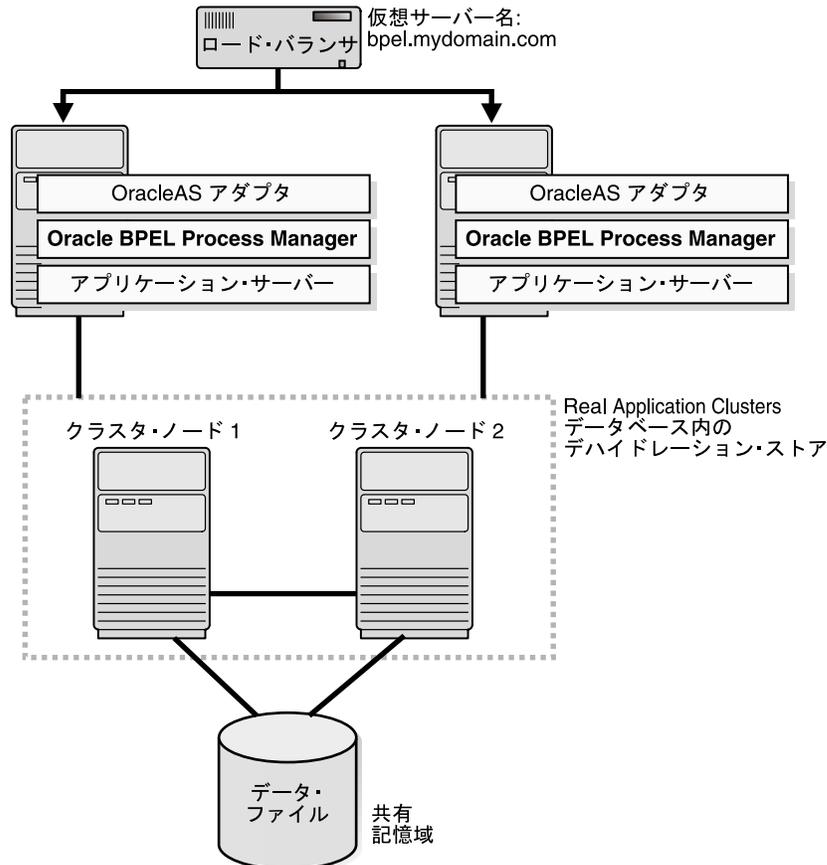
Oracle BPEL Process Manager は J2EE アプリケーションの1つであり、様々なアプリケーション・サーバーで実行できます。これはステートレス・アプリケーションですが、データベースを「デハイドレーション・ストア」として使用して、プロセスの状態に関する情報を格納します。

Oracle BPEL Process Manager の詳細は、『Oracle BPEL Process Manager 管理者ガイド』および『Oracle BPEL Process Manager 開発者ガイド』を参照してください。

## 5.2.2 アクティブ/アクティブ・トポロジの Oracle BPEL Process Manager

Oracle BPEL Process Manager のアーキテクチャもステートレスなため、高可用性の実現が簡単になります。図 5-1 に、Real Application Clusters データベースをデハイドレーション・ストアとして使用した、アクティブ/アクティブ・トポロジでの Oracle BPEL Process Manager を示します。

図 5-1 アクティブ/アクティブ・トポロジの Oracle BPEL Process Manager



アクティブ/アクティブ・トポロジには、次のような特性があります。

- すべてのコンポーネントがアクティブです (つまり、すべてのコンポーネントがリクエストを処理できます)。
- Oracle Application Server インスタンスの前にロード・バランサが配置されています。これには、Oracle Web Cache などのソフトウェア・ロード・バランサを使用できますが、本番での使用に備えて、F5 BIG-IP などのハードウェア・ロード・バランサをお勧めします。ロード・バランサによって、アクティブ・ノードの 1 つにリクエストが分散されます。該当するノードが使用不可の場合、リクエストは次に使用可能なノードに送信されます。
- すべての BPEL サーバーが、デハイドレーション・ストアとして同じデータベースを使用します。
- すべての BPEL エンジンで、生成される SOAP URL のサーバー URL およびコールバック・アドレスとしてロード・バランサを使用します。具体的には、Oracle BPEL Process Manager コンソールを使用して、soapServerUrl および soapCallbackUrl プロパティをロード・バランサの URL に設定します。

- OPMNによって返される JNDI プロバイダのリストを使用するように JNDI ルックアップ (サービスの取得のためなど) を変更します。たとえば、次のように JNDI プロバイダを指定するものではありません。

```
jndiProviderURL = "ormi://localhost/Service"
```

かわりに次のように指定します。

```
jndiProviderURL = "opmn:ormi://host1:port1:oc4j/Service,  
opmn:ormi://host2:port2:oc4j/Service"
```

これによって、JVM レベルの高可用性を実現できます。1 つの OC4J インスタンスに対して複数の JVM プロセスを実行している場合、OPMN では、使用可能な JVM の数に基づいて独立した JNDI オブジェクトにリクエストをルーティングできます。

---

**注意：** アクティブ / アクティブ・トポロジの Oracle BPEL Process Manager のインストールおよび構成方法の詳細な手順は、『Oracle BPEL Process Manager 管理者ガイド』の「Oracle BPEL Process Manager のクラスタリング」を参照してください。

---

## BPEL プロセスのデプロイ

アクティブ / アクティブ・トポロジに BPEL プロセスをデプロイする手順は次のとおりです。

- 設計環境 (たとえば、Oracle JDeveloper) で、クラスタ内の各ノードに BPEL プロセスをコンパイルしてデプロイします。これは手動またはスクリプトを使用して実行できます。詳細は、『Oracle BPEL Process Manager 開発者ガイド』を参照してください。
- クラスタ内のすべてのサーバーが同じドメインを持つようにします (前述の手順と同様、手動またはスクリプトを使用して実行できます)。

### 5.2.2.1 アクティブ / アクティブ・トポロジでの BPEL プロセスの起動

この項では、SOAP/WSDL または Oracle BPEL Process Manager Java API を使用して BPEL プロセスを起動する場合に必要な変更について説明します。

SOAP/WSDL を使用する場合は、ノードのホスト名ではなく、ロード・バランサの仮想サーバー名を使用する必要があります。

Oracle BPEL Process Manager Java API を使用する場合は、アクティブ / アクティブ・トポロジで各ノードのホスト名をリストします。前述の `jndiProviderURL` の例を参照してください。

### 5.2.2.2 デハイドレーション・ストアに Real Application Clusters データベースを使用

高可用性を実現するには、Real Application Clusters データベースなどの高可用性データベースでデハイドレーション・ストアを実行する必要があります。Real Application Clusters データベースを使用すると、すべてのコンポーネントで高可用性を実現できます。

デハイドレーション・ストアに Real Application Clusters データベースを使用する場合、構成に対し、次の変更を行う必要があります。

- Real Application Clusters データベースへの接続情報のフォーマットが次のようになるように、OC4J の `data-sources.xml` ファイルを変更します。

```
jdbc:oracle:thin:@(DESCRIPTION=  
  (ADDRESS_LIST=(LOAD_BALANCE=on)  
    (ADDRESS=(PROTOCOL=tcp) (HOST=hostname1) (PORT=1521))  
    (ADDRESS=(PROTOCOL=tcp) (HOST=hostname2) (PORT=1521))  
  )  
  (CONNECT_DATA=(SERVICE_NAME=orc1))  
)
```

*hostname1* および *hostname2* は、Real Application Clusters データベースを実行するノードの名前を指定します。

*orcl* は、データベースのサービス名を指定します。

アドレスおよびロード・バランサのオプションは両方とも ADDRESS\_LIST 要素内にあります。

また、Real Application Clusters データベースに対する高速接続フェイルオーバーを有効化できます。構成手順は、『Oracle Application Server エンタープライズ・デプロイメント・ガイド』の第 3.2 項「APPHOST1 および APPHOST2 での RAC データベースに対する高速接続フェイルオーバーの構成」を参照してください。

### 5.2.2.3 異なるサブネットにあるマシンでのアクティブ/アクティブ・トポロジの実行

マシンが異なるサブネットにある環境でアクティブ / アクティブ・トポロジを実行する場合は、トポロジを実行する前に、次の手順に従って ORACLE\_HOME\bpel\system\config\jgroups-protocol.xml ファイルに変更を加える必要があります。

1. Oracle BPEL Process Manager を停止します。
2. jgroups-protocol.xml ファイルの最初の <config> セクションをコメント・アウトします。例 5-1 の (2) を参照してください。
3. jgroups-protocol.xml ファイルの 2 番目の <config> セクションをコメント解除します。例 5-1 の (3) を参照してください。
4. *hostA* および *hostB* を、アクティブ / アクティブ・トポロジの実際のホスト名に置き換えます。例 5-1 の (4) を参照してください。
5. Oracle BPEL Process Manager を再起動します。

#### 例 5-1 jgroups-protocol.xml ファイルの例

```
<!-- ***** JGroups Protocol Stack Configuration ***** -->
<!-- generated by XmlConfigurator on Mon Apr 26 11:15:41 PDT 2004 -->
<!-- input file: default.old.xml -->

<!-- (2) Comment out this <config> section. -->
<config>
  <UDP mcast_send_buf_size="32000"
    mcast_port="45788"
    ucast_rcv_buf_size="64000"
    mcast_addr="228.8.15.24"
    bind_to_all_interfaces="true"
    loopback="true"
    mcast_rcv_buf_size="64000"
    max_bundle_size="48000"
    max_bundle_timeout="30"
    use_incoming_packet_handler="false"
    use_outgoing_packet_handler="false"
    ucast_send_buf_size="32000"
    ip_ttl="32"
    enable_bundling="false"/>
  <PING timeout="2000"
    num_initial_members="3"/>
  <MERGE2 max_interval="10000"
    min_interval="5000"/>
  <FD timeout="2000"
    max_tries="3"
    shun="true"/>
  <VERIFY_SUSPECT timeout="1500"/>
  <pbcast.NAKACK max_xmit_size="8192"
```

```

        use_mcast_xmit="false"
        gc_lag="50"
        retransmit_timeout="600,1200,2400,4800"/>
<UNICAST timeout="1200,2400,3600"/>

<!--
- desired_avg_gossip: periodically sends STABLE messages around. 0 disables
  this
- max_bytes: max number of bytes received from anyone until a STABLE message
  is sent. Use either this or
  desired_avg_gossip, but not both ! 0 disables it.
- stability_delay: range (number of milliseconds) that we wait until sending a
  STABILITY message.
  This prevents STABILITY multicast storms. If max_bytes is used, this should
  be set to a low value (> 0 though !).
-->
<pbcast.STABLE stability_delay="1000"
  desired_avg_gossip="20000"
  max_bytes="0"/>
<FRAG frag_size="8192"
  down_thread="false"
  up_thread="false"/>
<VIEW_SYNC avg_send_interval="60000" down_thread="false" up_thread="false" />
<pbcast.GMS print_local_addr="true"
  join_timeout="3000"
  join_retry_timeout="2000"
  shun="true"/>
</config>
<!-- (2) end of <config> section to comment out -->

<!-- For cluster across subnet, please use the following tcp config and
- change the initial_hosts instead of the above, the initial_hosts that are
going to be participating in the cluster.
-->
<!-- (3) Uncomment this <config> section
<config>
  <TCP start_port="7900" loopback="true" send_buf_size="32000"
    recv_buf_size="64000"/>
<!-- (4) Replace "hostA" and "hostB" with hostnames in your topology.
-->
  <TCPPING timeout="3000" initial_hosts="hostA[7900],hostB[7900]" port_range="3"
    num_initial_members="3"/>
  <FD timeout="2000" max_tries="4"/>
  <VERIFY_SUSPECT timeout="1500" down_thread="false" up_thread="false"/>
  <pbcast.NAKACK gc_lag="100" retransmit_timeout="600,1200,2400,4800"/>
  <pbcast.STABLE stability_delay="1000" desired_avg_gossip="20000"
    down_thread="false" max_bytes="0" up_thread="false"/>
  <VIEW_SYNC avg_send_interval="60000" down_thread="false" up_thread="false" />
  <pbcast.GMS print_local_addr="true" join_timeout="5000"
    join_retry_timeout="2000" shun="true"/>
</config>
--> <!-- (3) end of config section to comment out -->

```

## 5.2.3 アクティブ/パッシブ・トポロジの Oracle BPEL Process Manager

Oracle BPEL Process Manager は、OracleAS Cold Failover Cluster、つまりアクティブ / パッシブなトポロジでも動作できます。アクティブ / パッシブ・トポロジは、ハードウェア・クラスタ内の 2 つのノード、共有記憶域、および仮想ホスト名と仮想 IP アドレスで構成されます。共有記憶域にファイルをインストールして、ハードウェア・クラスタ内のどちらかのノードからこれらのファイルにアクセスできるようにします。クライアントは、仮想ホスト名を使用して、ハードウェア・クラスタ内のアクティブ・ノードにアクセスします。そのアクティブ・ノードが使用不可になった場合は、フェイルオーバー・イベントが発生し、パッシブ・ノードがかわりにプロセスを実行します。

## 5.2.4 アダプタとの Oracle BPEL Process Manager の使用

Oracle BPEL Process Manager を Oracle Application Server アダプタとともに使用して、Oracle BPEL Process Manager プロセスを外部リソースと統合します。これらのアダプタは、J2CA (J2EE Connector Architecture) に基づいています。

この項では、可用性の高い方法で Oracle BPEL Process Manager をアダプタとともに実行する方法について説明します。この項の項目は次のとおりです。

- [第 5.2.4.1 項「J2CA ベースのアダプタの概要」](#)
- [第 5.2.4.2 項「同時実行性のサポート」](#)
- [第 5.2.4.3 項「アダプタのアクティブ / アクティブ・トポロジ」](#)
- [第 5.2.4.4 項「アダプタの変更済アクティブ / アクティブ・トポロジ」](#)
- [第 5.2.4.5 項「アダプタのアクティブ / パッシブ・トポロジ」](#)

### 5.2.4.1 J2CA ベースのアダプタの概要

表 5-1 に示すように、Oracle Application Server の J2CA ベースのアダプタは、Oracle Application Server を様々な外部リソースと統合します。

表 5-1 アダプタのタイプ

アダプタ・タイプ	例
テクノロジー	テクノロジー・タイプのアダプタは、Oracle Application Server をトランスポート・プロトコル、データ・ストアおよびメッセージ・ミドルウェアと統合します。  テクノロジー・タイプのアダプタの例は次のとおりです。FTP、ファイル、データベース、JMS および Advanced Queuing。
パッケージ化されたアプリケーション	パッケージ化されたアプリケーションのタイプのアダプタは、Oracle Application Server を Siebel や SAP などのアプリケーションと統合します。
レガシーおよびメインフレーム	レガシーおよびメインフレームのタイプのアダプタは、Oracle Application Server を CICS や VSAM などのアプリケーションと統合します。

アダプタの詳細は、『Oracle Application Server Adapter 概要』を参照してください。

### 5.2.4.2 同時実行性のサポート

同時実行性のサポートとは、複数のアダプタ・サービスがデータ破損を起こすことなく同時に同じリソースにアクセスできることです。同時実行性のサポートを、トランザクションのサポートと考えることができます。例として、データベース・アダプタの複数のアダプタ・サービスからデータベース内の同じ表に同時にアクセスできることが挙げられます。

アダプタは同時実行性をサポートするものとサポートしないものに分けられます。

- 同時実行性をサポートしないアダプタは、ファイルやFTPのアダプタです。これは、ファイル・システムである外部リソースが同時アクセスをサポートしないためです。
- その他すべてのアダプタが、同時実行性をサポートします。

表 5-2 に示すように、同時実行性のサポートまたは非サポートによって、アダプタの高可用性オプションが影響を受けます。

表 5-2 アダプタの高可用性オプション

アダプタ・タイプ	高可用性オプション
同時実行性のサポート	<ul style="list-style-type: none"> <li>■ 第 5.2.4.3 項「アダプタのアクティブ / アクティブ・トポロジ」</li> <li>■ 第 5.2.4.4 項「アダプタの変更済アクティブ / アクティブ・トポロジ」</li> <li>■ 第 5.2.4.5 項「アダプタのアクティブ / パッシブ・トポロジ」</li> </ul>
同時実行性の非サポート (ファイルおよびFTPアダプタ)	<ul style="list-style-type: none"> <li>■ 第 5.2.4.4 項「アダプタの変更済アクティブ / アクティブ・トポロジ」</li> <li>■ 第 5.2.4.5 項「アダプタのアクティブ / パッシブ・トポロジ」</li> </ul>

すべての高可用性オプションについて、すべてのノードにアダプタがインストールされていることが前提となっています。ただし、高可用性オプションによっては、1つのノードでのみ Oracle BPEL Process Manager を実行するものがあります。

### 5.2.4.3 アダプタのアクティブ / アクティブ・トポロジ

このトポロジは、同時実行性をサポートするアダプタに使用できます。

図 5-1 に、アクティブ / アクティブ・トポロジを示します。このトポロジでは、1つ以上のノードの前面にロード・バランサを配置します。各ノードで、Oracle BPEL Process Manager およびビジネス・プロセスを配置して実行します。これは、すべてのノードですべてのコンポーネントを使用できるため、高可用性の点で理想のモデルといえます。

同時実行性をサポートしないアダプタをアクティブ / アクティブ・トポロジに配置すると、外部データソースのデータが破損するおそれがあります（同時に同じファイルの読取りと書込みを行うなど）。

#### 5.2.4.4 アダプタの変更済アクティブ/アクティブ・トポロジ

この変更済バージョンのアクティブ/アクティブ・トポロジは、次の相違点を除いて完全なアクティブ/アクティブ・トポロジと同様です。

- このトポロジでもすべてのノード上で Oracle BPEL Process Manager およびビジネス・プロセスを配置して実行しますが、最初のノード以外のすべてのノードで、同時実行性をサポートしないアダプタを使用するパートナー・リンクに対してアクティブ化エージェントを無効にします。

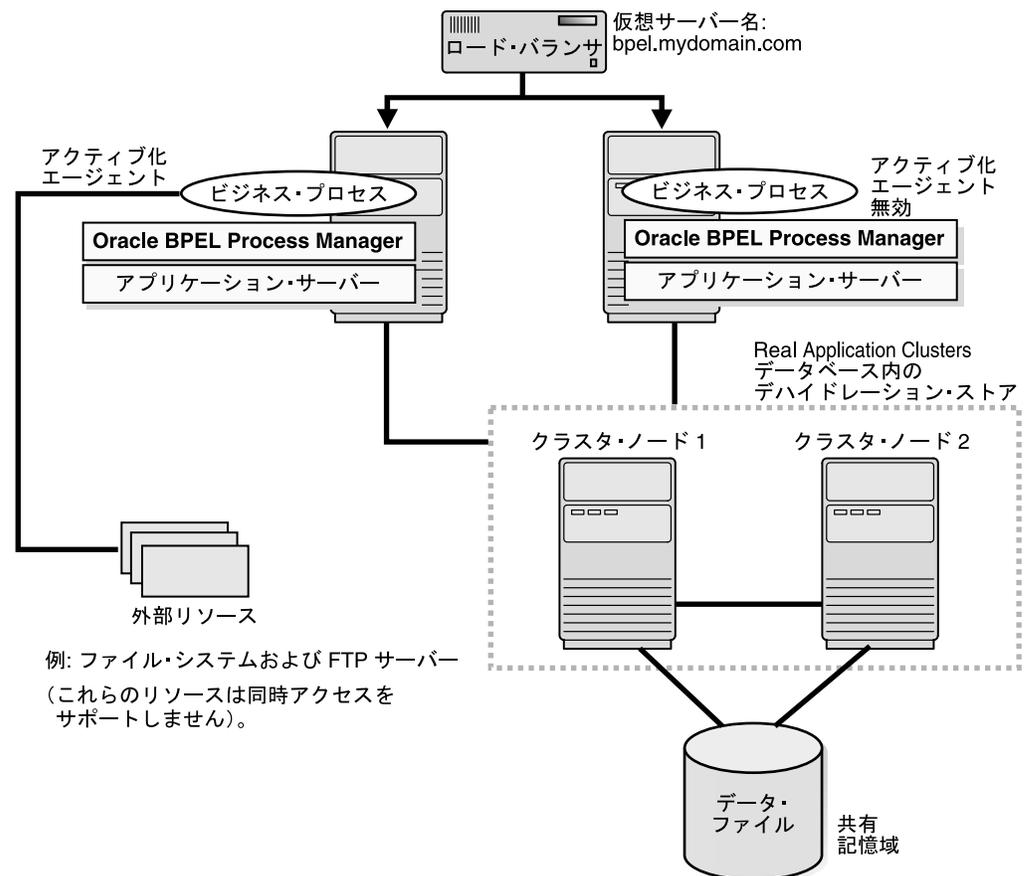
最初のノードのアダプタ・サービスのみが受信リクエストを取得します。

このトポロジは、次のアダプタに使用できます。

- 同時実行性をサポートしないアダプタ。
- 同時実行性をサポートするアダプタ。このタイプのアダプタは完全なアクティブ/アクティブ・トポロジで実行できますが、リソースを調整するために（メッセージの適切なシーケンスの管理と確認など）そのトポロジでは実行しないことにすると、実行されるアダプタ・サービスを常に1つのみにすることがこれを実行する唯一の方法になります。

図 5-2 に、変更済アクティブ/アクティブ・トポロジを示します。

図 5-2 変更済アクティブ/アクティブ・トポロジ



アクティブ化エージェントを使用しているノードに障害が発生した場合は、次の手順を実行する必要があります。

- 障害の発生したノードのアクティブ化エージェントを無効にし、このノードが再びアクティブになったときにアクティブ化エージェントが実行されないようにします（別のノードですでにアクティブ化エージェントが実行されているためです）。
- 別のノードのアクティブ化エージェントを有効にします。

#### アクティブ化エージェントを無効にするには

アクティブ化エージェントを無効にするには、`bpel.xml` ファイルの `activationAgent` 要素をコメントアウトします。次の例では、無効にするアクティブ化エージェントが含まれるコメント行を示します。

```
<activationAgents>
  <!-- start comment
  <activationAgent
    className="oracle.tip.adapter.fw.agent.jca.JCAActivationAgent"
    partnerLink="InboundPL">
    <property name="InputFileDir">C:/ora_home/integration/bpm/samples/tutorials/
      121.FileAdapter/ComplexStructure/InputDir/</property>
    <property name="portType">Read_ptt</property>
  </activationAgent>
  end comment -->
</activationAgents>
```

#### 5.2.4.5 アダプタのアクティブ/パッシブ・トポロジ

このトポロジは、すべてのアダプタに使用できます。アクティブ/パッシブ・トポロジは、OracleAS Cold Failover Cluster トポロジとも呼ばれます。

アクティブ/パッシブ・トポロジ (図 5-3 を参照) では、ハードウェア・クラスタに 2 つのノードがあります。そのうちの 1 つはアクティブ・ノード、もう 1 つはパッシブ・ノードです。共有記憶域もあり、これに Oracle ホーム・ディレクトリをインストールします。共有記憶域はアクティブ・ノードにのみマウントされます。

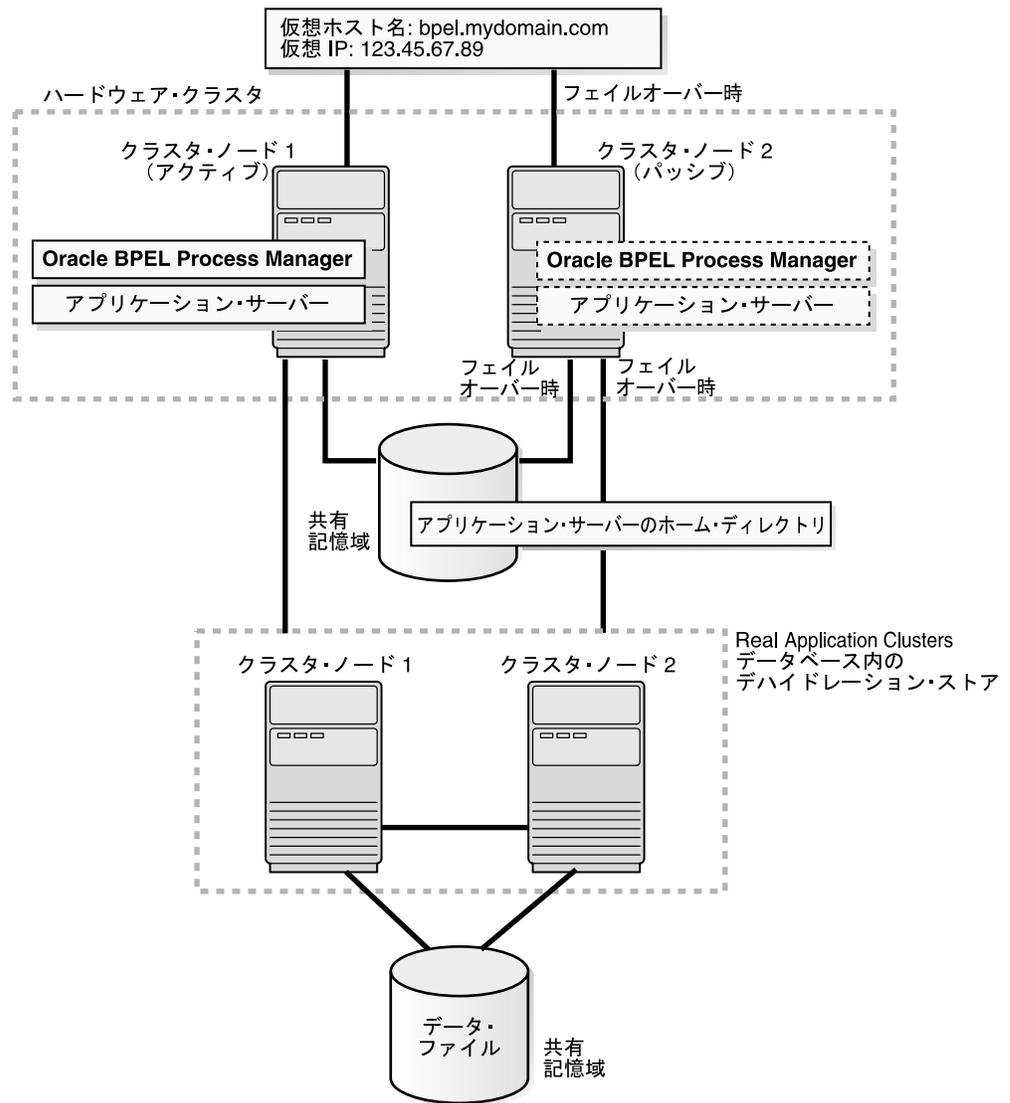
ハードウェア・クラスタのアクティブ・ノードは、仮想ホスト名および IP に関連付けられています。クライアントは、仮想ホスト名を使用して、クラスタ内のアクティブ・ノードにアクセスします。

実行時に、アクティブ・ノードはプロセスを実行します。仮想ホスト名はアクティブ・ノードを指し示します。アクティブ・ノードが使用不可になると、フェイルオーバー・イベントが発生します。パッシブ・ノードが新しいアクティブ・ノードになり、プロセスを実行します。

次の相違点を除いて単一ノード配置の場合と同じように、Oracle BPEL Process Manager をインストールして管理します。

- Oracle ホーム・ディレクトリを共有記憶域にインストールします。
- クライアントは、仮想ホスト名を使用してアクティブ・ノードにアクセスします。クライアントは、Oracle BPEL Process Manager を実際に実行しているノードを認識する必要はありません。

図 5-3 OracleAS Cold Failover Cluster トポロジにおけるアダプタとの Oracle BPEL Process Manager の使用



## 5.3 Oracle Enterprise Service Bus

この項では、Oracle Enterprise Service Bus の高可用性について説明します。この項の項目は次のとおりです。

- 第 5.3.1 項「Oracle Enterprise Service Bus について」
- 第 5.3.2 項「アクティブ / アクティブ・トポロジの Oracle Enterprise Service Bus」
- 第 5.3.3 項「Oracle Enterprise Service Bus での Oracle Application Server アダプタの使用」

### 5.3.1 Oracle Enterprise Service Bus について

Oracle Enterprise Service Bus は、SOA とイベントドリブン・アーキテクチャ (EDA) を使用するサービスのための基盤です。Oracle Enterprise Service Bus を使用することで、エンタープライズの内側および外側の両方にある複数のエンドポイント間でデータを移動できます。

Oracle Enterprise Service Bus では、異なるアプリケーション間でのビジネス・ドキュメント (XML メッセージなど) の接続、変換およびルーティングに、オープン・スタンダードが使用されます。これにより、既存のアプリケーションへの影響を最小限に抑えながら、ビジネス・データの監視と管理が可能になります。

Oracle Enterprise Service Bus は、高い可用性を必要とする次の 2 つの主要なコンポーネントで構成されます。

- ESB ランタイム・サーバー: 登録されている ESB サービスを実行します。
- ESB リポジトリ・サーバー: ORAESB スキーマを持つデータベースと通信します。

Oracle Enterprise Service Bus には、管理タスクに使用できる Web ベースのコンソールも用意されています。

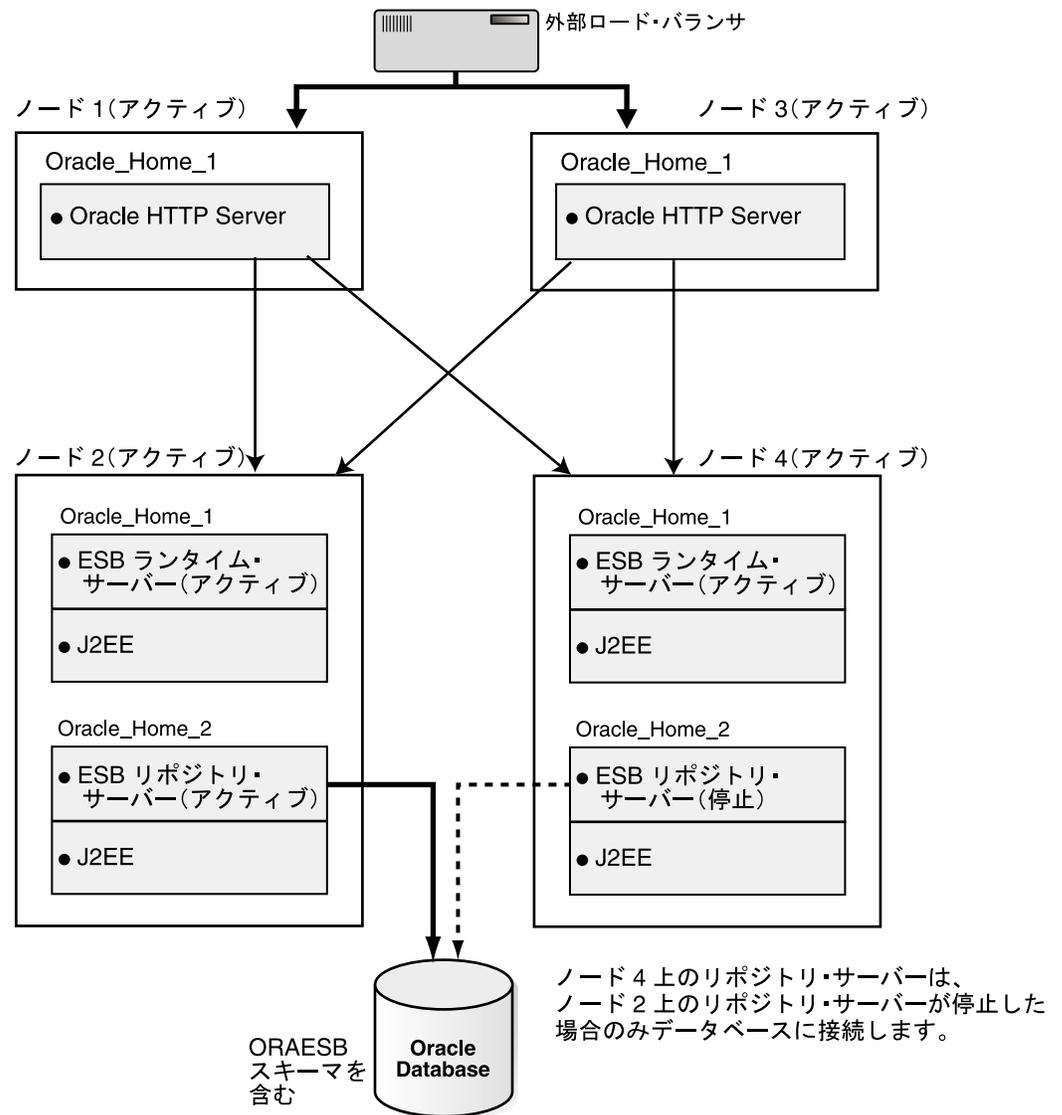
Oracle Enterprise Service Bus の詳細は、次のドキュメントを参照してください。

- Oracle Enterprise Service Bus のインストレーション・ガイド
- 『Oracle Enterprise Service Bus クイック・スタート・ガイド』
- 『Oracle Enterprise Service Bus 開発者ガイド』

### 5.3.2 アクティブ / アクティブ・トポロジの Oracle Enterprise Service Bus

Oracle Enterprise Service Bus を可用性の高い環境で実行するには、[図 5-4](#) に示すようなアクティブ / アクティブ・トポロジで実行する必要があります。通常のアクティブ / アクティブ・トポロジと同様、トポロジ内のすべてのノードがアクティブとなるため、各アクティブ・ノードへのリクエストの分散にロード・バランサが必要です。

図 5-4 アクティブ/アクティブ・トポロジの Oracle Enterprise Service Bus



この図は、4つのノードからなるアクティブ/アクティブ・トポロジを示しています。状況によっては、さらに多数のノードを配置できますが、次の注意事項を確実に理解して構成する必要があります。

---

**注意 1:** ESB のアクティブ / アクティブ・トポロジでは、常に 1 つの ESB リポジトリ・サーバーのみが実行されるようにする必要があります。他の ESB リポジトリ・サーバーは実行しないでください。

作業内容

- すべてのノードのすべての Oracle ホームを同じクラスタ内に構成します (同じマルチキャスト・アドレスを使用)。これによって、ノード 1 および 3 の Oracle HTTP Server が、ノード 2 または 4 のどちらかの OC4J にリクエストを送信可能になります。
- リポジトリ・サーバーを実行する各 OC4J インスタンスに、OPMN のサービス・フェイルオーバー機能の使用を構成します。OPMN のこの機能によって、トポロジ全体で 1 つのリポジトリ・サーバー・インスタンスのみの実行が可能になります (この例ではノード 2)。ノード 2 の OC4J に障害が発生した場合は、OPMN によって、ノード 4 のリポジトリ・サーバーが起動されます。  
  
また、OPMN によって、リポジトリ・サーバーを実行中のノードが Oracle HTTP Server に通知されるため、Oracle HTTP Server はリポジトリ・サーバーへのリクエストを適切なインスタンスに送信できます。
- リクエストがノード 1 または 3 の Oracle HTTP Server に送信されるようにロード・バランサを構成します (ロード・バランシング方式には、ラウンドロビンなどの任意の方式を使用)。

これらの手順の実行方法の詳細は、『Oracle Application Server エンタープライズ・デプロイメント・ガイド』の第 3 章「mySOACompany の Web 層とアプリケーション層のインストールと構成」を参照してください。

ESB ランタイム・サーバー・インスタンスはすべてがアクティブで、Oracle HTTP Server はこれらの任意のランタイム・サーバーにリクエストを送信できます。ESB ランタイム・サーバーへのリクエストには、event コンテキスト・ルート (つまり、`http://host:port/event/...` のような形式のリクエスト) が使用されます。

---

**注意 2:** ESB リポジトリ・サーバーと ESB ランタイム・サーバーは、異なる Oracle ホームにインストールする必要があります。同じ Oracle ホームにはインストールできません。

このことは、高可用性トポロジを構築するには、ESB を ESB のインストーラからインストールする必要があることを意味します。Oracle Application Server のインストーラでは、ESB リポジトリ・サーバーと ESB ランタイム・サーバーが同じ Oracle ホームにインストールされるため、ESB のインストーラには Oracle Application Server のインストーラを使用できません。

ESB リポジトリ・サーバーと ESB ランタイム・サーバーを異なる Oracle ホームにインストールする必要がある理由は、ESB リポジトリ・サーバーは常に 1 つのインスタンスしか実行できないのに対して、ESB ランタイム・サーバーは同時に複数のインスタンスを実行できることにあります。ESB リポジトリ・サーバーに障害が発生した場合は、別の ESB リポジトリ・サーバーが起動されサービスを引き継ぐように構成する必要があります。そのためには、異なる Oracle ホームにインストールしたほうが管理が容易になります。

---

**注意 3:** アクティブ / アクティブ・トポロジでは、同時実行性をサポートしないアダプタ (ファイル・アダプタと FTP アダプタ) を実行できません。これは、外部リソース (これらのアダプタの場合はファイル・システム) が同時実行アクセスをサポートしないためです。

詳細は、第 5.2.4.2 項「同時実行性のサポート」を参照してください。

---

## インストールおよび構成手順

このトポロジの構築手順の詳細は、『Oracle Application Server エンタープライズ・デプロイメント・ガイド』の第3章「mySOACompany の Web 層とアプリケーション層のインストールと構成」を参照してください。この章には、インストールの手順とインストール後の手順（構成手順）が説明されています。

---

**注意：** Oracle Enterprise Service Bus システムに仮想ホスト名およびポート番号を構成したら、すべてのノードで ESB ランタイム・サーバーを再起動し、更新された構成を有効化する必要があります。

---

### 5.3.2.1 ESB リポジトリ・サーバーが ESB ランタイム・サーバーと異なる ESB クラスタにあることの検証

（『Oracle Application Server エンタープライズ・デプロイメント・ガイド』の第3章「mySOACompany の Web 層とアプリケーション層のインストールと構成」に説明されている）インストール後の手順の1つに、ESB ランタイム・サーバー用の ESB クラスタの名前変更があります。名前変更の理由は、この ESB クラスタに ESB ランタイム・サーバー・インスタンスのみが含まれることを確実にすることです。この ESB クラスタには、ESB リポジトリ・サーバー・インスタンスを含めることはできません。

これを検証するには、テキスト・エディタで `ORACLE_HOME¥integration¥esb¥config¥esb_config.ini` ファイルを表示します。ESB ランタイム・サーバーおよび ESB リポジトリ・サーバーを実行するすべての Oracle ホームに対して、この手順を実行します。

すべての ESB ランタイム・サーバー・インスタンスが同じ ESB クラスタ名を使用すること、および ESB リポジトリ・サーバー・インスタンスが異なる ESB クラスタ名を使用することを確認します。クラスタ名は、ファイル内の `cluster_name` パラメータで指定されます。次に、例を示します。

```
# Cluster name
cluster_name=ESBcluster1
```

### 5.3.2.2 仮想ホスト名およびポートの ORAESB スキーマへの登録の検証

（『Oracle Application Server エンタープライズ・デプロイメント・ガイド』の第3章「mySOACompany の Web 層とアプリケーション層のインストールと構成」に説明されている）インストール後の手順の1つに、仮想ホスト名および仮想ポート（ロード・バランサに構成されるホスト名とポート）の ORAESB スキーマへの登録があります。これらの値は、次のコマンドを実行して検証できます。

```
ant export-params
-DDB_URL=jdbc:oracle:thin:@//dbhost:1521/ORCL
-DDB_USER=oraesb -DDB_PASSWORD=oraesbpassword
```

`dbhost`: ORAESB スキーマを持つデータベースを実行するマシンを指定します。

`1521`: データベース・リスナーがリスニングするポートを指定します。

`ORCL`: データベースのサービス名を指定します。

`oraesbpassword`: ORAESB スキーマのパスワードを指定します。

値が正しくない場合は、手順を再実行して、正しい値を登録します。

### 5.3.2.3 Oracle Enterprise Service Bus での Real Application Clusters データベースの使用

高可用性環境を構築するには、Real Application Clusters データベースに ORAESB スキーマをインストールする必要があります。Oracle Enterprise Service Bus のインストール時の指示に従って、Real Application Clusters データベース内のすべてのノードを入力する必要があります。

### 5.3.2.4 OC4J インスタンスの OC4J クラスタへのクラスタ化

OC4J インスタンスは OC4J クラスタとしてクラスタ化できますが、必須ではありません。

### 5.3.2.5 Oracle Enterprise Service Bus サービスへのアクセス

クライアントは、ロード・バランサに構成された仮想ホスト名およびポートを使用して、Oracle Enterprise Service Bus サービスにアクセスします。

Oracle Enterprise Service Bus コンソールにアクセスするには、物理ホスト名を使用します。

### 5.3.2.6 アプリケーションの登録

ESB アプリケーションを Oracle Enterprise Service Bus に登録するときは、アクティブ / アクティブ・トポロジ内のすべての ESB ランタイム・サーバーに登録する必要があります。

## 5.3.3 Oracle Enterprise Service Bus での Oracle Application Server アダプタの使用

J2CA ベースの Oracle Application Server アダプタは、Oracle Enterprise Service Bus にも使用できます。これらのアダプタの高可用性トポロジは、[第 5.2.4 項「アダプタとの Oracle BPEL Process Manager の使用」](#)を参照してください。

## 5.4 Oracle Business Activity Monitoring

この項では、高可用性環境での Oracle Business Activity Monitoring の実行について説明します。この項の項目は次のとおりです。

- [第 5.4.1 項「Oracle Business Activity Monitoring について」](#)
- [第 5.4.2 項「要件」](#)
- [第 5.4.3 項「インストールに関する重要項目」](#)
- [第 5.4.4 項「Microsoft Cluster Server \(MSCS\) の構成」](#)
- [第 5.4.5 項「Microsoft IIS 6 の Web ガーデンの設定」](#)
- [第 5.4.6 項「Enterprise Link と Plan Monitor の構成」](#)
- [第 5.4.7 項「既知の問題とトラブルシューティング」](#)
- [第 5.4.8 項「メッセージ整合の設定」](#)

### 5.4.1 Oracle Business Activity Monitoring について

Oracle Business Activity Monitoring は、様々なソースから情報をリアルタイムに収集し、フィルタ、変換処理を行って、操作メトリックやキー・パフォーマンス・インジケータに対する影響を分析します。Oracle Business Activity Monitoring は、このデータを対話型のダッシュボードにリアルタイムに表示できると同時に、アラートを送信できます。

高可用性環境で Oracle Business Activity Monitoring を実行するには、そのすべてのコンポーネントを高可用性対応にする必要があります。

- **Active Data Cache** は、トランザクション・フィード、データ・ウェアハウス、各種エンタープライズ・データ・ソースなど、多様なビジネス・ソースから収集したデータを処理します。Enterprise Link は、Active Data Cache をデータ・ソースに接続します。

Active Data Cache の高可用性を実現するには、ハードウェア・クラスタ内のノードにインストールします。また、Real Application Clusters データベースにデータを格納するよう Active Data Cache を構成します。

**Active Data Cache のフェイルオーバー:** ハードウェア・クラスタ内のノードに障害が発生した場合は、ハードウェア・クラスタ内のもう一方のノードが処理を引き継ぎます。

Active Data Cache とアクティブ・ノードの状態は、クラスタ・サービスによって監視されます。アクティブ・ノードまたは Active Data Cache サービスに障害が発生した場合は、クラスタ・サービスによって、クラスタ内のもう一方のノードの Active Data Cache が起動されます。

クラスタ・サービスは、Active Data Cache の実行状態だけでなく、デッドロックの発生も監視します。

- **Enterprise Link** は、Active Data Cache をビジネス・データ・ソースに接続します。また Enterprise Link は、Active Data Cache にデータを送信する前に、データの追加処理と不要なデータのフィルタ処理を実行できます。

Enterprise Link の高可用性を実現するには、2 つまたはそれ以上のノードにインストールします。インストールするのは、アクティブ / アクティブ構成のノードです。

**Enterprise Link のフェイルオーバー:** Enterprise Link を実行するノードに障害が発生しても、残りのアクティブ・ノードが実行を継続しています。

- **Plan Monitor** は、データ・ソースの特定、データの操作および Active Data Cache へのデータのロードが指示されているプランを監視します。プランには変換と呼ばれる手順が含まれ、それらをリンクすることで、Enterprise Link Design Studio を介した強力なデータ・フローを構築できます。プランとそのステータスは、Active Data Cache に格納されます。

Plan Monitor の高可用性を実現するには、2 つまたはそれ以上のノードにインストールします。インストールするのは、アクティブ / アクティブ構成のノードです。

**Plan Monitor のフェイルオーバー:** Plan Monitor を実行するノードに障害が発生した場合は、残りのアクティブ・ノードが、障害発生ノードによって監視されていたプランを監視できます。

- **Web アプリケーション (Report Server を含む)** は、Report Cache からのデータにレポート定義を適用し、データを表示可能な形式に変換します。Web アプリケーションは、Microsoft IIS と連携動作します。

Web アプリケーションの高可用性を実現するには、2 つまたはそれ以上のノードにインストールし、その前面にロード・バランサを配置します。ロード・バランサは、任意のノードにリクエストを転送できます。

**Web アプリケーションのフェイルオーバー:** Web アプリケーションを実行するノードに障害が発生した場合は、ロード・バランサによって、そのノードへのリクエストの転送が停止されます。ロード・バランサは、残りのアクティブ・ノードにリクエストを転送します。

- **Report Cache** は、Active Data Cache からのデータをキャッシュします。

Report Cache の高可用性を実現するには、ハードウェア・クラスタ内のノードにインストールします。また、Real Application Clusters データベースにデータを格納するよう Report Cache を構成します。

**Report Cache のフェイルオーバー:** ハードウェア・クラスタ内のノードに障害が発生した場合は、ハードウェア・クラスタ内のもう一方のノードが処理を引き継ぎます。Report Cache とアクティブ・ノードの状態は、クラスタ・サービスによって監視されます。アクティブ・ノードまたは Report Cache サービスに障害が発生した場合は、クラスタ・サービスによって、クラスタ内のもう一方のノードの Report Cache が起動されます。

フェイルオーバーが実行されている間 (Microsoft Cluster Server によって、スタンバイ・ノードの Report Cache が起動されている間)、Oracle BAM のリアルタイム・ダッシュボードおよびレポートには、リアルタイムなデータを表示できないことに注意してください。フェイルオーバーにかかる時間はシステムに依存します。

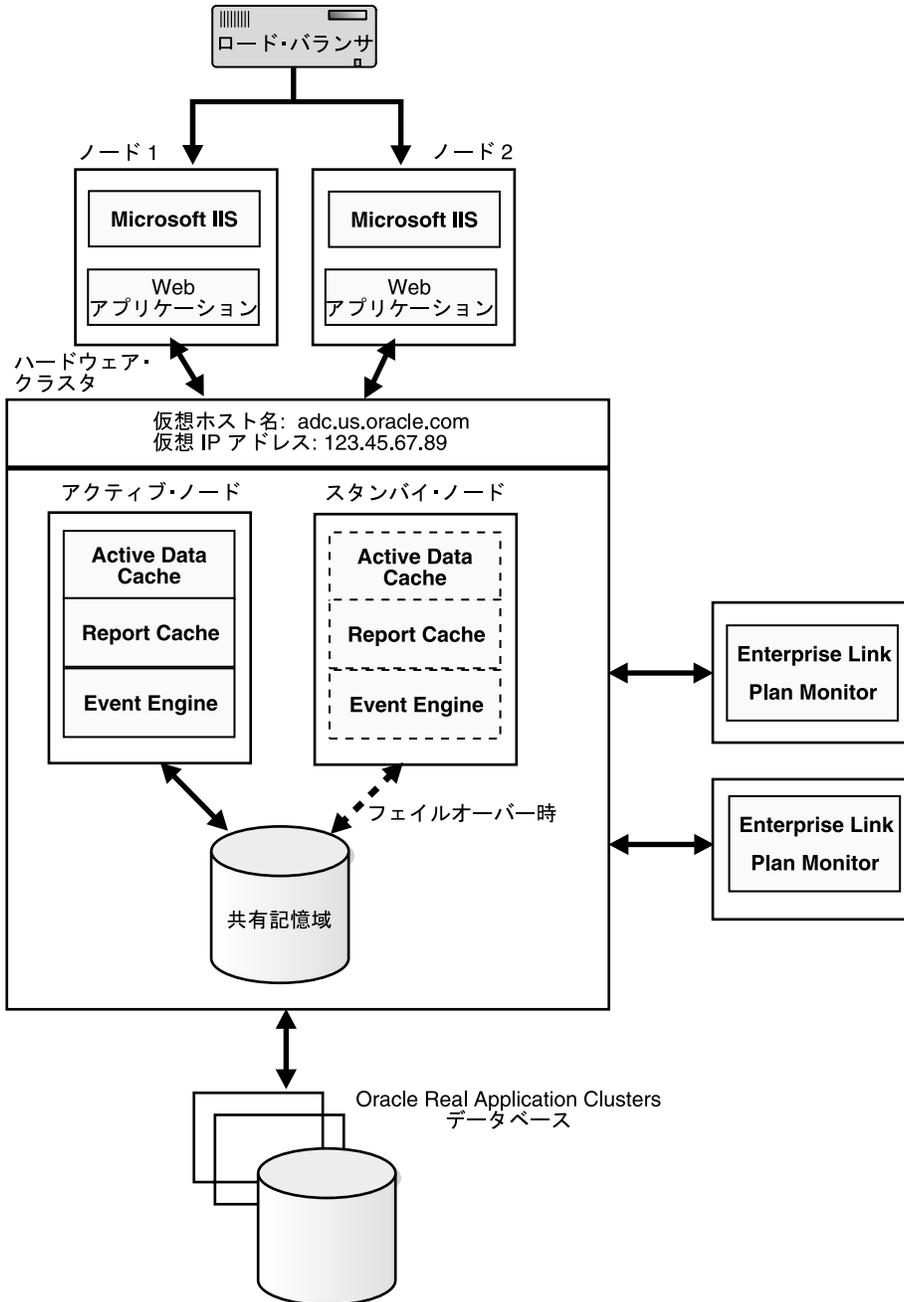
- **Event Engine** は、Active Data Cache のデータを、ルールに定義されている条件に対して比較監視します。条件が満たされると、条件に関連付けられたアクションが実行されます。たとえば、アクションには、アラート・メッセージの送信などを指定できます。

Event Engine の高可用性を実現するには、ハードウェア・クラスタ内のノードにインストールします。また、Real Application Clusters データベースにデータを格納するよう Event Engine を構成します。

**Event Engine のフェイルオーバー**：ハードウェア・クラスタ内のノードに障害が発生した場合は、ハードウェア・クラスタ内のもう一方のノードが処理を引き継ぎます。Event Engine とアクティブ・ノードの状態は、クラスタ・サービスによって監視されます。アクティブ・ノードまたは Event Engine サービスに障害が発生した場合は、クラスタ・サービスによって、クラスタ内のもう一方のノードの Event Engine が起動されます。

図 5-5 に、Oracle Business Activity Monitoring の高可用性トポロジを示します。

図 5-5 Oracle Business Activity Monitoring の高可用性トポロジ



## 5.4.2 要件

高可用性環境で Oracle Business Activity Monitoring を実行するには、次のアイテムが必要です。

- Web アプリケーション・ノードのフロントエンドとなるロード・バランサ。ロード・バランサには、HTTP トラフィック用の仮想ホスト名およびポートが構成されます。クライアントは、この仮想ホスト名およびポートを使用して Web アプリケーションにアクセスします。
- Active Data Cache のバックエンドとなる Oracle Real Application Clusters データベース。
- Active Data Cache がインストールされた 2 つのノードからなるハードウェア・クラスタ。  
ハードウェア・クラスタには、Microsoft Cluster Server (MSCS) の実行と、仮想ホスト名および IP アドレスの構成が必要です。クラスタ内のノードの一方はアクティブで、他方はパッシブです。1 つのノード (アクティブ・ノード) のみが、常にリクエストを処理します。アクティブ・ノードに障害が発生した場合は、フェイルオーバー・イベントが生じ、パッシブ・ノードがアクティブ・ノードになります。

クラスタ内のノードは、ストレージ・デバイスを共有します。アクティブ・ノードのみが、共有記憶域にアクセスできます。

## 5.4.3 インストールに関する重要項目

この項では、[図 5-5](#) のトポロジを構築する目的で Oracle Business Activity Monitoring コンポーネントをインストールする際の重要なポイントについて説明します。各コンポーネントは、任意の順序でインストールできます。

Oracle Business Activity Monitoring のインストーラおよび要件の詳細は、『Oracle Business Activity Monitoring インストレーション・ガイド』を参照してください。

この項の項目は次のとおりです。

- [第 5.4.3.1 項「Active Data Cache、Event Engine および Report Cache のインストール」](#)
- [第 5.4.3.2 項「Web アプリケーションのインストール」](#)
- [第 5.4.3.3 項「Enterprise Link および Plan Monitor のインストール」](#)

### 5.4.3.1 Active Data Cache、Event Engine および Report Cache のインストール

Active Data Cache、Event Engine および Report Cache は、ハードウェア・クラスタ内のノードにインストールします。これらのコンポーネントをインストールするときは、次の点に注意してください。

- 「Installation and Log Directory」画面：ハードウェア・クラスタ内の各ノードのローカル記憶域上のディレクトリを指定します。このディレクトリは、すべてのノード間で同じにする必要があります (たとえば、インストール・ディレクトリは C:\OracleBAM、ログ・ディレクトリは C:\OracleBAM\Logs など)。
- 「Select Components to Install and Configure」画面：コンポーネントを選択し、「Active Data Cache」を展開して「ADC Clustering Support」を選択します。
- 「Configure Location of Oracle BAM Components」画面：次の値を入力します。
  - 「Active Data Cache Server」には、ハードウェア・クラスタに関連付けられている仮想ホスト名を入力します。
  - 「Report Cache Server」には、ハードウェア・クラスタに関連付けられている仮想ホスト名を入力します。
  - 「Event Engine Server」には、ハードウェア・クラスタに関連付けられている仮想ホスト名を入力します。
  - 「Web Applications」には、ロード・バランサに構成されている仮想ホスト名を入力します。

- 「Database Connect Information」画面：Real Application Clusters データベース内のすべてのノードの名前を入力します。次に例を示します。

```
nodeA.oracle.com:1521^nodeB.oracle.com:1521^nodeC.oracle.com:1521
```

ノード名は ^ 文字で区切ります。

### 5.4.3.2 Web アプリケーションのインストール

Web アプリケーションは、ロード・バランサの背後にあるノードにインストールします。Web アプリケーションをインストールするときは、次の点に注意してください。

- 「Select Components to Install and Configure」画面：「**Web Applications**」を選択します。
- 「Configure Location of Oracle BAM Components」画面：第 5.4.3.1 項「**Active Data Cache、Event Engine および Report Cache のインストール**」の説明に従います。トポロジ内のすべてのコンポーネントのインストールで、この画面に同じ値を入力する必要があります。

### 5.4.3.3 Enterprise Link および Plan Monitor のインストール

Enterprise Link をインストールするときは、次の点に注意してください。

- 「Select Components to Install and Configure」画面：「**Enterprise Link**」、「**Plan Monitor**」（「Enterprise Link」の下にある）および「**Oracle Client**」を選択します。「**Oracle Client**」を選択しないと、エラーが表示されます。
- 「Configure Location of Oracle BAM Components」画面：第 5.4.3.1 項「**Active Data Cache、Event Engine および Report Cache のインストール**」の説明に従います。トポロジ内のすべてのコンポーネントのインストールで、この画面に同じ値を入力する必要があります。
- 「Database Connect Information」画面：第 5.4.3.1 項「**Active Data Cache、Event Engine および Report Cache のインストール**」の説明に従います。トポロジ内のすべてのコンポーネントのインストールで、この画面に同じ値を入力する必要があります。

## 5.4.4 Microsoft Cluster Server (MSCS) の構成

Oracle Business Activity Monitoring コンポーネントのインストールが完了したら、次の手順を実行して MSCS を構成します。

### 5.4.4.1 Oracle BAM Active Data Cache リソース・タイプの作成

1. ハードウェア・クラスタ内のいずれかのノードで、次のコマンドを実行します（1 行で記述）。

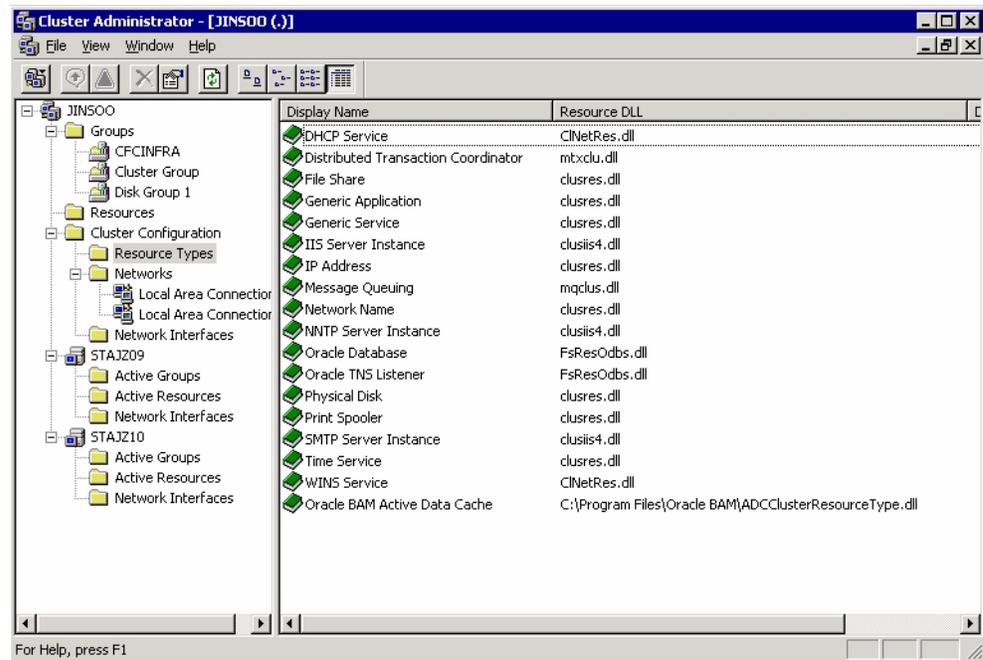
```
cluster.exe restype "Oracle BAM Active Data Cache" /create
/dll:"C:¥OracleBAM¥ADCClusterResourceType.dll"
```

C:¥OracleBAM は、Oracle Business Activity Monitoring をインストールしたディレクトリを表します。

このコマンドは、1 つのノードでのみ実行する必要があります。実行結果は、クラスタ内のすべてのノードに反映されます。

2. 新しいリソース・タイプが作成されたことを「クラスタ アドミニストレータ」で確認します。
  - a. Windows で、「コントロールパネル」の「管理ツール」にある「クラスタ アドミニストレータ」を起動します。
  - b. 「クラスタの構成」を展開して、その下にある「リソースの種類」を選択します。画面の右側に「Oracle BAM Active Data Cache」が表示されます。

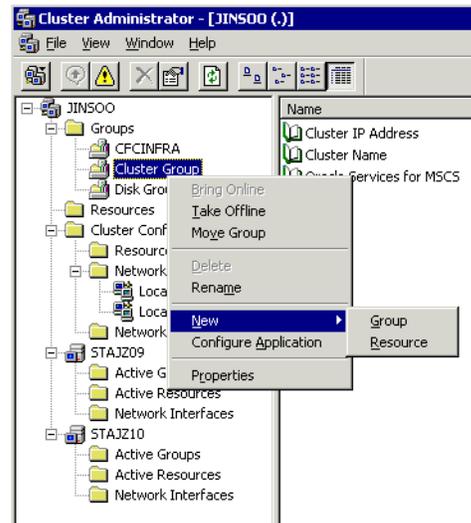
図 5-6 クラスタ アドミニストレータに表示された「Oracle BAM Active Data Cache」



#### 5.4.4.2 Oracle BAM Active Data Cache リソースの作成

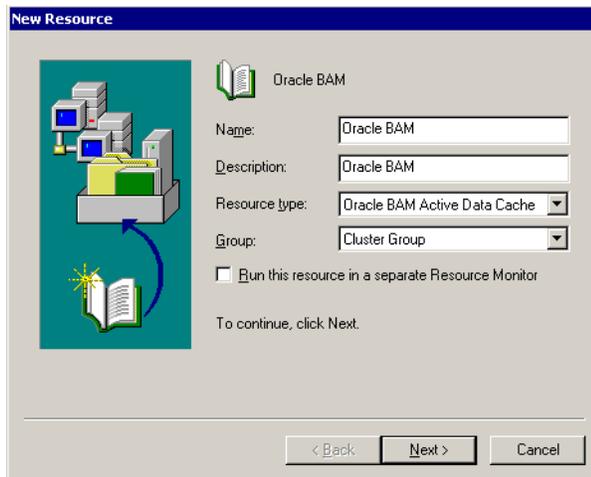
1. クラスタ アドミニストレータでクラスタ・グループ（通常はデフォルトの「クラスタ グループ」）を右クリックして、ポップアップ・メニューから「新規作成」→「リソース」を選択します。

図 5-7 「クラスタ グループ」を右クリックして「新規作成」→「リソース」を選択



2. 「新しいリソース」ダイアログで、次の値を入力します。
  - 名前: OracleBAM などの任意の値を入力します。
  - 説明: 任意の値を入力します。
  - リソースの種類: 「Oracle BAM Active Data Cache」を選択します。
  - グループ: クラスタ・グループを選択します。

図 5-8 「新しいリソース」 ダイアログ



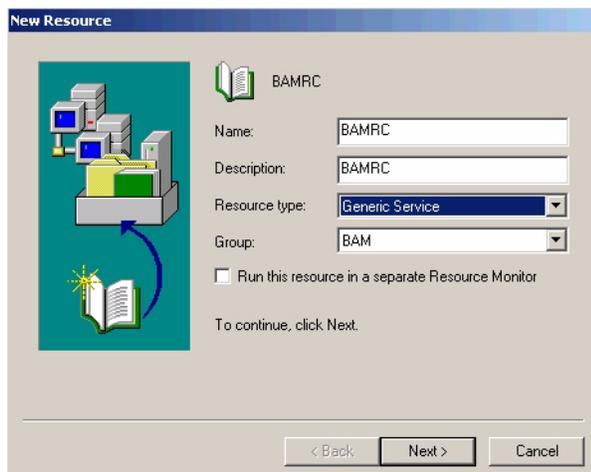
「次へ」をクリックします。

3. 「実行可能な所有者」で、すべてのノードを選択します。「次へ」をクリックします。
4. 「クラスタ名」および「クラスタ IP アドレス」の依存関係を追加します。「完了」をクリックします。
5. 新しいリソースを右クリックして「オンラインにする」を選択し、Oracle BAM Active Data Cache を起動します。

#### 5.4.4.3 Oracle BAM Report Cache リソースの作成

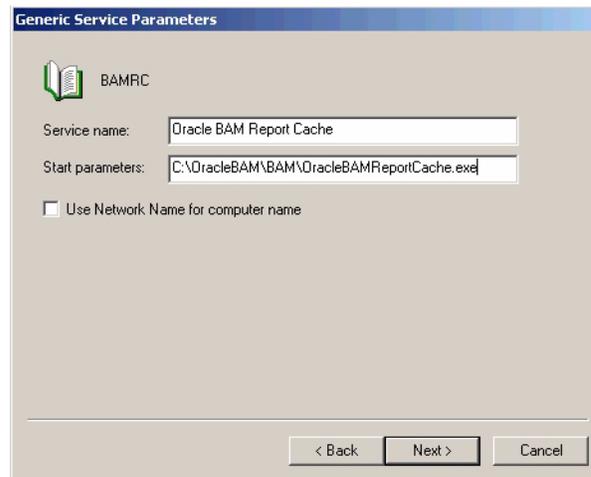
1. クラスタ アドミニストレータでクラスタ・グループ（通常はデフォルトの「クラスタ グループ」）を右クリックして、ポップアップ・メニューから「新規作成」→「リソース」を選択します。
2. 「新しいリソース」ダイアログで、次の値を入力します。
  - 名前: BAMRC などの任意の値を入力します。
  - 説明: 任意の値を入力します。
  - リソースの種類: 「汎用サービス」を選択します。
  - グループ: クラスタ・グループを選択します。

図 5-9 Oracle BAM Report Cache の「新しいリソース」ダイアログ



- 「次へ」をクリックします。
3. 「実行可能な所有者」で、すべてのノードを選択します。「次へ」をクリックします。
  4. 「クラスタ名」および「クラスタ IP アドレス」の依存関係を追加します。「次へ」をクリックします。
  5. 「汎用サービス パラメータ」ダイアログで、次の値を入力します。
    - サービス名：「Oracle BAM Report Cache」と入力します。
    - 起動パラメータ：「BAM\_HOME¥OracleBAMReportCache.exe」（たとえば、C:¥OracleBAM¥BAM¥OracleBAMReportCache.exe）と入力します。
    - ネットワーク名をコンピュータ名として使う：このオプションは選択しません。

図 5-10 Oracle BAM Report Cache の「汎用サービス パラメータ」ダイアログ

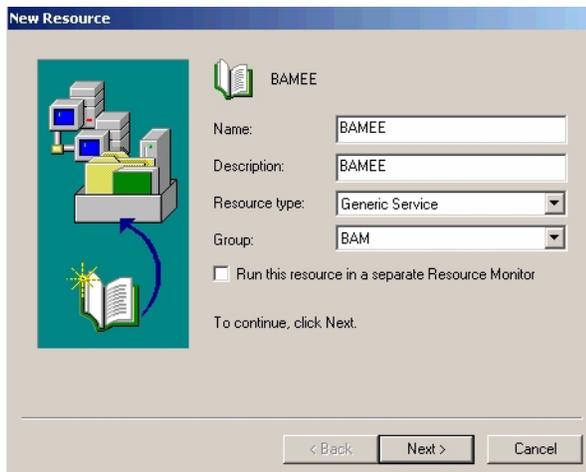


6. 「完了」をクリックします。次に新しいリソースを右クリックして「オンラインにする」を選択し、Oracle BAM Report Cache を起動します。

#### 5.4.4.4 Oracle BAM Event Engine リソースの作成

1. クラスタ アドミニストレータでクラスタ・グループ（通常はデフォルトの「クラスタ グループ」）を右クリックして、ポップアップ・メニューから「新規作成」→「リソース」を選択します。
2. 「新しいリソース」ダイアログで、次の値を入力します。
  - 名前：BAMEE などの任意の値を入力します。
  - 説明：任意の値を入力します。
  - リソースの種類：「汎用サービス」を選択します。
  - グループ：クラスタ・グループを選択します。

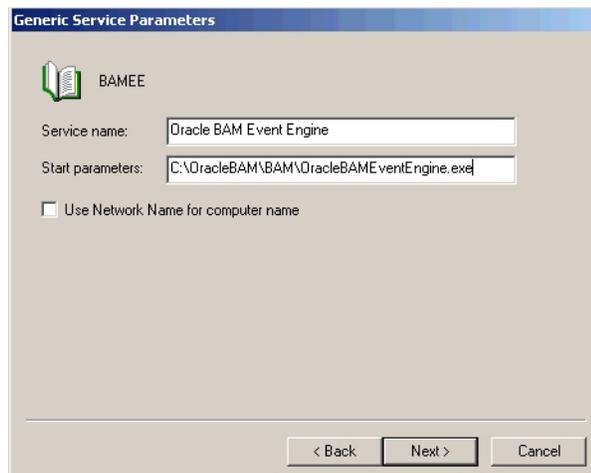
図 5-11 Oracle BAM Event Engine の「新しいリソース」ダイアログ



「次へ」をクリックします。

3. 「実行可能な所有者」で、すべてのノードを選択します。「次へ」をクリックします。
4. 「クラスタ名」および「クラスタ IP アドレス」の依存関係を追加します。「次へ」をクリックします。
5. 「汎用サービス パラメータ」ダイアログで、次の値を入力します。
  - サービス名：「Oracle BAM Event Engine」と入力します。
  - 起動パラメータ：「`BAM_HOME¥OracleBAMEventEngine.exe`」（たとえば、`C:¥OracleBAM¥BAM¥OracleBAMEventEngine.exe`）と入力します。
  - ネットワーク名をコンピュータ名として使う：このオプションは選択しません。

図 5-12 Oracle BAM Event Engine の「汎用サービス パラメータ」ダイアログ



6. 「完了」をクリックします。次に新しいリソースを右クリックして「オンラインにする」を選択し、Oracle BAM Event Engine を起動します。

## 5.4.5 Microsoft IIS 6 の Web ガーデンの設定

Web ガーデンは、複数のワーカー・プロセスで構成されます。Web アプリケーションに Web ガーデンを設定するには、次の手順を実行します。

1. IIS Manager で、ローカル・コンピュータおよび「Application Pools」を展開します。  
「Application Pools」が表示されない場合は、「IIS 5.0 プロセス分離モードで WWW サービスを実行する」オプションが選択解除されていることを確認します。このオプションは、「Web サイト」→「プロパティ」→「サービス」タブにあります。
2. アプリケーション・プールを右クリックして、ポップアップ・メニューから「プロパティ」を選択します。
3. 「パフォーマンス」タブを選択します。
4. 「Web ガーデン」の下の「最大ワーカー プロセス数」フィールドに、アプリケーション・プールに適用するワーカー・プロセス数を入力します。ワーカー・プロセスの数は 2 以上にする必要があります。
5. 「OK」をクリックします。

Web ガーデンの詳細は、次のページを参照してください。

<http://www.microsoft.com/technet/prodtechnol/WindowsServer2003/Library/IIS/659f2e2c-a58b-4770-833b-df96cabe569e.mspx?mfr=true>

## 5.4.6 Enterprise Link と Plan Monitor の構成

アクティブ / アクティブ構成の複数のノードで Enterprise Link と Plan Monitor の複数のインスタンスを実行する場合は、Oracle Business Activity Monitoring Architect を使用して、Enterprise Link と Plan Monitor の各インスタンスに特定のプランを割り当てる必要があります。

構成手順の詳細は、『Oracle Business Activity Monitoring インストレーション・ガイド』の第 3.7 項「複数の Plan Monitor のインストール」を参照してください。

Oracle Business Activity Monitoring Architect の実行方法の詳細は、『Oracle Business Activity Monitoring Architect ユーザーズ・ガイド』を参照してください。

## 5.4.7 既知の問題とトラブルシューティング

高可用性トポロジで Oracle Business Activity Monitoring を実行するときは、次の各項で説明する既知の問題に注意してください。

- 第 5.4.7.1 項「クラスタ・ノードの障害発生時に Enterprise Link でエラーが発生する」
- 第 5.4.7.2 項「Active Data Cache の実行ノードの障害発生時に Active Viewer が別ノードに自動再接続されない」
- 第 5.4.7.3 項「Real Application Clusters データベース内のノードの障害発生時にデータが失われる場合がある」
- 第 5.4.7.4 項「Plan Monitor が Active Data Cache または Data Flow Service (DFS) に再接続されない」
- 第 5.4.7.5 項「Plan Monitor が Enterprise Link に自動再接続されない」
- 第 5.4.7.6 項「ハードウェア・クラスタのスタンバイ・ノードで icommand を実行するとエラーが表示される」
- 第 5.4.7.7 項「フェイルオーバー時にアラートが起動されない」
- 第 5.4.7.8 項「高速接続フェイルオーバー (FCF) の未サポート」

### 5.4.7.1 クラスタ・ノードの障害発生時に Enterprise Link でエラーが発生する

#### 障害

Active Data Cache を実行するハードウェア・クラスタ内のノードに障害が発生し、フェイルオーバー・イベントが正常に完了したとき（つまり、別ノードの Active Data Cache が起動されたとき）、Enterprise Link がキューおよびログのメッセージを処理しません。一方、フェイルオーバー・イベントの完了後も、プランは引き続き実行されます。

また、Enterprise Link に次のエラーが表示される場合があります。

```
You are unable to connect to the Oracle BAM services. Contact your system
administrator if the error persists.
[ErrorSource="ActiveDataCache", ErrorID="ADCServerConnectionError"]
No connection could be made because the target machine actively refused it
[Error Source="mscolib"]
```

#### 解決策

Enterprise Link にエラーが表示される場合は、プランに対してメッセージ整合機能（保証されたメッセージング機能とも呼ばれる）を使用していることを確認します。メッセージ整合機能を有効化すると、エラーは表示されなくなります。この機能を有効化する手順は、[第 5.4.8 項「メッセージ整合の設定」](#)を参照してください。

メッセージ処理を再開するには、次の順序でコンポーネントを再起動する必要があります。

1. Active Data Cache
2. Data Flow Service
3. Plan Monitor。プランが監視されていない場合は、プランを再実行します。

### 5.4.7.2 Active Data Cache の実行ノードの障害発生時に Active Viewer が別ノードに自動再接続されない

#### 障害

Active Data Cache を実行するハードウェア・クラスタ内のノードに障害が発生し、フェイルオーバー・イベントが正常に完了したとき（つまり、別ノードの Active Data Cache が起動されたとき）、Active Viewer でチャートをドリルダウンするなどのアクションを実行すると、「Oracle BAM サービスに接続できません。」というエラーが表示されます。Active Viewer は、ノード障害およびフェイルオーバー・イベントが発生する前のハードウェア・クラスタに接続されています。

#### 解決策

Active Viewer でレポートを再度開いて表示します。

### 5.4.7.3 Real Application Clusters データベース内のノードの障害発生時にデータが失われる場合がある

#### 障害

Real Application Clusters データベース内のノードに障害が発生すると、次のエラーが表示されます。

```
An error has occurred in the ADC storage system. ORA-01089: immediate
shutdown in progress - no operations are permitted
[ErrorSource="ActiveDataCache", ErrorID="ADCStorageException"]
```

このエラーは、Enterprise Link、Plan Monitor、Active Data Cache などの個々のコンポーネントで生じる可能性があります。

また、ノードの障害によって、その時点で実行中のプランも停止されます。

**解決策**

現時点では、この問題の回避策はありません。ノードのフェイルオーバー後に、ログをチェックして、すべてのデータが揃っていることと、失われたデータがないことを確認する必要があります。

**5.4.7.4 Plan Monitor が Active Data Cache または Data Flow Service (DFS) に再接続されない****障害**

Plan Monitor が Active Data Cache または DFS への接続を失った場合（ネットワーク接続が失われたなどの理由で）、イベント・ログにメッセージが書き込まれ、処理が一時停止されます。このメッセージは次のようになります。

```
2006-03-10 10:46:32,808 [2472] ERROR - PlanMonitor Plan Monitoring
processing suspended. Service must be restarted.
[ErrorSource="PlanMonitor", ErrorID="PlanMonitor.BackgroundFatal"]
An error has occurred in the ADC storage system. ORA-03113:
end-of-file on communication channel
[ErrorSource="ActiveDataCache", ErrorID="ADCStorageException"]
```

**解決策**

Plan Monitor の停止と再起動を手動で行う必要があります。

**5.4.7.5 Plan Monitor が Enterprise Link に自動再接続されない****障害**

Plan Monitor が Enterprise Link への接続を失った場合、自動的に再接続されません。

**解決策**

Plan Monitor を再起動してください。

**5.4.7.6 ハードウェア・クラスタのスタンバイ・ノードで icommand を実行するとエラーが表示される****障害**

ハードウェア・クラスタのスタンバイ・ノードで icommand.exe を実行すると、次のようなエラーが表示されます。

```
> ICommand.exe cmd=import file=sla_adherence_stage.xml
Oracle BAM Command Utility
10g Release 3 (10.1.3.1.0) [Build 3 5 5777 0, ADC Version 1003.0]
Copyright (c) 2002, 2006 Oracle.
All rights reserved.
Error while processing command "import".
[ErrorSource="ICommandEngine", ErrorID="ICommandEngine.Error"]
You are unable to connect to the Oracle BAM services. Contact your system
administrator if the error persists.
[ErrorSource="ActiveDataCache", ErrorID="ADCServerConnectionError"]
Requested Service not found
[ErrorSource="System.Runtime.Remoting"]
```

**解決策**

アクティブ・ノードで icommand.exe を実行します。

### 5.4.7.7 フェイルオーバー時にアラートが起動されない

#### 障害

Microsoft Cluster Server がアクティブ・ノードからスタンバイ・ノードへのフェイルオーバーを実行している間に、アラートに指定されたルールまたは条件が満たされた場合、アラートが起動されないことがあります。

#### 解決策

フェイルオーバーの完了後に、起動対象のアラートがすべて起動されていることを検証します。起動されていない場合は、手動で起動できます。

### 5.4.7.8 高速接続フェイルオーバー (FCF) の未サポート

このリリースの Oracle Business Activity Monitoring では、高速接続フェイルオーバーはサポートされません。

## 5.4.8 メッセージ整合の設定

メッセージ整合は、障害の発生時にメッセージの再処理を可能にする機能です。プランにメッセージ整合を設定するには、次の各項の手順を実行します。

- 第 5.4.8.1 項「Oracle BAM Enterprise Message Receiver」ダイアログで「Run forever」を選択する
- 第 5.4.8.2 項「サブプランに各レコードの反復処理を設定する」
- 第 5.4.8.3 項「グローバル・トランザクションに変換を含める」
- 第 5.4.8.4 項「グローバル・トランザクションにメッセージ・トラッカを含める」

#### メッセージ整合について

メッセージ整合は、障害の発生時、具体的には第 5.4.7.1 項「クラスタ・ノードの障害発生時に Enterprise Link でエラーが発生する」に説明されている障害の発生時に、メッセージの再処理を可能にする機能です。メッセージの再処理がどのような場合に行われるかを理解するには、イベントの発生順序を知っておく必要があります。

メッセージはキューから読み取られ、メッセージ・ログ・データ・オブジェクトに記録されます。各メッセージには独自の ID があり、Active Data Cache に対して行われる各トランザクションの Message Tracker 変換によって使用されます。

メッセージがメッセージ・ログ・データ・オブジェクトに記録される前に障害が発生した場合、メッセージはキューから削除されていません。この場合、メッセージはまだ処理されていないため、メッセージの再処理は必要ありません。

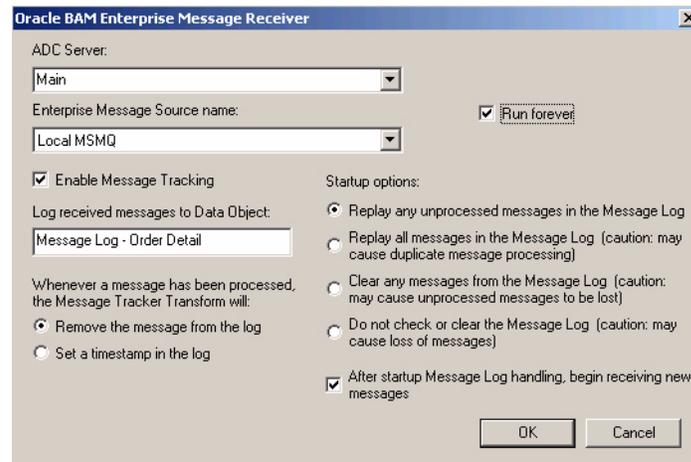
メッセージがメッセージ・ログ・データ・オブジェクトに記録されキューから削除された後に障害が発生した場合、プランによる処理が未完了のメッセージは再処理が必要です。

#### 5.4.8.1 「Oracle BAM Enterprise Message Receiver」ダイアログで「Run forever」を選択する

「Oracle BAM Enterprise Message Receiver」ダイアログ (図 5-13) で、次のオプションを選択します。

- 「Run forever」オプション。このオプションを選択すると、プランが継続的に実行され、ライブ・データが継続的にデータ・オブジェクトに提供されるようになります。プランは監視対象にもする必要があります。
- 「Enable Message Tracking」オプション。
- 起動オプションとして、「Replay any unprocessed messages in the Message Log」または「Replay all messages in the Message Log (caution: may cause duplicate message processing)」のどちらか。

図 5-13 「Run forever」 および 「Enable Message Tracking」 が選択された 「Oracle BAM Enterprise Message Receiver」 ダイアログ

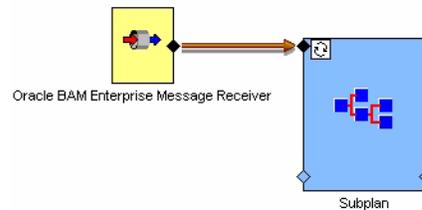


### 5.4.8.2 サブプランに各レコードの反復処理を設定する

サブプランを作成し、それに各レコードの反復処理を設定します。

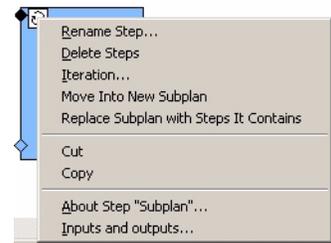
図 5-14 に、メッセージ・レシーバからサブプランのトップ・コネクタへのコネクタを示します。

図 5-14 Oracle BAM Enterprise Message Receiver からサブプランへのコネクタ



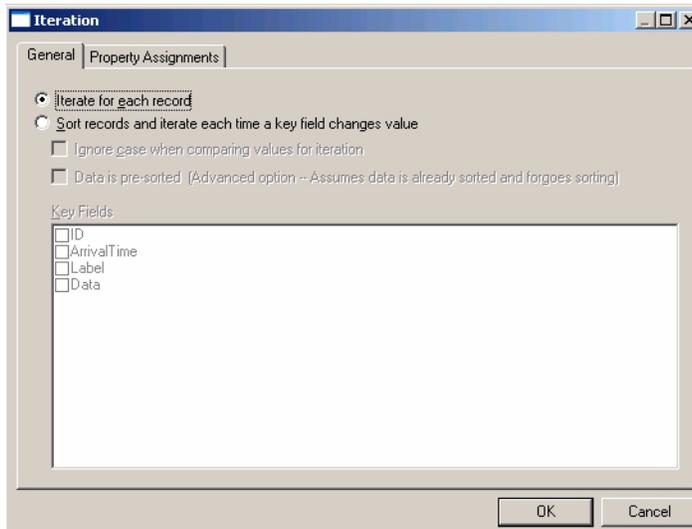
サブプラン上の反復アイコンを右クリックして、ポップアップ・メニュー (図 5-15) から「Iteration」を選択します。

図 5-15 サブプラン上の反復アイコンのポップアップ・メニュー



「Iteration」 ダイアログで、「Iterate for each record」を選択します (図 5-16)。

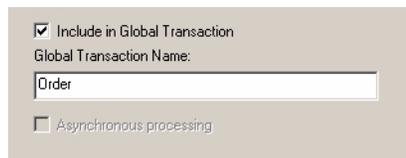
図 5-16 「Iterate for each record」が選択された「Iteration」ダイアログ



### 5.4.8.3 グローバル・トランザクションに変換を含める

反復設定したサブプラン内の各メッセージまたは各行がプランによってトランザクションとして処理されるように、「Include in Global Transaction」オプションを選択します（図 5-17）。

図 5-17 「Include in Global Transaction」を選択して名前を指定する



このオプションは、次の各ダイアログにあります。

- 「Oracle BAM Insert」ダイアログ
- 「Oracle BAM Update」ダイアログ
- 「Oracle BAM Delete」ダイアログ
- 「Oracle BAM Lookup」ダイアログ

グローバル・トランザクションには名前を指定する必要があります。この名前は、サブプラン内のすべてのトランザクションで同じにする必要があります。

#### 5.4.8.4 グローバル・トランザクションにメッセージ・トラッカを含める

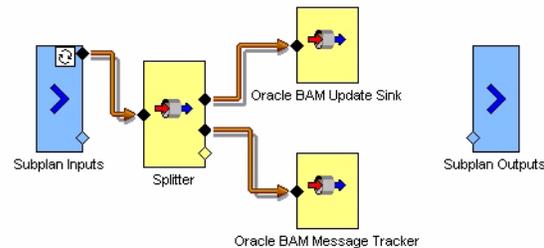
「Oracle BAM Message Tracker」ダイアログで、「Include in Global Transaction」オプションを選択し、変換に使用した名前と同じ名前を入力します（図 5-18）。

図 5-18 「Include in Global Transaction」を選択して名前を指定する



図 5-19 に、反復サブプラン内に表示された変換を示します。

図 5-19 反復サブプラン内の変換



## 5.5 Oracle Service Registry

Oracle Service Registry は、Web サービスの UDDI (Universal Description, Discovery and Integration) レジストリです。クライアントは Oracle Service Registry に問い合わせることで、Web サービスの場所を特定できます。また、クライアントは、Web サービスをレジストリに登録できます。

Oracle Service Registry の詳細は、アプリケーションに付属する製品ドキュメントを参照してください。

## 5.6 Oracle Business Rules

Oracle Business Rules は実行可能なコンポーネントではありません。つまり、プロセスまたはサービスとして実行されません。Oracle Business Rules は、アプリケーションが使用可能なライブラリです。Oracle Business Rules を使用するアプリケーションがある場合、Oracle Business Rules の高可用性は、アプリケーションがデプロイされている環境の高可用性に依存します。Oracle Business Rules は、アクティブ / アクティブまたはアクティブ / パッシブの高可用性環境で動作できます。

Oracle Business Rules の詳細は、『Oracle Business Rules ユーザーズ・ガイド』を参照してください。

この項の項目は次のとおりです。

- [第 5.6.1 項「リポジトリのタイプ」](#)
- [第 5.6.2 項「リポジトリへの WebDAV URL」](#)
- [第 5.6.3 項「Real Application Clusters データベースと Oracle Business Rules」](#)
- [第 5.6.4 項「高可用性環境の Rule Author」](#)

## 5.6.1 リポジトリのタイプ

本番環境では、ファイル・リポジトリではなく、WebDAV ベースのリポジトリを使用する必要があります。WebDAV リポジトリは、ファイル・リポジトリよりも安全です。ファイル・リポジトリは、開発環境でのみ使用してください。

## 5.6.2 リポジトリへの WebDAV URL

Rule Author に WebDAV リポジトリを指定する前に、リポジトリの作成が完了している必要があります (Rule Author ではリポジトリを作成できません)。リポジトリは、Oracle HTTP Server と同じファイル・システム上にある必要があります。Oracle HTTP Server は、ORADAV (mod\_oradav) モジュールを介して WebDAV をサポートします。

次に、Rule Author でリポジトリへの URL を入力するときは、完全修飾の物理ホスト名 (Oracle HTTP Server を実行するホスト) を使用します。

実行時には、リポジトリへの URL パスも指定する必要があります。すべてのノードがリポジトリへのアクセスに使用可能な URL を入力します。

## 5.6.3 Real Application Clusters データベースと Oracle Business Rules

Oracle Business Rules 自体はデータベースを使用しません。ただし、Oracle XML DB に組み込まれている WebDAV サポートを利用して、Oracle XML DB に WebDAV リポジトリを構成できます。

## 5.6.4 高可用性環境の Rule Author

Rule Author は、ルールの作成と変更に使用できるブラウザベースのツールです。これは J2EE アプリケーションのため、複数のインスタンスにデプロイ可能ですが、次の点に注意する必要があります。

- 複数のユーザーによる同じディレクトリの同時編集は安全ではありません。
- ルールの編集には、1 つのインスタンスのみが使用されるようにします。これは、ロード・バランサでスティッキーなルーティングを有効化するなどの方法で強制できます。
- Rule Author にアクセスする際は、URL に物理ホスト名を使用します。WebDAV リポジトリは特定の単一ノード上にあるため、URL には物理ホスト名が必須になります。通常は内部マシンから Rule Author を使用してリポジトリにアクセスするため、これが問題になることはありません。

## 5.7 Oracle Web Services Manager

Oracle Web Services Manager の高可用性に関する説明は、『Oracle Web Services Manager デプロイメント・ガイド』のクラスタ環境での Oracle WSM の構成に関する項を参照してください。

# 第 III 部

---

## 障害時リカバリ

この部の章では、Oracle Application Server Disaster Recovery のソリューションについて説明します。

この部は、次の章で構成されています。

- 第 6 章「OracleAS Disaster Recovery」
- 第 7 章「OracleAS Guard asgctl コマンドライン・リファレンス」
- 第 8 章「手動同期化操作」
- 第 9 章「OracleAS Disaster Recovery サイトのアップグレード手順」
- 第 10 章「DNS サーバーの設定」
- 第 11 章「Secure Shell (SSH) ポート・フォワーディング」



## OracleAS Disaster Recovery

障害時リカバリとは、自然災害またはそれ以外の災害によるサイトの壊滅的な障害からシステムを回復することを指します。これら壊滅的な障害の例として、地震、竜巻、洪水、火事などがあります。さらに、システムの計画停止の管理方法も障害時リカバリに含まれます。障害時リカバリを必要とするほとんどの状況において、障害の解決には個々のハードウェアやサブコンポーネントのレプリケーションだけでなく、サイト全体を対象としたレプリケーションを行う必要があります。これは、Oracle Application Server Disaster Recovery (OracleAS Disaster Recovery) ソリューションについても同様です。

この章では、OracleAS Disaster Recovery ソリューション、その環境の構成および設定方法、およびソリューションの高可用性の向上を目的とした管理方法について説明します。ここでは、本番用およびスタンバイ用の2つのサイト内の OracleAS 中間層および Infrastructure 層の両方を対象にします。スタンバイ・サイトは、本番サイトに対して、同じように対称的に構成される (同じ数のインスタンス) か、または非対称的に構成されます (Infrastructure サービスに対してより少ないインスタンスまたは OracleAS Disaster Recovery 機能のみがサポートされず)。通常の運用では、本番サイトがリクエストを積極的に処理します。スタンバイ・サイトは、本番サイトがホスティングするアプリケーションおよびコンテンツを完全にミラー化した状態またはそれに近い状態で維持されます。

これらのサイトの OracleAS Disaster Recovery 機能は、Oracle Application Server Guard を使用して管理します。Oracle Application Server Guard には、各種の管理タスク (これらの管理コマンドの詳細は、第7章「OracleAS Guard asgctl コマンドライン・リファレンス」を参照) がカプセル化されたコマンドライン・ユーティリティ (asgctl) が用意されています。OracleAS Disaster Recovery ソリューションでは、サイト全体で使用できるシステム・サービスの中でも特に次のサービスを活用します。背後では、OracleAS Guard によって、トポロジ内に分散された OracleAS Recovery Manager (ファイル・システムの構成ファイルの管理) と Oracle Data Guard (OracleAS Infrastructure データベースの管理) が自動的に使用されます。表 6-1 に、OracleAS Disaster Recovery の実行計画と、この Oracle ソフトウェアが背後で使用される方法を要約します。

**表 6-1 OracleAS Disaster Recovery の実行計画の概要**

対象	ツール	目的
中間層構成ファイル	OracleAS Recovery Manager	本番サイトの中間層ノードにある構成ファイルをバックアップしてクローニングし、その構成ファイルをスタンバイ・サイトの中間層ノードでリストアします。
OracleAS Infrastructure 構成ファイル	OracleAS Recovery Manager	本番サイトの OracleAS Infrastructure ノードにある OracleAS 構成ファイルをバックアップしてクローニングし、その構成ファイルをスタンバイ・サイトの OracleAS Infrastructure ノードでリストアします。

**表 6-1 OracleAS Disaster Recovery の実行計画の概要 (続き)**

対象	ツール	目的
OracleAS Infrastructure データベース	Oracle Data Guard	本番サイトの OracleAS Infrastructure データベースにあるアーカイブ・ログを、スタンバイ・サイトの OracleAS Infrastructure データベースへ転送します。ログはすぐには適用されません。

OracleAS 10g リリース 2 (10.1.2.0.2) 以降、OracleAS Disaster Recovery ソリューションの説明に使用する概念をわかりやすくするために、同じ Oracle Internet Directory インスタンスを含む本番サイトまたはスタンバイ・サイトのすべてのファームを示すのに「トポロジ」という用語を使用するようになりました。OracleAS 10g リリース 3 (10.1.3) 用 OracleAS Disaster Recovery ソリューションを除き、OracleAS 10g リリース 2 (10.1.2.0.0) 用 OracleAS Disaster Recovery ソリューションのマニュアルに記載されている以前のファームの概念は、トポロジという用語に置き換えられています。トポロジという用語は、本番サイトに対して同じ Oracle Internet Directory を共有しているすべてのインスタンスを指します。discover topology コマンドは、Oracle Internet Directory に問合せを行い、インスタンスのリストを調べ、本番トポロジを記述するトポロジ XML ファイルを生成します。discover topology within farm コマンドは、Oracle Internet Directory を使用できない場合に使用され、OracleAS Guard によって OPMN が使用されてファーム内のトポロジが検出されます。

**注意：** 他のデータベースにも全体的な障害時リカバリの実行計画を適用する必要があります。そのソリューションとして Oracle Data Guard を使用する必要があります。

これらリカバリの実行計画に加えて、両サイトの構成およびインストールについても説明します。これらのタスクでは、中間層ノードをネーミングする 2 つの異なる方法、さらにはサイト間およびサイト内でホスト名を解決する 2 つの方法について説明します。

OracleAS Disaster Recovery では、OracleAS Guard スイッチオーバー操作を使用してスタンバイ・サイトに切り替えることで、サービスを中断することなく本番サイトを計画停止できます。計画外停止は、OracleAS Guard フェイルオーバー操作を使用してスタンバイ・サイトにフェイルオーバーすることで管理できます。スイッチオーバーおよびフェイルオーバーの手順についてはこの章の第 6.12 項「実行時操作：OracleAS Guard のスイッチオーバーおよびフェイルオーバー操作」で説明します。

この章は、次の項で構成されています。

- 第 6.1 項「Oracle Application Server 10g Disaster Recovery ソリューション」
- 第 6.2 項「OracleAS Disaster Recovery 環境の準備」
- 第 6.3 項「Oracle Application Server のインストールの概要」
- 第 6.4 項「OracleAS Guard と asgctl の概要」
- 第 6.5 項「データベースの認証」
- 第 6.6 項「トポロジの検出、ダンプおよび検証」
- 第 6.7 項「いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」
- 第 6.8 項「冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ内の OracleAS 10.1.3 インスタンスの検出」
- 第 6.9 項「既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除」
- 第 6.10 項「OracleAS Guard 操作：スタンバイ・システムへの 1 つ以上の本番インスタンスのスタンバイ・サイト・クローニング」

- 第 6.11 項「OracleAS Guard の操作 : スタンバイのインスタンス化とスタンバイの同期化」
- 第 6.12 項「実行時操作 : OracleAS Guard のスイッチオーバーおよびフェイルオーバー操作」
- 第 6.13 項「Real Application Clusters データベースを使用しない OracleAS Disaster Recovery の構成」
- 第 6.14 項「Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery」
- 第 6.15 項「OracleAS Guard 操作の監視とトラブルシューティング」
- 第 6.16 項「ワイド・エリア DNS の操作」
- 第 6.17 項「OracleAS Guard コマンドライン・ユーティリティ (asgctl) の使用」
- 第 6.18 項「いくつかの OracleAS Metadata Repository 構成の特別な考慮事項」
- 第 6.19 項「OracleAS Disaster Recovery 環境の特別な考慮事項」

**関連項目：** OracleAS Disaster Recovery ソリューションのインストール手順は、Oracle Application Server のインストール・ガイドを参照してください。

### 地理的に分散された Identity Management Infrastructure のデプロイ

Identity Management (IM) Infrastructure を地理的に分散配置するレプリケーション方法は、アクティブ / アクティブ構成の例になりますが、Oracle Internet Directory (OID)、OracleAS Metadata Repository (MR) および OracleAS Single Sign-On (SSO) がレプリケーション構成で設定され、異なる地域に分散配置される OracleAS Disaster Recovery ソリューションと特徴が似ています。OracleAS Single Sign-On の各サイトでは、そのローカル・サイトに配置されたそれ自体の Oracle Internet Directory と OracleAS Metadata Repository が使用され、アクティブ / アクティブ構成となります。このように類似していることには、2 つの目的があります。1 つは、データベース障害が 1 つのサイトで検出された場合、ユーザー・リクエストが地理的に最も近い領域にルーティングされるように Oracle Internet Directory および OracleAS Single Sign-On サーバーを再構成することです。もう 1 つは、OracleAS Single Sign-On 中間層の障害が検出された場合に、トラフィックがリモート中間層にルーティングされるようにネットワークを再構成することです。しかし、このソリューションでは、Infrastructure データベース内の OracleAS Portal、OracleAS Wireless および Distributed Configuration Management (DCM) スキーマの同期化が行われません。これは、これらのいずれでも、Oracle Internet Directory と OracleAS Single Sign-On の情報に使用するレプリカ・モデルがサポートされていないためです。地理的に分散された Identity Management Infrastructure の詳細は、『Oracle Identity Management 概要および配置プランニング・ガイド』を参照してください。

## 6.1 Oracle Application Server 10g Disaster Recovery ソリューション

Oracle Application Server Disaster Recovery ソリューションは、プライマリ (本番 / アクティブ) とセカンダリ (スタンバイ) という構成を持つ 2 つのサイトからなります。両方のサイトには、同数の中間層と同数の OracleAS Infrastructure ノードおよび同数のコンポーネントがインストールされている場合もされていない場合もあります。つまり、両サイトのインストールで、中間層および OracleAS Infrastructure が同じ場合 (対称トポロジ) も同じでない場合 (非対称トポロジ) もあります。これらのサイトは通常、地理的に分散されます。その場合、Wide Area Network 経由でつながっています。

Oracle Application Server Disaster Recovery ソリューションで重要な点は次のとおりです。

- サイトを運用するためにスタンバイ・サイトで必要なインスタンスの数は、本番サイトと同じである場合 (対称) も、本番サイトより少ない場合 (非対称) もあります。
- Disaster Recovery を設定するには、スタンバイ・サイトに必要なインスタンスのセットを作成およびインストールする必要があります。
- スタンバイ・サイトには、サイトの運用に必要な最小セットのインスタンスが必要です。

この項では、ソリューション全体のレイアウト、関連する主要なコンポーネントおよびそれらのコンポーネントの構成について説明します。この項の内容は次のとおりです。

- 第 6.1.1 項「OracleAS Disaster Recovery の要件」
- 第 6.1.2 項「サポートされる Oracle Application Server リリースとオペレーティング・システム」
- 第 6.1.3 項「サポートされているトポロジ」

## 6.1.1 OracleAS Disaster Recovery の要件

OracleAS Disaster Recovery ソリューションの実装を計画どおり動作させるには、次の要件を満たしている必要があります。

- スタンバイ・サイトの各ホストでは、次の項目を本番サイトの同等のホストと同じにする必要があります。
  - 中間層ホストの場合は、物理ホスト名。

---

**注意：**すでにシステムをインストールしている場合は、スタンバイ・サイトの中間層システムの物理名を変更し、OracleAS Infrastructure の物理ホスト名に仮想ホスト名を作成するだけです（次の項目を参照）。物理ホスト名と仮想ホスト名を変更する手順は、第 6.2.1 項「ホスト名の計画と割当て」を参照してください。物理ホスト名と仮想ホスト名の詳細は、第 1.2.1 項「用語」を参照してください。

---

- OracleAS Infrastructure の仮想ホスト名とポート番号。仮想ホスト名は、インストーラによって表示される「仮想ホストの指定」画面で指定できます。
- ハードウェア・プラットフォーム。
- オペレーティング・システムのリリースとパッチ・レベル。
- Oracle Application Server のすべてのインストールで、Oracle Application Server のインストール・ガイドに記載されている要件を満たしている必要があります。
- Oracle Application Server ソフトウェアは、本番サイトの各ホストとスタンバイ・サイトの同等のホストで、同一のディレクトリ・パスにインストールします。
- 本番サイトのホストとスタンバイ・サイトの同等のホスト間で、次の項目が同じである必要があります。
  - Oracle Application Server をインストールしたユーザーのユーザー名とパスワードが本番サイトのホストとスタンバイ・サイトの同等のホストで、同じである必要があります。
  - 特定のノードに Oracle Application Server をインストールしたユーザーのユーザー ID 番号。
  - 特定のノードに Oracle Application Server をインストールしたユーザーのグループ名。
  - 特定のノードに Oracle Application Server をインストールしたユーザーのグループのグループ ID 番号。
  - 環境プロファイル。
  - シェル（コマンドライン環境）。
  - ノード上にある各 OracleAS インストールの Oracle ホーム・ディレクトリのディレクトリ構造、Oracle ホーム名およびパス。パス内では、シンボリック・リンクをいっさい使用しないでください。

- Oracle Application Server インストール・タイプ (スタンバイ・システムにインストールされているインスタンスはすべて、本番サイトにインストールされているインスタンスと同一である必要があります) :
  - \* 中間層 : J2EE and Web Cache および Portal and Wireless
  - \* OracleAS Infrastructure: Metadata Repository (MR) および Identity Management (IM)

## 6.1.2 サポートされる Oracle Application Server リリースとオペレーティング・システム

OracleAS Guard は、Oracle Application Server 10g (9.0.4)、10g リリース 2 (10.1.2.0.0 および 10.1.2.0.2) および 10g リリース 3 (10.1.3.0.0) をサポートしています。

トポロジ内に Oracle ホームまたは Oracle Application Server インスタンスがあるすべてのシステムに、OracleAS Companion CD 2 にある OracleAS Guard キットをインストールする必要があります。詳細は、Oracle Application Server のインストール・ガイドの OracleAS Disaster Recovery のインストール情報を参照してください。

デフォルトにより、OracleAS Guard は、すでに JDK または JRE 環境をインストールしている各 10.1.2 および 10.1.3 インストール・タイプにインストールされます。OracleAS 10g リリース 3 (10.1.3) の Web サーバーとプロセス管理や、OracleAS 10g リリース 2 (10.1.2.0.2) の OracleAS Portal など、OracleAS Guard がデフォルトによりインストールされていない環境では、OracleAS Companion CD 2 にある OracleAS Guard スタンドアロン・キットをこれらの Oracle ホームにインストールする必要があります。

OracleAS Guard では、基になるサービスによってサポートされている混在環境をサポートしています。たとえば、OracleAS Guard は、アップグレード・シナリオの中で発生する可能性がある OracleAS 10g (9.0.4) と OracleAS 10g リリース 2 (10.1.2) 以降のリリースの混在環境をサポートします。この場合、スタンバイ・サイトで中間層は OracleAS 10g リリース 2 (10.1.2) Oracle ホームにアップグレードしたが、Infrastructure は依然として OracleAS 10g (9.0.4) Infrastructure であることがあります。このリリース混在環境は、本番サイトの中間層 Oracle ホームの OracleAS Disaster Recovery の同等のホームが同じリリースである OracleAS 10g リリース 2 (10.1.2) などにアップグレードされ、かつ、その Infrastructure が依然として前のリリースである OracleAS 10g (9.0.4) Infrastructure などのままであるかぎり動作します。つまり、トポロジ内のすべての Oracle ホームの同等の中間層のリリース番号が、すべての同等の中間層 (つまりスタンバイ・サイトと本番サイトの両方) のリリース番号と完全に一致する必要があります。また、Infrastructure もスタンバイ・サイトと本番サイトの両方でリリース番号が完全に一致する必要があります。Oracle AS Guard も両方のサイトで同じバージョンである必要があります。ただし、アップグレードに基づいた要件として、Infrastructure は中間層よりも前のリリースにすることができます。それは、この状況がアップグレード・シナリオの中で発生するためです。

## 6.1.3 サポートされているトポロジ

OracleAS Disaster Recovery では、本番サイトとスタンバイ・サイトの Infrastructure と中間層の構成についていくつかの基本トポロジがサポートされています。OracleAS Disaster Recovery でサポートされている基本トポロジは次のとおりです。

- 対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による本番サイトの完全なミラー
- 非対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による単純な非対称のスタンバイ・トポロジ
- OracleAS Portal の OracleAS Metadata Repository は別の場所にあり、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure (部門別トポロジ)
- アプリケーション OracleAS Metadata Repository が分散され、Oracle Identity Management と OracleAS Metadata Repository が同じ場所のない Infrastructure

- 冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ
- 既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合された冗長な単一 OracleAS 10.1.3 Oracle ホーム J2EE トポロジ

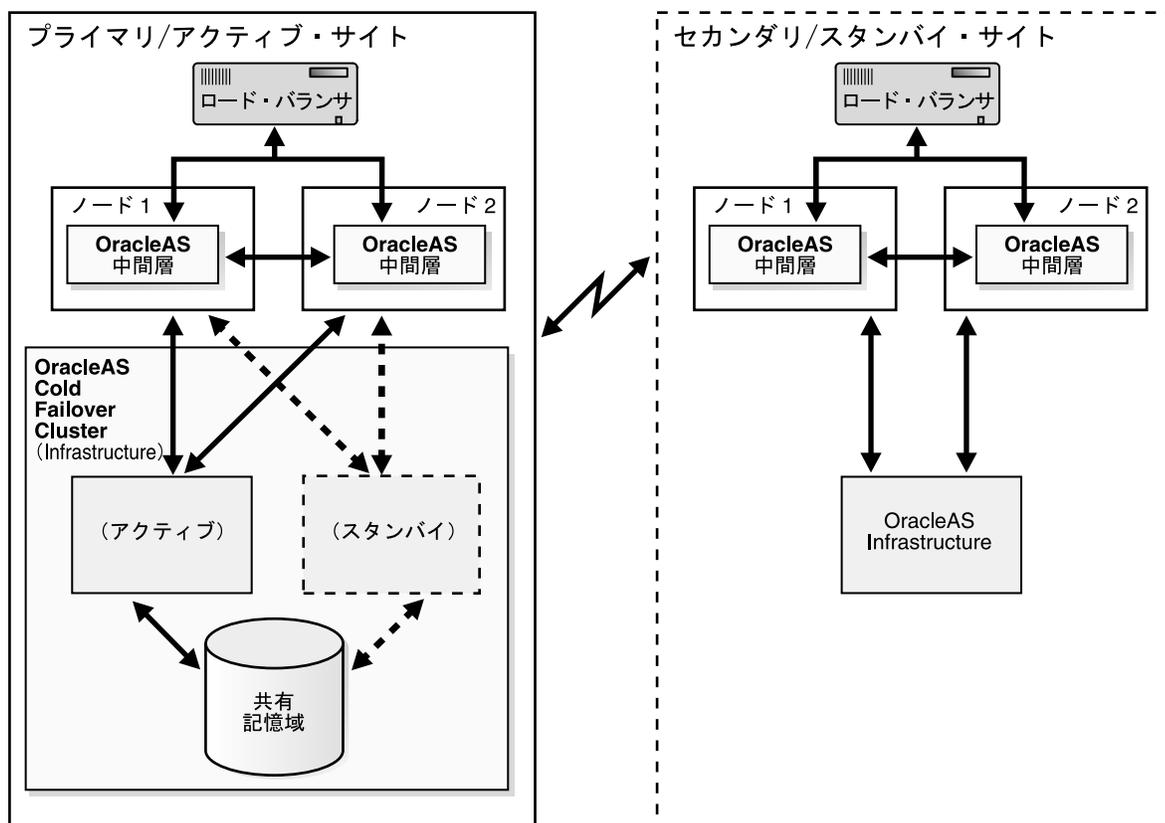
### 6.1.3.1 対称トポロジ : Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による本番サイトの完全なミラー

OracleAS Disaster Recovery 10g リリース 2 (10.1.2.0.1) の場合、サポートされていたのは、OracleAS Disaster Recovery の対称トポロジ環境だけでした。この OracleAS Disaster Recovery 環境には、2つの主要な要件があります。

- この配置では、OracleAS Metadata Repository と Oracle Identity Management が同じ場所にある、1つのデフォルトの Infrastructure インストールを使用する必要があります。
- スタンバイ・サイトは、本番サイトと同数のインスタンスを持つ本番サイトの完全なミラーにする必要があります (対称トポロジ)。

図 6-1 は、プライマリ・サイトに Cold Failover Cluster がある、対称トポロジを持つ OracleAS Disaster Recovery ソリューションの例を示しています。これは、Oracle Application Server の視点から見ると、両方のサイトに2つの OracleAS 中間層と1つの Infrastructure が含まれているため、対称トポロジとみなされます。

図 6-1 Oracle Application Server のサイト間 Disaster Recovery ソリューションの例 (中間層ノードが1つの場合は、ロード・バランサ機器はオプション)



OracleAS Disaster Recovery ソリューションの構成および運用手順では、本番サイトにおける、任意の数の中間層インストールがサポートされます。スタンバイ・サイトには、同じ数の中間層インストールを配置する必要があります。本番サイトとスタンバイ・サイトでは、中間層を相互にミラー化します。

OracleAS Infrastructure の場合は、本番サイトとスタンバイ・サイトで同じ数のインストールが必要とされません（名前またはインスタンスは同一である必要があります）。たとえば、[図 6-1](#) に示すように、本番サイトには OracleAS Cold Failover Cluster (Infrastructure) ソリューションを配置し、スタンバイ・サイトには、1 つのノードに OracleAS Infrastructure インストールを配置できます。このように、OracleAS Cold Failover Cluster を使用して本番サイトの OracleAS Infrastructure をホストの障害から保護します。また、このソリューションでは仮想ホスト名を利用して、ハードウェアの冗長性を実現できます。OracleAS Cold Failover Cluster の詳細は、OracleAS 10g リリース 2 (10.1.2.0.2) のドキュメント・セットに含まれている『Oracle Application Server 高可用性ガイド』の第 6.2.2 項「アクティブ / パッシブ型の高可用性トポロジ」を参照してください。

OracleAS Disaster Recovery ソリューションは、様々な単一サイトの Oracle Application Server アーキテクチャを拡張したものです。たとえば、単一サイトのアーキテクチャには、OracleAS Cold Failover Cluster (Infrastructure) と、アクティブ / アクティブ構成の Oracle Application Server 中間層アーキテクチャを組み合わせたものがあります。どのような単一サイト・アーキテクチャがサポートされているかについての最新情報は、Oracle Technology Network (OTN) の Web サイトで最新の動作要件を確認してください。

[http://www.oracle.com/technology/products/ias/hi\\_av/index.html](http://www.oracle.com/technology/products/ias/hi_av/index.html)

OracleAS Disaster Recovery の対称ソリューションの重要な特徴は、次のとおりです。

- 中間層インストールは、本番サイトとスタンバイ・サイト間で同じです。つまり、本番サイトの中間層の各インストールは、スタンバイ・サイトに同一インストールを持ちます。中間層ノードは複数にすることをお勧めします。これにより、それぞれのサイトの中間層インストールの各セットを冗長にできるためです。中間層インストールが複数のマシンにあるため、そのサイトでの障害および停止はクライアントにとって透過的になります。
- OracleAS Disaster Recovery ソリューションのサイト構成は同一にする必要があります。これは、プロセスおよび処理手順を両サイトで同一にすることで、運用タスクの実行および維持が容易になるためです。また同一のサイト構成は、Oracle Application Server コンポーネント構成ファイルのサイト間での同期化を手動で行う際にエラーを防ぎます。
- 障害が原因で本番サイトが使用不能になったときは、適切な時間内に、スタンバイ・サイトが運用可能になります。クライアント・リクエストは、本番として運用されているサイトに必ずルーティングされます。サイトの停止に伴うフェイルオーバーまたはスイッチオーバー処理が実行された後、クライアント・リクエストは運用を引き継いだ別のサイトへとルーティングされます。対称トポロジの場合、新しい本番サイトが提供するサービスの質は、停止前に元の本番サイトが提供していたサービスと同じになります。
- 1 つのサイトの観点では、サイトはアクティブ / パッシブ構成で設定されます。アクティブ / パッシブ設定は、本番として使用される 1 つのプライマリ・サイトと最初はパッシブ（スタンバイ）になっている 1 つのセカンダリ・サイトで構成されます。セカンダリ・サイトは、フェイルオーバーまたはスイッチオーバー操作が実行された後のみアクティブになります。両サイトには対称性があるため、フェイルオーバーまたはスイッチオーバー後は、元のスタンバイ・サイトを新しい本番サイトとしてアクティブな状態に維持できます。OracleAS Disaster Recovery のサイトの要件が満たされている場合は、修復またはアップグレードが終わると、元の本番サイトは新しいスタンバイ・サイトになります。どちらのサイトも、クライアントに対して同じレベルのサービスを提供します。アクティブ / パッシブ設定では、スタンバイ・サイトは、Disaster Recovery 環境で使用している Oracle ホームがパッシブ（非アクティブ）であるかぎり、同じホストでアクティブになることができる個別の Oracle ホームで構成できます。

- この後で説明するデータベース・リカバリ (DBR) サイト (OracleAS 10g リリース 3 (10.1.3) サイトは通常含まれないが、OracleAS 10g リリース 2 (10.1.2.0.2) サイトは含まれる) では、スタンバイとして運用されるサイトには Oracle Data Guard が調整する Oracle Application Server Infrastructure のフィジカル・スタンバイが含まれています。OracleAS Guard では、Oracle Data Guard の構成と使用を、OracleAS Infrastructure 構成ファイルのバックアップおよびリストア手順とともに自動化し、本番サイトとスタンバイ・サイト間の構成を同期します。スイッチオーバーおよびフェイルオーバー操作は、2 つのサイトの OracleAS Infrastructure 間で、運用形態の切替えを可能にします。OracleAS Disaster Recovery ソリューションでのクローニング、インスタンス化、同期化、スイッチオーバー、フェイルオーバーなどの OracleAS Guard 管理タスクを実行する asgctl コマンドライン・インタフェースの使用の詳細は、第 6.10 項「OracleAS Guard 操作: スタンバイ・システムへの 1 つ以上の本番インスタンスのスタンバイ・サイト・クローニング」、第 6.11 項「OracleAS Guard の操作: スタンバイのインスタンス化とスタンバイの同期化」、第 6.12 項「実行時操作: OracleAS Guard のスイッチオーバーおよびフェイルオーバー操作」および第 6.17 項「OracleAS Guard コマンドライン・ユーティリティ (asgctl) の使用」を参照してください。

### 6.1.3.2 非対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による単純な非対称のスタンバイ・トポロジ

OracleAS Disaster Recovery 10g リリース 2 (10.1.2.0.2) 以降、サポート対象となる非対称トポロジには、次の単純な非対称スタンバイ・トポロジも含まれています。

- リソースの数が本番サイトより少ない (中間層の数が少ない) スタンバイ・サイト。これはスケール変更以外のすべての本番サービスのサポートを意味します。この方法では、スケール変更以外のすべてのサービスの維持が保証されます (この OracleAS Disaster Recovery ソリューションの例は、図 6-2 を参照)。

図 6-2 リソースの数が少ない単純な非対称スタンバイ

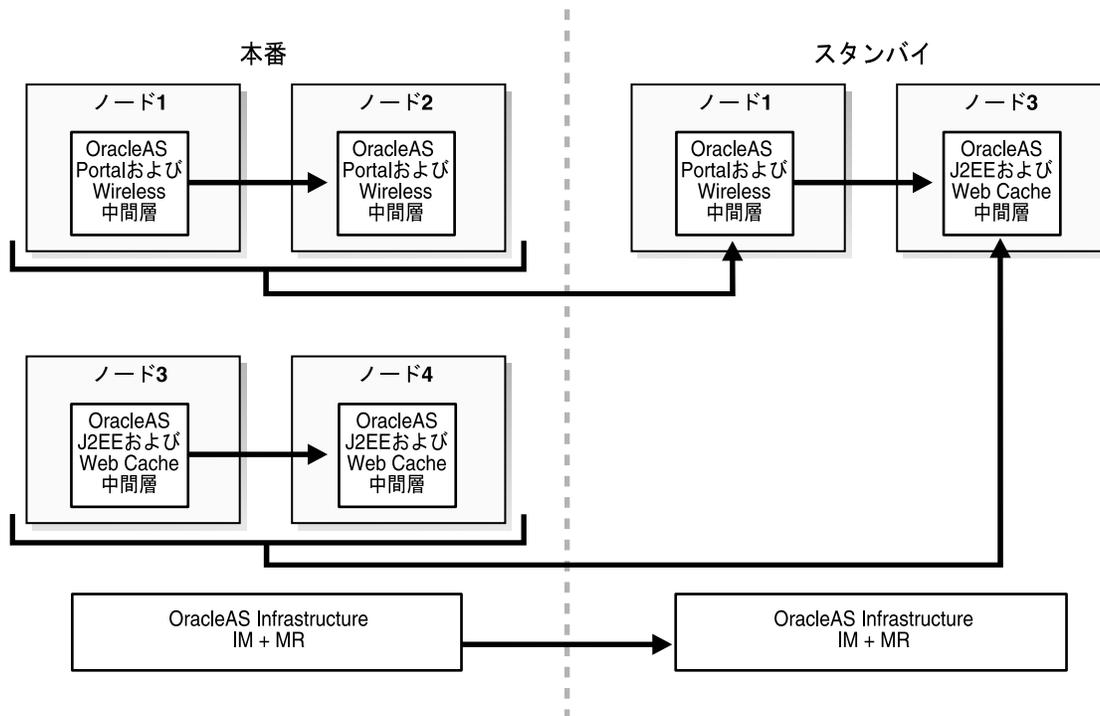


図 6-2 は、4つの中間層インスタンスと1つの Infrastructure (Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある) で構成された本番サイトを示しています。この例では、中間層 1 に配置されたサービスとアプリケーションは、中間層 2 を含むようにスケール変更されます。さらに、中間層 3 に配置されたサービスとアプリケーションは、中間層 4 を含むようにスケール変更されます。障害時リカバリのための本番サイトより少ない数のリソースの要件を満たすために、スタンバイ・サイトでのスケール変更は必要ありません。したがって、本番中間層 1 と 2 に配置されたサービスは、スタンバイ・サイトの障害時リカバリの同等の中間層 1 によって満たされ、このスタンバイ・サイトの中間層 1 は本番中間層 1 と同期化されます。同様に、本番中間層 3 と 4 に配置されたサービスは、スタンバイ・サイトの障害時リカバリの同等の中間層 3 によって満たされ、このスタンバイ・サイトの中間層 3 は本番中間層 3 と同期化されます。

- Infrastructure サービスのためにのみ OracleAS Disaster Recovery サポートを維持しているスタンバイ・サイト。中間層インスタンスは本番サイトの管理を介してのみサポートされます。この方法で保証されるのは、Infrastructure サービスの維持のみです (この OracleAS Disaster Recovery ソリューションの例は、図 6-3 を参照)。

図 6-3 Infrastructure が保証された単純な非対称スタンバイ

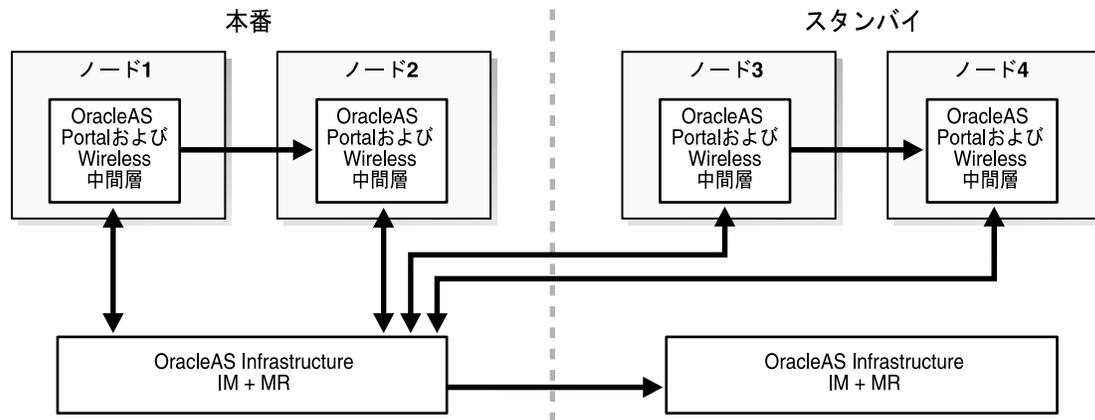


図 6-3 は、4つの中間層インスタンスで構成された本番サイトを示しています。そのうちの2つの中間層 (1 と 2) と本番の Infrastructure サービスが同じ場所にあり、残りの2つの中間層 (3 と 4) がスタンバイ・サイトにリモート配置されています。スタンバイ・サイトは、Infrastructure サービスに対してのみ障害時リカバリの機能を提供するために使用されます。この構成では、中間層リソースは、アクティブ / アクティブ・モデルで、技術的には1つの本番トポロジとして構成されます。

通常の状態では、アプリケーション・リクエストに対して中間層 1 ~ 4 がサービスを提供できます。このモデルでは、中間層 3 と 4 に配置されたサービスとアプリケーションは、このトポロジに関連する待機時間、ファイアウォールおよびネットワークの問題に対応できると想定されます。障害時リカバリの運用では、維持する必要があるのは Infrastructure サービスのみです。中間層インスタンスの配置とメンテナンスは、通常の本番サイトの管理によって行われます。

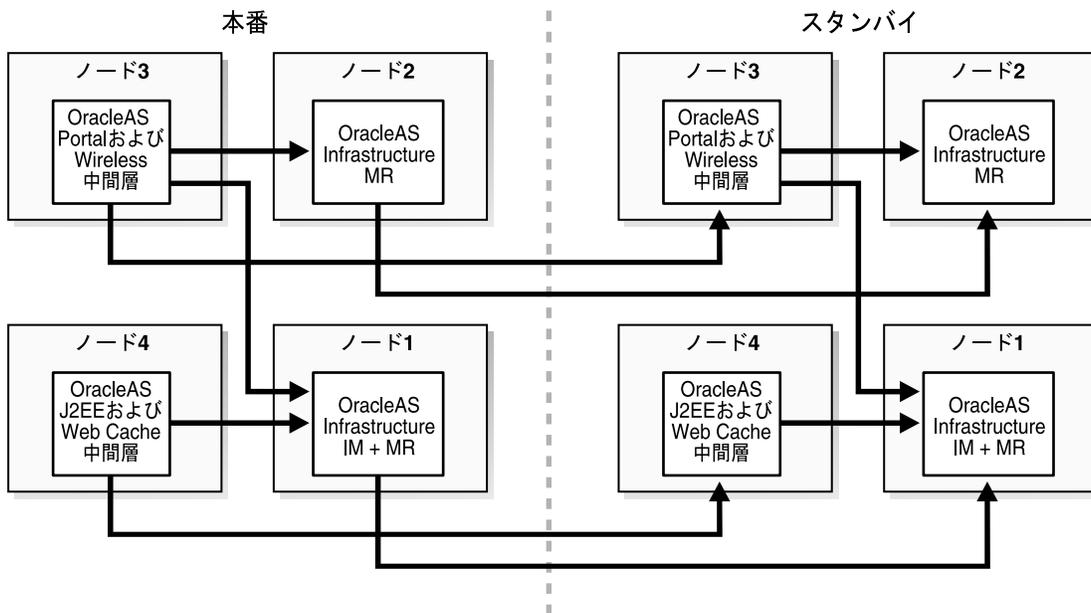
一般的に、非対称トポロジのサポートとは、OracleAS Disaster Recovery のスタンバイ・サイトでは、リソースの数が本番サイトより少なく (または少ない可能性があり)、縮小した Oracle ホームを維持し、最小レベルのサービス機能が保証されることを意味します。

### 6.1.3.3 OracleAS Portal の OracleAS Metadata Repository は別の場所であり、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure (部門別トポロジ)

このトポロジ (図 6-4) は、2つの OracleAS Metadata Repository を持つ1つの OracleAS Infrastructure と複数の中間層で構成されています。1つの OracleAS Metadata Repository は Oracle Identity Management コンポーネント (Oracle Internet Directory や OracleAS Single Sign-On など) によって使用されます。すべての中間層および拡張されたときにこのトポロジに追加される中間層は、Oracle Identity Management サービスに、この OracleAS Metadata Repository を使用します。もう1つの OracleAS Metadata Repository は、OracleAS Portal および OracleAS Wireless 中間層コンポーネントによって製品メタデータのために使用されます。2つのメタデータ・リポジトリを持つこの配置は2つの DCM 本番ファームといえます。

OracleAS Disaster Recovery のスタンバイ構成は、第 6.1.3.1 項「対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による本番サイトの完全なミラー」の説明のように対称トポロジとして設定すること (2つの DCM スタンバイ・ファームの構成が必要) も、第 6.1.3.2 項「非対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による単純な非対称のスタンバイ・トポロジ」の説明のように単純な非対称トポロジとして設定すること (最低1つの DCM スタンバイ・ファームの構成が必要なサービスを保証) もできます。

図 6-4 同じ場所にある Oracle Identity Management および OracleAS Metadata Repository と、別の場所にある OracleAS Metadata Repository



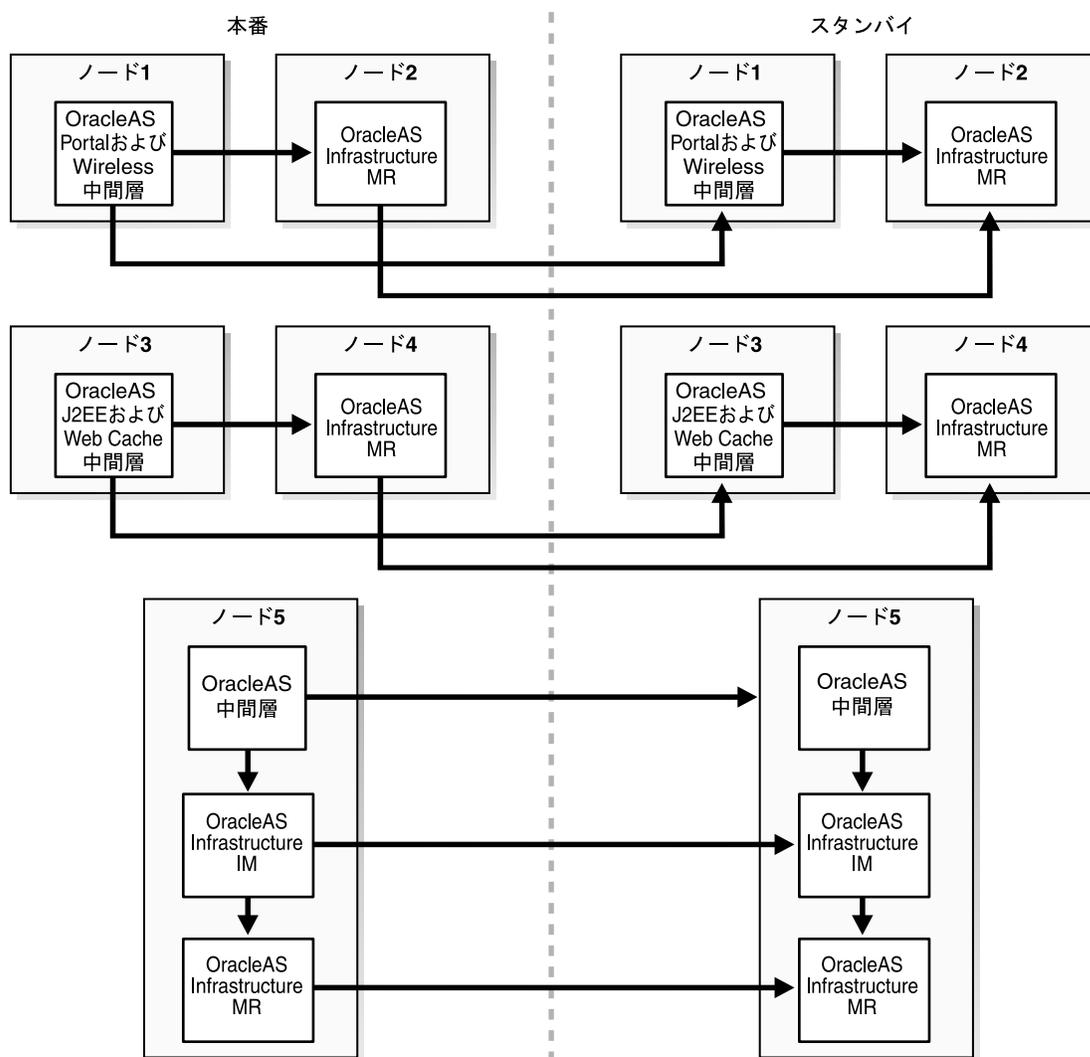
### 6.1.3.4 アプリケーション OracleAS Metadata Repository が分散され、Oracle Identity Management と OracleAS Metadata Repository が同じ場所のない Infrastructure

第 6.1.3.1 項「対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による本番サイトの完全なミラー」、第 6.1.3.2 項「非対称トポロジ: Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure による単純な非対称のスタンバイ・トポロジ」および第 6.1.3.3 項「OracleAS Portal の OracleAS Metadata Repository は別の場所であり、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure (部門別トポロジ)」のトポロジは、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure によるデフォルトのデータベース・リポジトリの配置を示しています。第 6.1.3.3 項「OracleAS Portal の OracleAS Metadata Repository は別の場所であり、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にある Infrastructure (部門別トポロジ)」は、OracleAS Metadata Repository が別の場所にあるトポロジも示しています。

アプリケーション OracleAS Metadata Repository が分散し、Oracle Identity Management と OracleAS Metadata Repository Infrastructure が同じ場所のないトポロジでは、Oracle Identity Management Infrastructure と 1 つの OracleAS Metadata Repository Infrastructure は別々のホストにインストールされ、他の OracleAS Metadata Repository は異なるホストに、それぞれのアプリケーションと常駐するようにインストールされます。したがって、1 つの OracleAS Metadata Repository は、1 つのデフォルト Infrastructure インストールを使用した配置の結果であり、1 つ以上の OracleAS Metadata Repository は、OracleAS ユーザーが、管理またはポリシー、あるいはその両方の理由のために、OracleAS Metadata Repository Creation Assistant などのツールを使用して 1 つ以上のアプリケーション OracleAS Metadata Repository をアプリケーション・データを持つ 1 つ以上のシステムにインストールした結果である可能性があります。

図 6-5 は、Oracle Identity Management と OracleAS Metadata Repository が同じ場所のない Infrastructure で、OracleAS Metadata Repository が分散している OracleAS Disaster Recovery ソリューションの例を示しています。

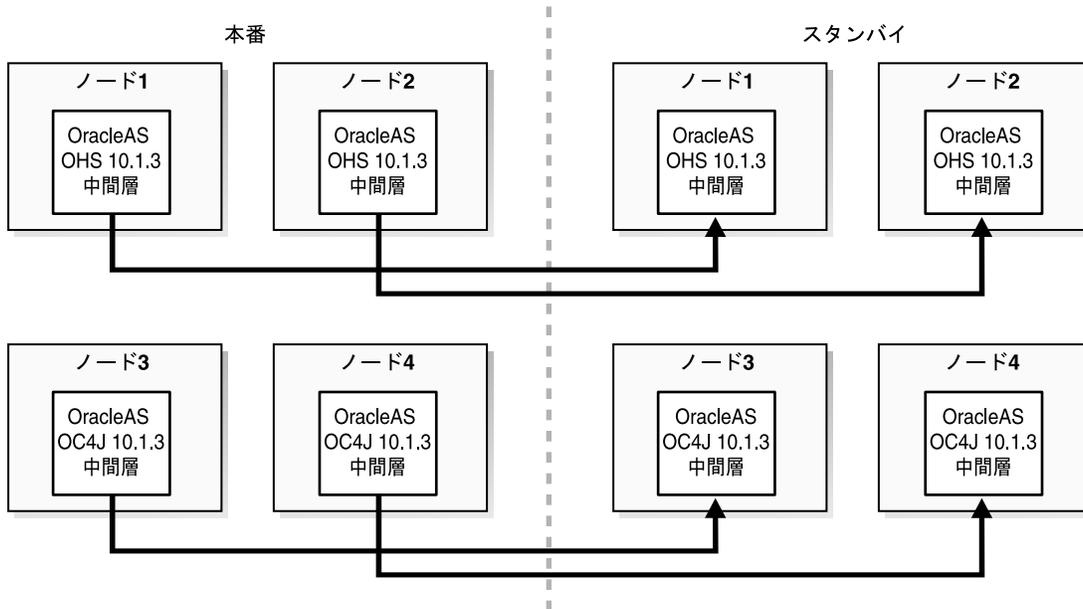
図 6-5 Oracle Identity Management (IM) と OracleAS Metadata Repository (MR) が同じ場所のない Infrastructure によるトポロジで、OracleAS Metadata Repository が分散している



### 6.1.3.5 冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ

図 6-6 は、クラスタ（または OPMN インスタンス）と呼ばれる、4つのノードで構成される OracleAS 10.1.3 トポロジにおける J2EE サーバーの高可用性をサポートする構成を持つ、OracleAS Disaster Recovery ソリューションの例を示しています。Web サーバー（Oracle HTTP Server つまり OHS）とプロセス管理（OPMN）のインストール・タイプはノード 1 と 2 にインストールされ、J2EE サーバー（OC4J）とプロセス管理（OPMN）のインストール・タイプはノード 3 と 4 にインストールされます。Identity Management はインストールされません。

図 6-6 冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ



OracleAS 10.1.3 以降では、新しい動的なノード検出メカニズムが、OPMN のコンポーネントである Oracle Notification Server (ONS) 内で稼動しています。動的なノード検出により、クラスタによる自己管理が可能になっています。新しい ONS ノードがクラスタに追加されると、既存の各 ONS ノードはマルチキャスト・メッセージを使用して新しいノードの存在を通知し、新しいノードとその接続情報を現在のクラスタのマップに追加します。同時に、新しい ONS ノードのマップには既存のノードすべてが追加されます。これによって、OracleAS クラスタが現在構成されている OracleAS Disaster Recovery の要件の 1 つが満たされます。この場合、OPMN 構成ファイル `opmn.xml` は、新しい OracleAS サーバー・ノードがクラスタに追加されるか、クラスタから削除されるたびに更新されます。このクラスタリング構成は、OHS や OC4J をはじめとする、ノードにインストールされている OracleAS サーバー・コンポーネントのインスタンスすべてに適用されます。

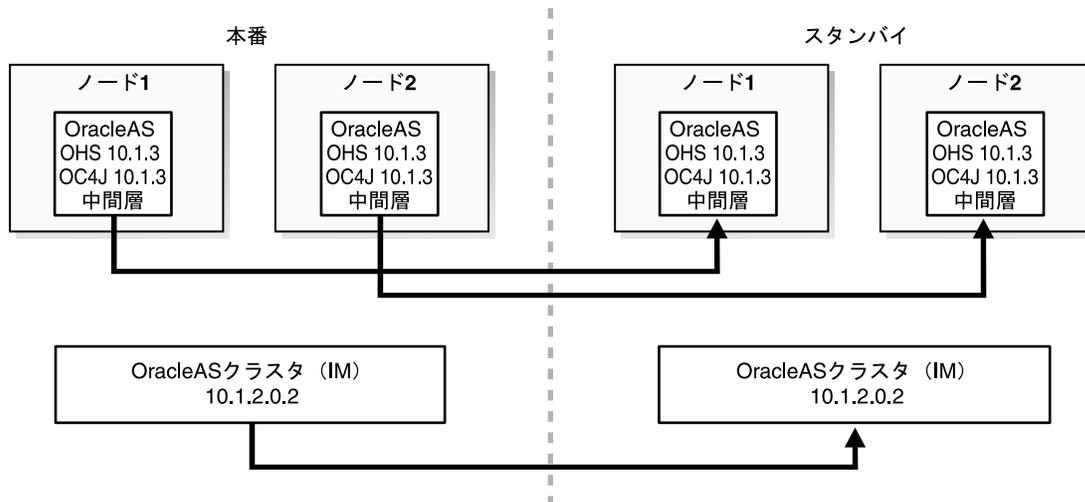
デフォルトでは、OracleAS Guard は OracleAS クラスタ内の各ノードの各 Oracle ホームにインストールされます。OracleAS Guard クライアントが OracleAS Guard サーバーに接続され、Disaster Recovery 管理者が `discover topology within farm` コマンドを実行すると、OracleAS Guard では OPMN を利用してクラスタ内のノードのインスタンスをすべて検出し、OracleAS Guard サーバーに Disaster Recovery XML トポロジ・ファイルを作成し、このファイルを Disaster Recovery 本番トポロジのシステムすべてに伝播します。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ・トポロジ全体に本番トポロジ・ファイルが自動的に伝播されます。

その後 Disaster Recovery 管理者は、OracleAS クラスターの 1 つ以上のノードを追加または削除した後に、**discover topology within farm** コマンドを使用できます。クラスタ構成情報は、ONS の動的なノード検出メカニズムによって自動的に管理され、常に最新状態に更新されます。ONS の動的な検出メカニズムの詳細は、『Oracle Containers for J2EE 構成および管理ガイド』を参照してください。**add instance** および **remove instance** コマンドを使用してローカル・トポロジ・ファイルのインスタンスを管理し、指定に応じて、更新されたこのローカル・トポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播することもできます。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ環境全体に本番トポロジ・ファイルが自動的に伝播されます。ユースケース例については、第 6.8 項「冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ内の OracleAS 10.1.3 インスタンスの検出」を参照してください。したがって、ノードがクラスタに追加されているか、Disaster Recovery 環境の一部であるこれらのノードにインスタンスが追加されている場合には、**discover topology within farm** コマンドを実行することが重要です。

### 6.1.3.6 既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合された冗長な単一 OracleAS 10.1.3 Oracle ホーム J2EE トポロジ

図 6-7 は、バージョンが混在したトポロジをサポートしている OracleAS Disaster Recovery ソリューションの例を示しています。この例では、OracleAS 10.1.3 のインスタンスが既存の OID ベースのトポロジ (OracleAS Cluster (IM) 10.1.2.0.2) に統合されています。ここでは、統合された Web サーバー、J2EE サーバーおよび OPMN インストール・オプションを単一の Oracle ホームに含む複数の OracleAS 10g リリース 3 (10.1.3) を個別のシステムにインストールすることで、冗長な単一の OracleAS 10.1.3 Oracle ホーム J2EE トポロジが実現されています。

図 6-7 冗長な単一の Oracle ホーム J2EE トポロジと OracleAS Cluster (IM) 10.1.2.0.2 (OracleAS 混在バージョン・トポロジ)



デフォルトでは、OracleAS Guard は各 Oracle ホームにインストールされます。OracleAS Guard クライアントが IM 10.1.2.0.2 OracleAS トポロジの OracleAS Guard サーバーに接続し、**discover topology** コマンドを実行すると、OracleAS Guard では OID を利用して、既存の OID ベース・トポロジ内の OracleAS 10.1.2.0.2 インスタンスすべてを自動的に認識します。この検出トポロジ操作によって、Disaster Recovery トポロジ・ファイルが作成され、本番トポロジのインスタンスすべてに伝播されます。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ・トポロジ全体に本番トポロジ・ファイルが自動的に伝播されます。

Disaster Recovery の管理者は、`asgctl add instance` または `remove instance` コマンドを使用して、ローカル・トポロジ・ファイルの単一の OracleAS 10.1.3 J2EE インスタンスを既存の OID ベースの 10.1.2.0.2 本番トポロジに追加するか、10.1.2.0.2 本番トポロジから削除できます。どちらの操作を実行する場合も、ローカル・トポロジ・ファイルが更新され、指定に応じてローカルの更新済トポロジ・ファイルが本番トポロジのインスタンスすべてに伝播されます。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ・トポロジ全体に本番トポロジ・ファイルが自動的に伝播されます。最終的なトポロジは、既存の OID ベース・トポロジ (OracleAS Cluster (IM) 10.1.2.0.2) に統合された冗長な単一の OracleAS 10.1.3 Oracle ホーム J2EE になります。ユースケース例については、第 6.9 項「既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除」を参照してください。トポロジ・ファイルの詳細は、第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。

## 6.2 OracleAS Disaster Recovery 環境の準備

OracleAS Disaster Recovery ソリューション用に OracleAS ソフトウェアをインストールする前に、システム・レベルで必要な構成やオプションの構成を行います。これらの構成を実行するタスクには、次のものがあります。

- 第 6.2.1 項「ホスト名の計画と割当て」
- 第 6.2.2 項「ホスト名解決の構成」
- 第 11 章「Secure Shell (SSH) ポート・フォワーディング」(オプション)

この項では、非対称トポロジで、これらのタスクの実行に必要な手順について説明します。同じ手順は、単純な非対称スタンバイ・サイトにも適用できます。また、OracleAS Metadata Repository が分散されているかどうかに関係なく、Oracle Identity Management と OracleAS Metadata Repository が同じ場所にないトポロジにも適用できます。

### 6.2.1 ホスト名の計画と割当て

物理ホスト名およびネットワーク・ホスト名の設定手順を実行する前に、OracleAS Disaster Recovery ソリューション全体で使用する物理ホスト名およびネットワーク・ホスト名を計画します。ホスト名の計画および割当てにおける全体的なアプローチでは、次の目標を達成する必要があります。

- 中間層および OracleAS Infrastructure の OracleAS コンポーネントが、本番サイトまたはスタンバイ・サイトのどちらにあるかに関係なく、構成設定で同一の物理ホスト名を使用できる必要があります。また、OracleAS Infrastructure の物理ホスト名の仮想ホスト名も作成する必要があります。

たとえば、本番サイトの中間層コンポーネントが同じサイト内のホストへのアクセスに名前 `asmid1` を使用するのであれば、スタンバイ・サイトの同じコンポーネントはこの同じ名前を使用して、スタンバイ・サイト内にある `asmid1` と同等のホストにアクセスする必要があります。同様に、本番サイトの OracleAS Infrastructure の仮想ホスト名に `infra` という名前を使用する場合は、スタンバイ・サイトの OracleAS Infrastructure の物理ホスト名にも `infra` という仮想ホスト名を使用する必要があります。

- スタンバイ・サイトが本番サイトを引き継ぐ際に、ホスト名 (物理、ネットワークまたは仮想) の変更が必要ないこと。ただし、DNS スイッチオーバーを実行する必要があります。詳細は、第 6.16 項「ワイド・エリア DNS の操作」を参照してください。

---

**注意：** 本番サイトとスタンバイ・サイトの物理ホスト名は両サイト間で統一する必要がありますが、物理ホスト名を解決した後の適切なホストは異なってもかまいません。ホスト名解決は、第 6.2.2 項「ホスト名解決の構成」で説明します。

---

図 6-8 は、ホスト名の計画と割当てのプロセスを示しています。

図 6-8 本番サイトとスタンバイ・サイトの名前割当ての例

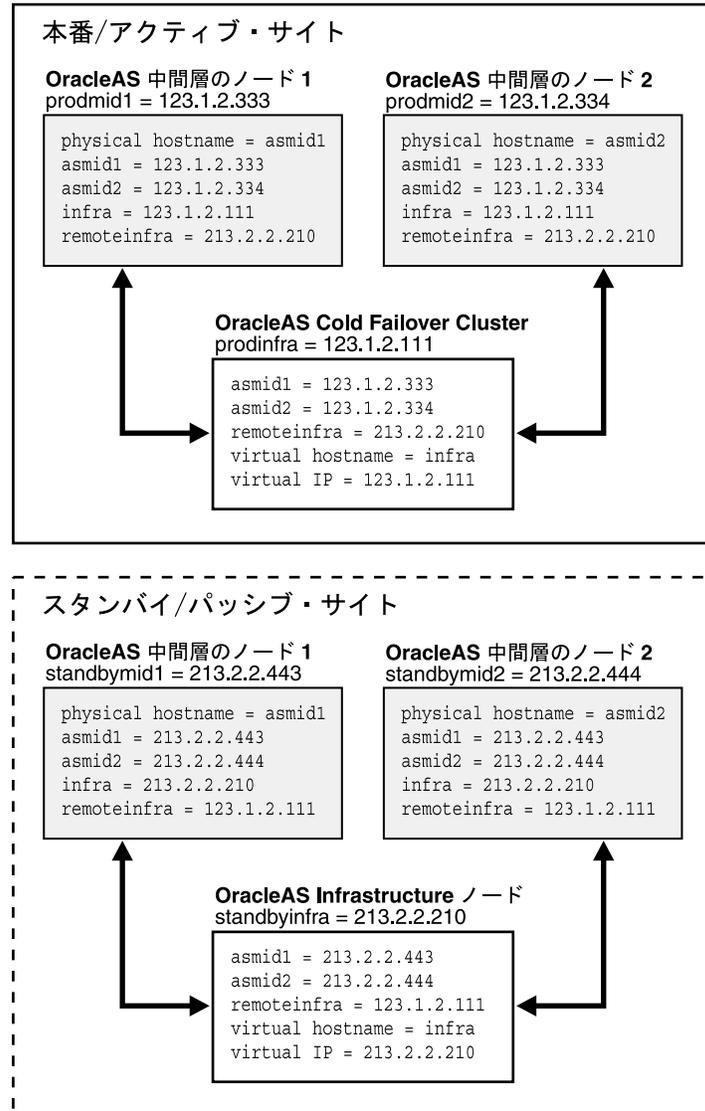


図 6-8 では、本番サイトに 2 つの中間層ノードがあります。OracleAS Infrastructure は単一ノードまたは OracleAS Cold Failover Cluster ソリューション（単一ノードの OracleAS Infrastructure と同様、1 つの仮想ホスト名と仮想 IP で表される）にすることができます。2 つのサイトの一般名は、中間層ノードの物理ホスト名と OracleAS Infrastructure の仮想ホスト名です。表 6-2 は、例における物理、ネットワークおよび仮想ホスト名を示しています。

表 6-2 図 6-8 の物理、ネットワークおよび仮想ホスト名

物理ホスト名	仮想ホスト名	ネットワーク・ホスト名
asmid1	-	prodmid1, standbymid1
asmid2	-	prodmid2, standbymid2
- <sup>1</sup>	infra	prodmid1, standbymid1

<sup>1</sup> この例では、物理ホスト名はネットワーク・ホスト名になります。そのため、このネットワーク・ホスト名を、`asgctl` コマンドの `<host>`、`<host-name>` または `<standby_ topology_host>` パラメータ引数に使用します。

---



---

**注意：** すべての OracleAS Guard asgctl コマンドでは、システムのホスト名を指定できるパラメータがある場合、システムの物理ホスト名を指定して実行する必要があります。

---



---

■ 非 OracleAS アプリケーションの共同ホスティング

本番サイトに非 OracleAS アプリケーションを実行しているホストがあり、そのホストで OracleAS を共同ホスティングする場合、このホストの物理ホスト名を変更すると、非 OracleAS アプリケーションが使用不能になることがあります。このようなケースでは、本番サイトのホスト名はそのままにして、スタンバイ・サイトの物理ホスト名を本番サイトと同じホスト名に変更します。その後で、非 OracleAS アプリケーションをスタンバイ・ホストにもインストールすることで、このホストは非 OracleAS アプリケーションに対するスタンバイの役割も果たせます。

第 1.2.1 項「用語」の説明にあるように、OracleAS Disaster Recovery ソリューションでの物理ホスト名、ネットワーク・ホスト名および仮想ホスト名の用途は他のアプリケーションとは異なります。また、これらの設定方法も異なります。次の項では、様々なタイプのホスト名の設定方法について説明します。

### 6.2.1.1 物理ホスト名

本番サイトとスタンバイ・サイトの両方の各ホストで、中間層ホストの物理ホスト名を変更する必要があります。

Solaris で、ホストの物理ホスト名を変更する手順は次のとおりです。

---



---

**注意：** 他の UNIX 系の OS については、各手順で使用する同等のコマンドをシステム管理者に問い合せてください。

---



---

1. 既存のホスト名の設定を次のようにしてチェックします。

```
prompt> hostname
```

2. vi などのテキスト・エディタを使用して、/etc/nodename の名前を、計画した物理ホスト名に編集します。
3. 中間層ホストごとに、ホストをリブートし変更を有効にします。
4. 手順 1 を再度実行し、ホスト名が正しく設定されていることを確認します。
5. 本番サイトとスタンバイ・サイトの各ホストで、前述の手順を繰り返します。

Windows で、ホストの物理ホスト名を変更する手順は次のとおりです。

---



---

**注意：** 使用されている Windows のバージョンによっては、この後の手順で説明されているユーザー・インタフェースの項目が、実際のものとは異なる場合があります。

---



---

1. 「スタート」メニューから「コントロールパネル」を選択します。
2. 「システム」アイコンをダブルクリックします。
3. 「詳細」タブを選択します。
4. 「環境変数」を選択します。
5. インストール担当者のアカウントの「ユーザー環境変数」で、「新規」を選択して変数を新規作成します。
6. 変数名として "\_CLUSTER\_NETWORK\_NAME\_" を入力します。
7. この変数の値として、計画した物理ホスト名を入力します。

### 6.2.1.2 ネットワーク・ホスト名

OracleAS Disaster Recovery ソリューションで使用されるネットワーク・ホスト名は、ドメイン・ネーム・システム (DNS) に定義されます。これらのホスト名はソリューションで使用されるネットワークで参照可能で、DNS により、DNS システムで割り当てられた IP アドレスを使用して適切なホストへと解決されます。そのため、ネットワーク・ホスト名と対応する IP アドレスを、DNS システムに追加する必要があります。

図 6-8 の例では、本番サイトとスタンバイ・サイトを有するネットワーク全体を管理する DNS システムに、次のエントリを追加する必要があります。

```
prodmid1.oracle.com      IN      A       123.1.2.333
prodmid2.oracle.com      IN      A       123.1.2.334
prodmid3.oracle.com      IN      A       123.1.2.335
prodinfra.oracle.com     IN      A       123.1.2.111
standbymid1.oracle.com   IN      A       213.2.2.443
standbymid2.oracle.com   IN      A       213.2.2.444
standbyinfra.oracle.com  IN      A       213.2.2.210
```

### 6.2.1.3 仮想ホスト名

第 1.2.1 項「用語」の項で定義したように、仮想ホスト名は OracleAS Infrastructure にのみ適用されます。仮想ホスト名は、OracleAS Infrastructure のインストール時に指定します。

OracleAS Infrastructure インストール・タイプを実行すると、「仮想ホストの指定」という画面が表示され、インストール中の OracleAS Infrastructure の仮想ホスト名を入力するテキストボックスが表示されます。詳細は、Oracle Application Server のインストール・ガイドを参照してください。

図 6-8 の例の場合は、本番サイトの OracleAS Infrastructure のインストール時に「仮想ホストの指定」画面が表示されたら、仮想ホスト名 `infra` を入力します。スタンバイ・サイトの OracleAS Infrastructure のインストール時にも、同じ仮想ホスト名を入力します。

---

**注意：** OracleAS Infrastructure を OracleAS Cold Failover Cluster ソリューションにインストールする場合、仮想ホスト名は OracleAS Cold Failover Cluster の仮想 IP に関連付けられた名前と同じになります。

スタンバイ・サイトではなく、本番サイトのみでロード・バランサを使用している複数の OracleAS インスタンスを含む高可用性デプロイでは、仮想ホスト名は `lbr01.us.oracle.com` など、ロード・バランサ用の DNS ベースの仮想ホスト名になります。詳細は、Oracle Technology Network (OTN) に掲載されている、ロード・バランサの使用と OracleAS の高可用性に関するホワイト・ペーパーを参照してください。

---

### 6.2.1.4 仮想ホストの別名

Oracle RAC データベースを含む環境に Disaster Recovery ソリューションを設定する場合、プライマリ・サイトとスタンバイ・サイト間で移行する仮想ホスト名は、Oracle RAC のインスタンスが稼働する各ノードの Windows および UNIX プラットフォームのホスト・ファイルで定義する必要があります。エントリの構文は次のとおりです。

```
<IP>          <HOSTNAME WITH DOMAIN> <HOSTNAME> <VIRTUALHOSTNAME>
```

ここで <VIRTUALHOSTNAME> は別名を示すので、たとえば `racdb1.oracle.com` は RAC Node1 の別名、`racdb2.oracle.com` は RAC Node2 の別名などのように指定します。同じ別名をスタンバイ・サイトにも同様に作成する必要があります。

## 6.2.2 ホスト名解決の構成

OracleAS Disaster Recovery ソリューションでは、第 6.2.1 項「ホスト名の計画と割当て」で計画して割り当てたホスト名の解決に、次の 2 つのいずれかのホスト名解決方法を構成できます。それらのコンポーネントには次のものがあります。

- 第 6.2.2.1 項「ローカル・ホストネーミング・ファイル解決の使用」
- 第 6.2.2.2 項「DNS 解決の使用」

UNIX では、ファイル `/etc/nsswitch.conf` の `hosts` パラメータを使用して、名前解決の方法の順序を指定できます。次に、`hosts` エントリの例を示します。

```
hosts:      files dns nis
```

この文では、ローカル・ホスト名ファイル解決が DNS 解決および NIS (Network Information Service) 解決よりも優先されています。ホスト名の IP アドレスへの解決が必要になったら、`/etc/hosts` ファイル (UNIX の場合) または `C:\WINDOWS\system32\drivers\etc\hosts` ファイルが最初に参照されます。ローカル・ホスト名ファイル解決でホスト名を解決できなかったときは、DNS が使用されます (NIS 解決は、OracleAS Disaster Recovery ソリューションでは使用されません)。ファイル `/etc/nsswitch.conf` を使用した名前解決の詳細は、UNIX システムのドキュメントを参照してください。

### 6.2.2.1 ローカル・ホストネーミング・ファイル解決の使用

ホスト名解決のこの方法は、必要なホスト名とその IP アドレスの対応付けが記載されたローカル・ホストネーミング・ファイルに依存します。UNIX では、このファイルは `/etc/hosts` です。Windows では、このファイルは `C:\WINDOWS\system32\drivers\etc\hosts` です。

UNIX で OracleAS Disaster Recovery ソリューションのホスト名解決にローカル・ホストネーミング・ファイルを使用するには、本番サイトとスタンバイ・サイト両方の中間層および OracleAS Infrastructure の各ホストで、次の手順を実行します。

1. `vi` などのテキスト・エディタを使用して、`/etc/nsswitch.conf` ファイルを編集します。`hosts:` パラメータで、ホスト名解決の第一の選択肢として `files` を指定します。
2. 次のエントリが含まれるように、`/etc/hosts` ファイルを編集します。
  - 現行サイト内のすべての中間層ノードの物理ホスト名とその適切な IP アドレス。最初のエントリは、現行ノードのホスト名と IP アドレスにする必要があります。

---

**注意：** ホスト・ファイルにエントリを追加する場合は、計画したホスト名をホスト・ファイルの 2 番目の列に指定してください。そうしないと、`asgctl` の `verify topology with <host>` の操作に失敗し、本番トポロジがスタンバイ・トポロジと対称ではないというエラー・メッセージが表示されます。この種の問題のトラブルシューティングと解決方法は、付録 A 「高可用性のトラブルシューティング」を参照してください。

---

たとえば、本番サイト内にある中間層ノードの `/etc/hosts` ファイルを編集している場合は、現行ホストのエントリをリストの最初にして、本番サイト内のすべての中間層ノードの物理ホスト名とその IP アドレスを入力します (短縮されたホスト名に加えて、完全修飾されたホスト名も含める必要があります。表 6-3 を参照)。

- 現行サイト内の OracleAS Infrastructure の仮想ホスト名。
 

たとえば、スタンバイ・サイト内にある中間層ノードの `/etc/hosts` ファイルを編集している場合は、スタンバイ・サイト内の OracleAS Infrastructure ホストの仮想ホスト名 (完全修飾名と短縮名の両方) とその IP アドレスを入力します。
3. 前述の手順で説明したファイルを編集した後、各ホストを再起動します。

4. 各ホストから、その特定のサイト内で有効な各物理ホスト名に対して ping コマンドを使用して、IP アドレスが適切に割り当てられていることを確認します。

図 6-8 の例の asmid1 ホストの場合は、次のコマンドを続けて使用します。

```
ping asmid1
```

返された IP アドレスは 123.1.2.333 のはずです。

```
ping asmid2
```

返された IP アドレスは 123.1.2.334 のはずです。

```
ping infra
```

返された IP アドレスは 123.1.2.111 のはずです。

---

**注意：** Solaris などの一部の UNIX 系 OS で IP アドレスを取得するには、`-s` オプションが必要です。

---

Windows では、ホスト名解決の順序を指定する方法は、Windows のバージョンによって異なります。適切な手順については、ご使用のバージョンの Windows のドキュメントを参照してください。

表 6-3 に、図 6-8 の例における、各本番ノードの `/etc/hosts` ファイルのエントリを示します。これには、各 UNIX ホストで必須のエントリが含まれています。Windows の `C:\WINDOWS\system32\drivers\etc\hosts` ファイルでも、必要なエントリは同様です。

**表 6-3 図 6-8 の例の各 `/etc/hosts` ファイルにあるネットワーク・ホスト名と仮想ホスト名のエントリ**

ホスト	<code>/etc/hosts</code> のエントリ
本番サイトの asmid1	123.1.2.333 asmid1.oracle.com asmid1 123.1.2.334 asmid2.oracle.com asmid2 123.1.2.111 infra.oracle.com infra 213.2.2.210 remoteinfra.oracle.com remoteinfra
本番サイトの asmid2	123.1.2.334 asmid2.oracle.com asmid2 123.1.2.333 asmid1.oracle.com asmid1 123.1.2.111 infra.oracle.com infra 213.2.2.210 remoteinfra.oracle.com remoteinfra
本番サイトの infra	123.1.2.111 infra.oracle.com infra 123.1.2.333 asmid1.oracle.com asmid1 123.1.2.334 asmid2.oracle.com asmid2 213.2.2.210 remoteinfra.oracle.com remoteinfra
スタンバイ・サイトの asmid1	213.2.2.443 asmid1.oracle.com asmid1 213.2.2.444 asmid2.oracle.com asmid2 213.2.2.210 infra.oracle.com infra 123.1.2.111 remoteinfra.oracle.com remoteinfra
スタンバイ・サイトの asmid2	213.2.2.444 asmid2.oracle.com asmid2 213.2.2.443 asmid1.oracle.com asmid1 213.2.2.210 infra.oracle.com infra 123.1.2.111 remoteinfra.oracle.com remoteinfra
スタンバイ・サイトの infra	213.2.2.210 infra.oracle.com infra 213.2.2.443 asmid1.oracle.com asmid1 213.2.2.444 asmid2.oracle.com asmid2 123.1.2.111 remoteinfra.oracle.com remoteinfra

### 6.2.2.2 DNS 解決の使用

DNS ホスト名解決を使用するよう OracleAS Disaster Recovery ソリューションを設定するには、企業全体の DNS サーバーに加えて、サイト固有の DNS サーバーを本番サイトとスタンバイ・サイトに設定する必要があります（企業ネットワークには、冗長性を実現するために複数の DNS サーバーがあるのが一般的です）。図 6-9 に、この設定の概要を示します。

**関連項目：** UNIX 環境で DNS サーバーを設定する方法は、第 10 章「DNS サーバーの設定」を参照してください。

図 6-9 DNS 解決トポロジの概要

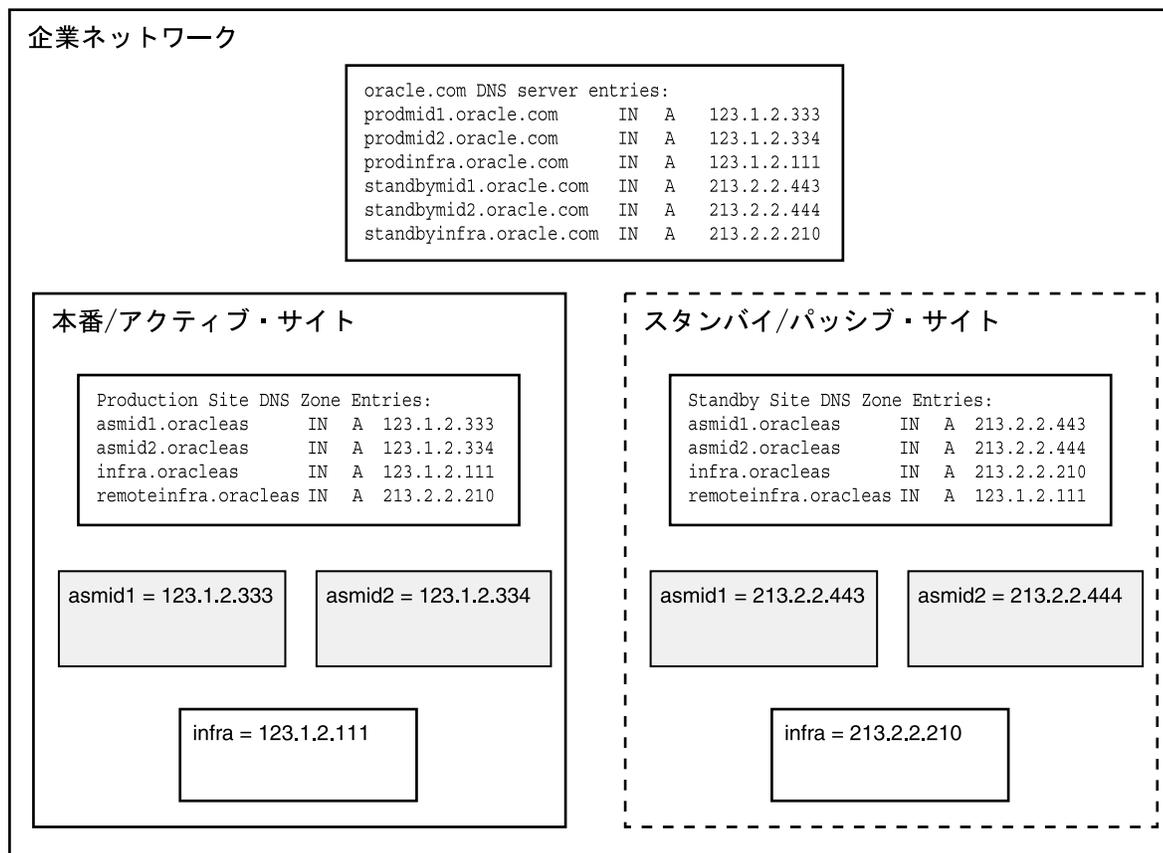


図 6-9 のトポロジが機能するには、次のことが要件または前提となります。

- 本番サイトとスタンバイ・サイトの DNS サーバーが互いを認識しないこと。サイトの DNS サーバーがサイト内のホスト名を解決できない場合は、企業全体の DNS サーバーに対して、non authoritative 検索リクエストが作成されます。
- 本番サイトとスタンバイ・サイトの DNS サーバーに、中間層の物理ホスト名と OracleAS Infrastructure の仮想ホスト名のエントリがあること。各 DNS サーバーには、担当サイト内のホスト名のみエントリが含まれます。両サイトには、企業全体のドメイン名とは異なる共通のドメイン名があります。
- 企業全体の DNS サーバーに、本番サイトとスタンバイ・サイト両方の中間層ホストと OracleAS Infrastructure ホストのネットワーク・ホスト名のエントリがあること。
- UNIX の場合、各ホストの /etc/hosts ファイルに、本番とスタンバイの両方のサイトのどのホストについても、物理ホスト名、ネットワーク・ホスト名または仮想ホスト名のエントリがないこと。Windows では、この要件はファイル C:\WINDOWS\system32\drivers\etc\hosts に適用されます。

DNS 解決を使用する OracleAS Disaster Recovery ソリューションを設定する手順は次のとおりです。

1. 企業全体の DNS サーバーのそれぞれを、本番サイトとスタンバイ・サイトにあるすべてのホストのネットワーク・ホスト名で構成します。図 6-8 の例では、次のエントリを作成します。

```
prodmid1.oracle.com      IN    A    123.1.2.333
prodmid2.oracle.com      IN    A    123.1.2.334
prodinfra.oracle.com     IN    A    123.1.2.111
standbymid1.oracle.com   IN    A    213.2.2.443
standbymid2.oracle.com   IN    A    213.2.2.444
standbyinfra.oracle.com  IN    A    213.2.2.210
```

2. 本番およびスタンバイの各サイトで、DNS サーバーを次の手順で構成して、一意の DNS ゾーンを作成します。
  - a. 企業ドメイン名とは異なる、2つのサイト用の一意のドメイン名を選択します。この例では、図 6-8 の両サイトのドメイン名として、名前 `oracleas` を使用します。上位レベルの企業ドメイン名は `oracle.com` です。
  - b. リクエストを解決できない場合は、企業全体の DNS サーバーを指し示すように各サイトの DNS サーバーを構成します。
  - c. 各サイトの DNS サーバーに、各中間層ホストの物理ホスト名と各 OracleAS Infrastructure ホストの仮想ホスト名を追加します。この名前には、前の手順で選択したドメイン名も含めます。

図 6-8 の例のエントリは、次のようになります。

本番サイトの DNS サーバーの場合：

```
asmid1.oracleas         IN    A    123.1.2.333
asmid2.oracleas         IN    A    123.1.2.334
infra.oracleas          IN    A    123.1.2.111
```

スタンバイ・サイトの DNS サーバーの場合：

```
asmid1.oracleas         IN    A    213.2.2.443
asmid2.oracleas         IN    A    213.2.2.444
infra.oracleas          IN    A    213.2.2.210
```

---

**注意：** ロード・バランサを使用している場合は、DNS ベースの仮想ホスト名を使用して、ホスト・ファイルに IP アドレスの別名を指定する必要があります。これは、ローカル・ホストの OracleAS Guard に対し、`discover topology within farm` コマンドを使用してトポロジ・ファイルのローカル書込みを実行し、`add instance` コマンドを正しく実行してトポロジ・ファイルを更新するために必要です。

これらのサイトいずれかで、OracleAS Infrastructure に OracleAS Cold Failover Cluster ソリューションを使用する場合は、クラスタの仮想ホスト名と仮想 IP アドレスを入力します。たとえば、前の手順では `infra` が本番サイトのクラスタの仮想ホスト名で、`123.1.2.111` が仮想 IP アドレスになります。OracleAS Cold Failover Cluster ソリューションの詳細は、OracleAS 10g リリース 2 (10.1.2.0.2) ドキュメント・セットに含まれている『Oracle Application Server 高可用性ガイド』の第 6.2.2 項「アクティブ / パッシブ型の高可用性トポロジ」を参照してください。

---

### 6.2.2.2.1 DNS サーバーへの Oracle Data Guard 用エントリの追加

OracleAS Guard は、Oracle Data Guard テクノロジの使用を自動化して、本番とスタンバイの OracleAS Infrastructure データベースを同期化するため、本番 OracleAS Infrastructure はスタンバイ OracleAS Infrastructure を参照できる必要があります、その逆も同様です。

これを機能させるには、本番サイトの DNS サーバーに、スタンバイ OracleAS Infrastructure ホストの IP アドレスを、本番サイトに対して一意のホスト名とともに入力する必要があります。同様に、スタンバイ・サイトの DNS サーバーに、本番 OracleAS Infrastructure ホストの IP アドレスを、同じホスト名を使用して入力する必要があります。これらの DNS エントリが必要なのは、Oracle Data Guard では、本番およびスタンバイ OracleAS Infrastructure へのリクエストの転送に TNS 名が使用されるためです。したがって、tnsnames.ora ファイルにも適切なエントリを作成する必要があります。また、OracleAS Guard の asgctl コマンドライン・コマンドにも、ネットワーク・ホスト名を使用する必要があります。

図 6-8 の例では、リモート OracleAS Infrastructure 用に選択した名前が remoteinfra であるとする、本番サイトの DNS サーバーのエントリは次のようになります。

```
asmid1.oracleas      IN    A    123.1.2.333
asmid2.oracleas      IN    A    123.1.2.334
infra.oracleas       IN    A    123.1.2.111
remoteinfra.oracleas IN    A    213.2.2.210
```

同様に、スタンバイ・サイトの DNS サーバーのエントリは次のようになります。

```
asmid1.oracleas      IN    A    213.2.2.443
asmid2.oracleas      IN    A    213.2.2.444
infra.oracleas       IN    A    213.2.2.210
remoteinfra.oracleas IN    A    123.1.2.111
```

## 6.3 Oracle Application Server のインストールの概要

この項では、OracleAS Disaster Recovery ソリューションをインストールする手順の概要を示します。これらの手順は、第 6.1.3 項「サポートされているトポロジ」に示したトポロジに適用できます。第 6.2 項「OracleAS Disaster Recovery 環境の準備」の手順に従ってソリューション環境を設定したら、この項でインストール・プロセスの概要をお読みください。その後、Oracle Application Server のインストール・ガイドの詳細な手順に従って、ソリューションをインストールします。

---

**注意：** 本番サイトとスタンバイ・サイトの対称ホストが使用するようにそれぞれに同一のポートを割り当てるには、静的ポートの定義を使用します。この定義は、ファイル (staticports.ini など) で定義されます。インストーラの「ポート構成オプションの指定」画面で staticports.ini ファイルを指定します (静的ポート・ファイルの詳細は、Oracle Application Server のインストール・ガイドを参照)。

---

OracleAS Disaster Recovery ソリューションをインストールするための一般的な手順は次のとおりです。

1. 本番サイトに OracleAS Infrastructure をインストールします (Oracle Application Server のインストール・ガイドを参照)。
2. スタンバイ・サイトに OracleAS Infrastructure をインストールします (Oracle Application Server のインストール・ガイドを参照)。
3. 中間層をインストールする前に、各サイトで OracleAS Infrastructure を起動します。
4. 本番サイトに中間層をインストールします (Oracle Application Server のインストール・ガイドを参照)。
5. スタンバイ・サイトに中間層をインストールします (Oracle Application Server のインストール・ガイドを参照)。

インストールを実行するときは、次の点が重要です。

- 両サイトの同等のホスト間で同じポートが使用されていることを確認します。たとえば、スタンバイ・サイトの `asmid1` ホストは、本番サイトの `asmid1` ホストと同じポートを使用する必要があります。静的ポート定義ファイルを使用します（この項の前述の注意と次の点を参照）。
- インストーラの「**ポート構成オプションの指定**」画面で、`staticports.ini` ファイルへのフル・パスを指定します。
- インストーラの「**構成オプションの選択**」画面で、「高可用性およびレプリケーション」オプションを選択していることを確認します。
- OracleAS Infrastructure のインストール中に「**仮想ホストの指定**」画面で OracleAS Infrastructure に割り当てる仮想アドレスを指定します。
- 中間層ホストの場合は、使用可能な中間層インストール・タイプのいずれかをインストールします（サイトに中間層をインストールする前に、そのサイトで OracleAS Infrastructure サービスが起動されていることを確認します）。
- 中間層の各インストールでは、OracleAS Infrastructure データベースとして OracleAS Infrastructure の仮想ホスト名を指定します。
- 各サイトのホストで OracleAS サービスを起動します（OracleAS Infrastructure のサービスを最初に起動します）。

## 6.4 OracleAS Guard と asgctl の概要

この項では、OracleAS Guard とそのコマンドライン・インタフェースである `asgctl` の概要について説明します。この概要についてすでに理解している場合は、[第 6.5 項「データベースの認証」](#)に進んでください。この項の項目は次のとおりです。

- [第 6.4.1 項「asgctl の概要」](#)
- [第 6.4.2 項「OracleAS Guard クライアント」](#)
- [第 6.4.3 項「OracleAS Guard サーバー」](#)
- [第 6.4.4 項「asgctl 操作」](#)
- [第 6.4.5 項「OracleAS Guard と OPMN との統合」](#)
- [第 6.4.6 項「サポートされる OracleAS Disaster Recovery 構成」](#)
- [第 6.4.7 項「OracleAS Guard の構成とその他の関連情報」](#)

### 6.4.1 asgctl の概要

`asgctl` コマンドライン・ユーティリティによって、OracleAS Disaster Recovery の設定および管理に関する複雑で長い手順が大幅に簡素化されました。このユーティリティは、クライアント・コンポーネントとサーバー・コンポーネントで構成される分散ソリューションを提供します。クライアント・コンポーネント（OracleAS Guard クライアント）は、トポロジ上のシステムにインストールできます。サーバー・コンポーネント（OracleAS Guard サーバー）は、デフォルトでは、OracleAS Disaster Recovery 環境を構成するプライマリおよびスタンバイ Oracle ホームをホスティングするシステムにインストールされます。

### 6.4.2 OracleAS Guard クライアント

OracleAS Guard クライアントは、すべての OracleAS インストール・タイプでインストールされます。OracleAS Guard クライアントは、OracleAS Guard サーバーとの接続の確立と維持を試みます。

OracleAS Guard クライアントには、`asgctl` コマンドライン・インタフェース（CLI）が用意されており（[第 7 章「OracleAS Guard asgctl コマンドライン・リファレンス」](#)を参照）、[第 6.4.4 項「asgctl 操作」](#)で説明する管理タスクを実行する一連のコマンドが含まれています。

### 6.4.3 OracleAS Guard サーバー

OracleAS Guard サーバーは、分散サーバーとして（デフォルトでインストールされ）、OracleAS Disaster Recovery 構成のすべてのシステムで実行されます。OracleAS Guard クライアントは、OracleAS Disaster Recovery 構成とネットワーク接続されている 1 つのシステムの OracleAS Guard サーバーとアクティブな接続を維持します。この調整サーバーは、必要に応じて、OracleAS Disaster Recovery 構成内の他のシステムの OracleAS Guard サーバーと通信し、スタンバイ・サイトのクローニング、インスタンス化、同期化、検証、スイッチオーバーおよびフェイルオーバー操作の処理を実行します。OracleAS Guard サーバーは、OracleAS Guard クライアントから直接発行された asgctl コマンドを実行したり、ネットワーク内の別の OracleAS Guard サーバーから OracleAS Guard クライアントにかかわって発行されたコマンドをそのクライアントのセッション中に実行します。操作を実行する手順は、本番トポロジとスタンバイ・トポロジ両方のすべてのシステムで実行されます。ほとんどの操作手順は、OracleAS Disaster Recovery 構成にあるこれらのシステムの OracleAS Guard サーバーで、必要に応じて並列または順番に実行されます。

### 6.4.4 asgctl 操作

asgctl コマンドを使用した主な asgctl 操作は、次のいずれかの操作のカテゴリに分類されます。

- 認証: プライマリ・トポロジの OracleAS Infrastructure データベースを識別します ([set primary database](#) コマンド)。複数の Infrastructure が含まれるトポロジがある場合は、本番トポロジとスタンバイ・トポロジの両方に関する操作を実行する前にこのコマンドを使用してそれぞれを識別する必要があります。

フェイルオーバー操作の前にスタンバイ・トポロジで新しい OracleAS Infrastructure データベースを識別します ([set new primary database](#) コマンド)。

特定のホストに対して、OracleAS Guard クライアントから OracleAS Guard サーバーへの接続、および OracleAS Guard サーバー間の接続を認証するための資格証明を設定します ([set asg credentials](#) コマンド)。例は、「[set asg credentials](#)」コマンドを参照してください。詳細は、[第 6.18.1.1 項「asgctl 資格証明の設定」](#)を参照してください。

OracleAS Guard がトポロジを検出する場合 ([discover topology](#) コマンド)、Oracle Internet Directory に問い合せて、その本番サイトのインスタンス情報を取得するために Oracle Internet Directory 認証資格証明 (Oracle Internet Directory パスワード) を提供する必要があります。

- トポロジの検出: Oracle Internet Directory に問い合せることによって、トポロジ内で本番サイトの同じ Oracle Internet Directory を共有するすべてのインスタンスを検出し ([discover topology](#) コマンド)、そのトポロジを記述するトポロジ XML ファイルを生成して、そのファイルをトポロジ内のすべてのインスタンスにレプリケートします。詳細は、[第 6.6 項「トポロジの検出、ダンプおよび検証」](#)を参照してください。

コマンド [discover topology within farm](#) は、Oracle Internet Directory を使用できない特別な場合に、本番サイトにある OPMN を使用して OracleAS 10.1.3 OPMN トポロジなどのトポロジを検出します。

- ローカル・トポロジ・ファイルでインスタンスを追加または削除すると、既存のローカル・トポロジ・ファイルでインスタンスの追加または削除が行われます。これは、OracleAS 10.1.3 のみのトポロジで特に役立ちます。このトポロジでは、OracleAS Guard クライアントが既存の OracleAS 10.1.3 インスタンスに接続し、指定した OracleAS 10.1.3 インスタンスをトポロジ・ファイルに追加する [add instance](#) コマンドを実行するか、指定した OracleAS 10.1.3 インスタンスをローカル・トポロジ・ファイルから削除する [remove instance](#) コマンドを実行できます。どちらの操作を実行した場合も、指定に応じて更新済トポロジ・ファイルが Disaster Recovery 本番環境のインスタンスすべてに伝播されます。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ環境全体に本番トポロジ・ファイルが自動的に伝播されます。

このコマンドは、OracleAS 10.1.3 J2EE インスタンスを OID ベースの 10.1.2.0.2 ローカル・トポロジ・ファイルに追加して、バージョンが混在した Disaster Recovery 環境をサポートする場合にも役立ちます。たとえば、`add instance` コマンドを使用して、OracleAS 10.1.3 J2EE インスタンスを OID ベースの 10.1.2.0.2 ローカル・トポロジ・ファイルに追加できます。ユースケースについては、第 6.9 項「既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除」を参照してください。

- スタンバイ・サイトのクローニング: 1 つの本番インスタンスを 1 つのスタンバイ・システムにクローニングするか (`clone instance` コマンド)、2 つ以上の本番インスタンスを複数のスタンバイ・システムにクローニングします (`clone topology` コマンド)。詳細は、第 6.10 項「OracleAS Guard 操作: スタンバイ・システムへの 1 つ以上の本番インスタンスのスタンバイ・サイト・クローニング」を参照してください。スタンバイ・サイトのクローニング操作を使用すると、これらの Oracle インスタンスをスタンバイ中間層システムにインストールしてからインスタンス化操作を実行するという作業を行わずに済みます。
- スタンバイ・サイトのインスタンス化: 障害時リカバリ環境を作成します。これによって、スタンバイ・インスタンスと本番インスタンス間の関係が確立され、構成がミラー化され、スタンバイ Infrastructure が作成され、プライマリ・サイトとスタンバイ・サイトが同期化されます (`instantiate topology` コマンド)。詳細は、第 6.11.1 項「スタンバイのインスタンス化」を参照してください。
- スタンバイ・サイトの同期化: OracleAS Infrastructure のデータベースの REDO ログをスタンバイ・サイトに適用するとともに、トポロジ全体で外部構成ファイルを同期化します (`sync topology` コマンド)。詳細は、第 6.11.2 項「スタンバイの同期化」を参照してください。
- スイッチオーバー: スタンバイ・サイトにデータベース REDO ログを適用して本番サイトとの同期化を行った後に、本番サイトからスタンバイ・サイトへ切り替えます (`switchover topology` コマンド)。詳細は、第 6.12.1.1 項「スケジューリングした停止」を参照してください。
- フェイルオーバー: 構成ファイルをリストアし、OracleAS サーバー環境を、最後に成功した同期化操作の時点までリストアした後、スタンバイ・サイトを本番サイトにします (`failover` コマンド)。詳細は、第 6.12.1.2 項「計画外停止」を参照してください。
- 検証: プライマリ・トポロジが実行状態であること、および構成が有効であることを検証します (`verify topology` コマンド)。またはスタンバイ・トポロジを指定した場合は、ローカル・ホスト・システムがメンバーとなっているプライマリ・トポロジと、スタンバイ・トポロジを比較し、両方のトポロジが一致していること、および OracleAS Disaster Recovery の要件を順守していることを検証します。詳細は、第 6.15.1 項「トポロジの検証」を参照してください。
- ポリシー・ファイルの使用: 非対称トポロジをサポートするために、不要なインスタンスを除外するフィルタとして使用します。 `dump policies` コマンドは、選択したセットの `asgctl` コマンドに対して XML フォーマットのそれぞれのファイルに詳細なデフォルト・ポリシー情報を書き込みます。その後、それぞれの XML ポリシー・ファイルを編集し、それを、選択したセットの `asgctl` コマンド (`dump topology`、`verify topology`、`clone topology`、`failover`、`instantiate topology`、`switchover topology` および `sync topology`) のいずれかで `using policy <file>` パラメータで指定し、インスタンス別にこれらの各 `asgctl` コマンドに許可される実行操作のドメインを定義します。XML ポリシー・ファイルの各インスタンスのリストのエントリは、本番とスタンバイのピア組合せと、正常な操作のための成功要件を定義する特定の属性を論理的に結び付けます。たとえば、前述の操作の 1 つを実行するときに、対称トポロジの 1 つのノードを省くことができます。ポリシー・ファイルを使用して、無視するノードを指定します。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。
- インスタンス管理: トポロジを停止し (`shutdown topology` コマンド)、起動できます (`startup topology` コマンド)。

- **トラブルシューティング**: `dump topology` コマンドを使用して、トポロジに関する詳細情報を画面またはファイルに書き込みます。また、現在どの操作が実行されているかを調べ (`show operation` コマンド)、停止する必要がある場合はその操作を停止できます (`stop operation` コマンド)。

表 6-4 は、`asgctl` のクローニング、インスタンス化、同期、フェイルオーバーおよびスイッチオーバー操作の前と後の OracleAS Disaster Recovery 本番サイト環境とスタンバイ・サイト環境を示しています。

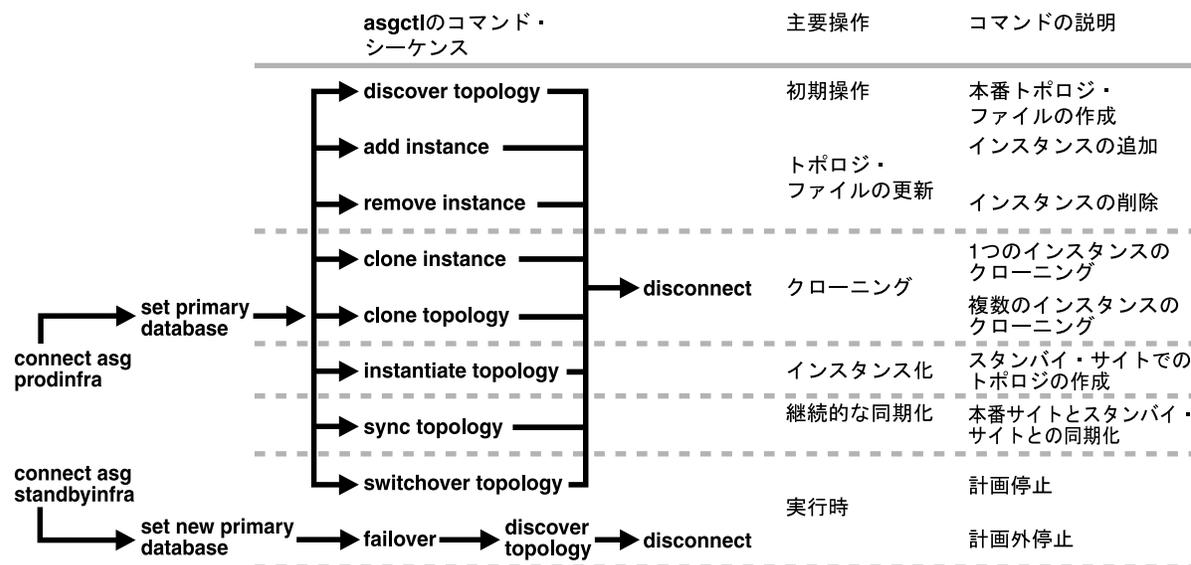
**表 6-4 OracleAS Guard 操作の実行前と後の Disaster Recovery の本番環境およびスタンバイ環境の説明**

OracleAS Guard 操作	操作前の DR サイトの環境	操作後の DR サイトの環境
クローニング	本番サイトには、スタンバイ・サイトにインストールしてインスタンス化する必要がある 1 つ以上のインスタンスがあります。クローニング操作がこのタスクを実行します。	スタンバイ・サイトには、本番サイトのインスタンスの論理ミラーである 1 つ以上の新しいスタンバイ・インスタンスがあります。
インスタンス化	それ自体の Oracle ホームを持つスタンバイ・サイトは存在するが、OracleAS Disaster Recovery 操作を実行する対象となる、サイト間の OracleAS Disaster Recovery の関係は存在していません。	スタンバイ・サイトには、本番サイトの論理ミラーが設定および維持されています。
同期化	スタンバイ・サイトは本番サイトと一致していません。OracleAS Disaster Recovery は、手動による操作を行わないと、一貫性のあった時点でさかのぼってスタンバイ・サイトをリストアできません。	データベースの REDO ログを OracleAS Infrastructure に適用するとともに、トポロジ全体で外部構成ファイルを同期化します。同期化操作は、フェイルオーバーやスイッチオーバー操作が必要なイベントで実行され、スタンバイ・サイトを一貫性のあった時点までリストアできます。 <code>asgctl</code> の同期化操作を実行した後は、サイトを同期化するための手動操作は必要ありません。
スイッチオーバー	本番サイトを計画停止すると、スタンバイ・サイトが一定期間、本番サイトになります。つまり、各サイトの役割が入れ替わります。	スタンバイ・サイトが本番サイトになりました。すべての OPMN サービスが起動します。本番サイトは、計画停止の後、再び使用可能になります。その時点で、もう一度スイッチオーバー操作を実行し、スタンバイ・サイトから元の本番サイトにアクティビティを戻すことができます。
フェイルオーバー	本番サイトでの計画外停止によって、本番サイトが、不特定の期間、停止または使用不能になりました。本番サイトは、予期せぬ状況によって失われています。	スタンバイ・サイトは、永続的に本番サイトになりました。構成および Infrastructure データは、スタンバイ・サイトで一貫性のある時点までリストアされます。最後に同期化操作を行った時点から、一貫性のある方法でサイトのサービスが開始されます。すべての OPMN サービスが起動します。

図 6-10 では、OracleAS Disaster Recovery の主要操作とこれらの操作に使用されるコマンド・シーケンスを要約しています。たとえば、新たに設定され、稼動し始めた Disaster Recovery 環境では、本番サイトに必ずトポロジ・ファイルを作成する必要があります。このファイルを作成するには、connect コマンドを使用して OracleAS Guard クライアントを OracleAS Guard サーバに接続し、set primary database コマンドを使用して本番 Infrastructure データベースを識別し、discover topology コマンドを実行し、最後に disconnect コマンドを実行して OracleAS クライアントを OracleAS サーバから切断します。本質的に、asgctl コマンドを使用して OracleAS Disaster Recovery 操作を実行するには、必ず OracleAS Guard クライアントを OracleAS Guard サーバに接続し、Infrastructure データベースの場所を識別し、目的の OracleAS Disaster Recovery 操作を実行し、最後に OracleAS クライアントを OracleAS サーバから切断する必要があります。この一般的なシーケンスにおける唯一の例外は、フェイルオーバー操作を実行する場合です。この場合、本番サイトは計画外の問題が原因で、永久に使用不可になっています。この場合には、OracleAS Guard クライアントをスタンバイ・サイトの OracleAS Guard サーバに接続し、スタンバイ・サイトの Infrastructure データベースを新しい Infrastructure データベースとして識別し、フェイルオーバー操作を実行します。スタンバイ・サイトではフェイルオーバー操作後にトポロジ・ファイルが作成されないため、このファイルを初めて作成するには、discover topology コマンドを実行する必要があります。ファイルを作成した後は、OracleAS Guard クライアントを OracleAS Guard サーバから切断できます。

図 6-10 に示す asgctl コマンド・シーケンスでは、最も単純なトポロジ構成を想定しています。たとえば、より複雑なケースでは、本番サイトまたはスタンバイ・サイトに複数の MR インスタンスがインストールされている場合、インスタンス化、同期化、スイッチオーバー、フェイルオーバー操作などの Disaster Recovery の主要操作を実行する前に、set primary database コマンドを使用して各インスタンスを識別する必要があります。同様に、インスタンス化、同期化、スイッチオーバー、フェイルオーバーなどの OracleAS Guard の主要操作を実行する前に、接続している OracleAS Guard サーバと異なる資格証明を持つトポロジの OracleAS Guard サーバ・システムに対して、OracleAS Guard で資格証明を設定する必要があります。このため、これらのどちらのケースに対しても、Disaster Recovery の主要操作を実行するには、コマンド・シーケンスで追加の asgctl コマンドが必要になります。詳細は、set asg credentials コマンドの使用方法を参照してください。

図 6-10 次の OracleAS Guard asgctl コマンド・シーケンスを使用して実行される Disaster Recovery の主要操作



## 6.4.5 OracleAS Guard と OPMN との統合

一般的な Oracle Application Server サイトには、複数のファームがあります。OracleAS Guard サーバーとその ias-component ASG プロセスは、障害時リカバリ・サイトのコンテキストでのみ必要であるため、デフォルトでは OPMN によって起動されません。この項で後述するように、この ias-component ASG プロセスをすべての Oracle ホームで起動する必要があります。このコンポーネントのステータスをチェックし、コンポーネントが実行中であるかどうかを調べるには、トポロジ内の各システムで次の `opmnctl` コマンドを実行します。

```
On UNIX systems
> <ORACLE_HOME>/opmn/bin/opmnctl status
```

```
On windows systems
> <ORACLE_HOME>%opmn%bin%opmnctl status
```

本番サイトの OracleAS Guard クライアントと OPMN は、スタンバイ・サイトの OracleAS Guard サービスを起動できないため、OracleAS Guard はスタンバイ・トポロジの Infrastructure ノードで `opmnctl` を使用して直接起動する必要があります。ノードに接続し、UNIX システム上で次の OPMN コマンドを実行します。

```
> <ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=ASG
```

Windows システムでは、次の OPMN コマンドを発行して OracleAS Guard を起動します (Oracle ホームがドライブ C: にある場合)。

```
C:%<ORACLE_HOME>%opmn%bin%opmnctl startproc ias-component=ASG
```

OracleAS Guard サーバーを起動した後、そのサーバーは一時サーバーではなくなり、スタンバイ・トポロジ内の他の OracleAS Guard サーバーが一時サーバーになります。この構成によって、トポロジ間の通信が実現されます。

---

**注意：** OracleAS Guard が実行されているシステムで `opmnctl status` コマンドを実行するときに、`ias-component` と `process-type` には ASG が指定されません。これは、OracleAS Guard サーバーの OracleAS コンポーネント名とサーバー・プロセス名を示しています。

---

## 6.4.6 サポートされる OracleAS Disaster Recovery 構成

OracleAS 10g リリース 2 (10.1.2) の場合、OracleAS Guard は、Oracle Application Server Cold Failover Cluster と単一インスタンスのデフォルトの OracleAS Infrastructure 構成だけでなく、第 6.1.3 項「サポートされているトポロジ」に示すトポロジもサポートします。

## 6.4.7 OracleAS Guard の構成とその他の関連情報

デフォルトでは、OracleAS Guard と、OracleAS Guard のコマンドライン・ユーティリティである `asgctl` は、次のデフォルトの構成情報ですべてのインストール・タイプにインストールされます。

- 構成可能な OracleAS Guard パラメータは次のとおりです。ここには値の説明とデフォルト値が示されています。構成可能な OracleAS Guard パラメータは、`<ORACLE_HOME>%dsa%doc` ディレクトリにある OracleAS Guard `readme.txt` ファイルにもリストされています。
  - `port`: OracleAS Guard サーバーおよびクライアント用 TCP/IP ポート。OracleAS Guard では、たとえば `port=7890` などのように、7890 のデフォルトのポート (port) 番号が使用されます。システムに 2 番目の Oracle ホームがインストールされている場合、この 2 番目の Oracle ホームには異なる OracleAS Guard ポート番号が必要です。通常は、`port=7891` などのように、番号が 1 つ増分されます。

値: 整数。有効な任意の TCP/IP ポート番号を表します。デフォルトは 7890 です。

- `exec_timeout_secs`: オペレーティング・システム・コマンド実行用のタイムアウト値。  
値: 整数。秒数を表します。デフォルトは 60 秒です。
- `trace_flags`: オンにするトレース・フラグ。  
値: 文字列リスト。"," で区切ります。デフォルトは未指定です。
- `backup_mode`: 全体バックアップまたは増分バックアップのどちらかを実行するかを示します。  
値: 文字列。"full" または "incremental" のどちらかです。デフォルトは "incremental" です。
- `backup_path`: OracleAS Guard サーバーが使用するバックアップ・ディレクトリ・パス。  
値: 文字列。ディレクトリ・パスを表します。デフォルトは <ORACLE\_HOME>/dsa/backup です。
- `ha_path`: バックアップ・スクリプトを含む高可用性ディレクトリ・パス。  
値: 文字列。ディレクトリ・パスを表します。デフォルトは <ORACLE\_HOME>/backup\_restore です。
- `port.<host>`: 任意のホストの TCP/IP ポート。  
値: 整数。有効な任意の TCP/IP ポート番号を表します。

---

**注意:** ポート番号を変更する必要がある場合は (ポート番号は、マシン上の各 Oracle ホームの各 OracleAS Guard サーバーに対して一意である必要があり、インストール時に自動的に割り当てられます)、その値を <ORACLE\_HOME>/dsa/dsa.conf ファイルで変更することができます。変更したら、OracleAS Guard サーバーを停止してから (<ORACLE\_HOME>/opmn/bin/opmnctl stopproc ias-component=ASG)、OracleAS Guard サーバーを再起動し (<ORACLE\_HOME>/opmn/bin/opmnctl startproc ias-component=ASG)、変更内容を有効にします。詳細は、[第 6.4.5 項「OracleAS Guard と OPMN との統合」](#)を参照してください。

トポロジ内でポート番号を変更する必要がある場合は、`asgctl` の `shutdown` コマンドを使用して OracleAS Guard サーバーを停止した後、`asgctl` の `start` コマンドを使用して OracleAS Guard サーバーを起動します。これらの停止と起動操作は、Oracle AS Guard トポロジの一部である各 Oracle ホームで実行する必要があります。

---

- `copyfile_buffersize`: ファイルのコピー操作バッファ・サイズ (キロバイト単位)。  
値: 整数。最大バッファ・サイズは 500K です。
- `server_inactive_timeout`: サーバーが非アクティブのときに、停止する前に待機する秒数。  
値: 整数。秒数を表します。デフォルト値は 600 秒 (10 分) です。
- `inventory_location`: Oracle Inventory の代替格納場所。  
値: 文字列。oraInst.loc ファイルのフルパスを指定します。
- `dsa_realm_override`: dsa レルムの上書きを指定します。  
値: 整数。1= 上書き、0= 上書きなし

- OracleAS Guard のコマンドライン・ユーティリティである `asgctl` は、本番トポロジとスタンバイ・トポロジのすべてのノードで、UNIX システムの場合は `<ORACLE_HOME>/dsa/bin` ディレクトリに、Windows システムの場合は、`<ORACLE_HOME>%dsa%bin` ディレクトリにインストールされます。
- OracleAS Guard は、本番トポロジ全体の OracleAS コンポーネント・サービスを起動します。
- OracleAS Guard 操作によるトポロジのステータス情報は (`asgctl show operation full` または `show operation history` コマンド)、現在の OracleAS Guard クライアントの `asgctl connect` セッションが有効であるかぎり、利用することができます。OracleAS Guard クライアントが OracleAS Guard サーバーとの接続を切断した場合は、そのトポロジの操作の履歴情報 (`operation history`) は利用できなくなります。
- `asgctl` 操作を開始した後は、実行されているコマンドが完了するか、強制終了されるまで (詳細は、`asgctl` 「[stop operation](#)」コマンドを参照)、同じ OracleAS Guard サーバーで次の `asgctl` コマンドを実行できません。また、`asgctl` 操作をバックグラウンドで実行している間に、`asgctl` ユーティリティを終了 (`quit` または `exit` コマンドを実行) することはできません。
- 構成可能な OracleAS Guard パラメータの詳細は、`<ORACLE_HOME>%dsa%doc` ディレクトリにある OracleAS Guard `readme.txt` ファイルを参照してください。
- OracleAS Guard のコマンドライン・ユーティリティである `asgctl` は、本番トポロジとスタンバイ・トポロジのすべてのノードで、UNIX システムの場合は `<ORACLE_HOME>/dsa/bin` ディレクトリに、Windows システムの場合は、`<ORACLE_HOME>%dsa%bin` ディレクトリにインストールされます。
- OracleAS Guard は、本番トポロジ全体の OracleAS コンポーネント・サービスを起動します。
- OracleAS Guard 操作によるトポロジのステータス情報は (`asgctl show operation full` または `show operation history` コマンド)、現在の OracleAS Guard クライアントの `asgctl connect` セッションが有効であるかぎり、利用することができます。OracleAS Guard クライアントが OracleAS Guard サーバーとの接続を切断した場合は、そのトポロジの操作の履歴情報 (`operation history`) は利用できなくなります。
- `asgctl` 操作を開始した後は、実行されているコマンドが完了するか、強制終了されるまで (詳細は、`asgctl` 「[stop operation](#)」コマンドを参照)、同じ OracleAS Guard サーバーで次の `asgctl` コマンドを実行できません。また、`asgctl` 操作をバックグラウンドで実行している間に、`asgctl` ユーティリティを終了 (`quit` または `exit` コマンドを実行) することはできません。

## 6.5 データベースの認証

OracleAS Guard クライアントが OracleAS Guard サーバーに接続し、本番トポロジ内または本番トポロジとスタンバイ・トポロジの両方で管理操作を実行するセッションを開始する場合は、次のようないくつかのレベルの認証が必要です。

- Infrastructure の認証
- OracleAS Guard サーバーに対する OracleAS Guard クライアントの認証
- Oracle Internet Directory の認証

### Infrastructure の認証

OracleAS Guard 管理セッションをインスタンス化する場合、OracleAS Guard クライアントと OracleAS Guard サーバーとの間の接続を確立した後、`set primary database` コマンドを使用してプライマリ・トポロジのすべての OracleAS Infrastructure データベースを識別する必要があります。Infrastructure の認証は、本番トポロジまたは本番トポロジとスタンバイ・トポロジの両方に関連する操作を開始する前に実行する必要があります。

別の形式の Infrastructure の認証は、フェイルオーバー操作の一部として実行されます。このシナリオでは、本番サイトは停止しており、スタンバイ・サイトにフェイルオーバーしてこのサイトを新しい本番サイトとする必要があります。最初に、フェイルオーバー操作を実行する前に `set new primary database` コマンドを使用してスタンバイ・トポロジの新しい OracleAS Infrastructure データベースを識別します。詳細は、第 6.12.1.2 項「計画外停止」を参照してください。

### OracleAS Guard サーバーに対する OracleAS Guard クライアントの認証

デフォルトでは、これらは、Oracle Application Server のインストール中に作成され、`connect asg` コマンドで使用された Oracle Application Server アカウント (`ias_admin/password`) でのインスタンス・レベルの認証に使用されるものと同じ認証資格証明です

---

**注意：** OracleAS 10.1.3 インストールの場合、ユーザー名は `oc4jadmin` を使用し、この `oc4jadmin` アカウントのパスワードは Oracle Application Server 10.1.3 のインストール時に作成されたものを使用する必要があります。

---

これらの同じ資格証明は、管理操作を実行する場合に本番トポロジとスタンバイ・トポロジの OracleAS Guard サーバーに OracleAS Guard クライアントが接続するときに使用されます。

特定の OracleAS Guard サーバーに対して異なる資格証明を使用したり、プライマリ・トポロジで使用する資格証明とは異なる一般的なセットの資格証明をスタンバイ・トポロジに設定する必要がある場合があります。OracleAS Guard サーバーに対して資格証明を設定するには、`set asg credentials` コマンドと 1 つ以上のそのパラメータ・オプションを使用し、資格証明を適用するホスト名またはトポロジを新しいセットの資格証明 (ユーザー名 / パスワード) とともに指定します。

あるトポロジに資格証明を設定した場合、その資格証明はトポロジ全体に継承されます。トポロジ上の個別のホストに資格証明を設定した場合、そのホストの資格証明が、トポロジに設定されたデフォルトの資格証明よりも優先されます。ホスト・システムまたはトポロジ全体にデフォルトの接続資格証明とは異なる資格証明を設定すると、OracleAS Guard 管理セッションを開始するたびに、本番トポロジ内または本番トポロジとスタンバイ・トポロジの両方のすべての OracleAS Guard サーバーに関係する操作を実行する前に、ホスト・システムまたはトポロジのデフォルト接続情報とは異なるすべての資格証明を指定する必要があります。指定しない場合、操作は認証エラーのために失敗します。例は、「`connect asg`」コマンドを参照してください。

### Oracle Internet Directory の認証

`discover topology` コマンドでは、Oracle Internet Directory に問い合せて、その本番サイトのインスタンス情報を取得するために Oracle Internet Directory 認証資格証明 (Oracle Internet Directory パスワード) を提供する必要があります。詳細および「`discover topology`」コマンドは、後述のセクションを参照してください。

## 6.6 トポロジの検出、ダンプおよび検証

`discover topology` コマンドは、Oracle Internet Directory に問い合せることによって、本番サイトの同じ Oracle Internet Directory を共有する、トポロジ内のすべてのインスタンスを検出します。トポロジのすべてのインスタンスを記述するトポロジ XML ファイルが作成され、トポロジ内のすべての Oracle ホームに配布されます。このトポロジ・ファイルは、すべての OracleAS Guard 操作で使用されます。

トポロジ XML ファイルを内部的に作成するために、OracleAS Disaster Recovery 環境を最初に設定するとき、`discover topology` コマンドを実行する必要があります。したがって、本番サイトに別の Oracle ホームを配置した場合や、スイッチオーバーまたはフェイルオーバー操作によって本番サイトからスタンバイ・サイトに役割を変更した場合は、常にトポロジの検出操作を実行する必要があります。詳細は、「`discover topology`」コマンドを参照してください。

トポロジを記述する情報を調べるには、`dump topology` コマンドを実行する必要があります。詳細は、「`dump topology`」コマンドを参照してください。トポロジ・ファイルの詳細は、第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。

プライマリ・トポロジが実行中であり、その構成が有効であることを確認するには、`verify topology` コマンドを実行する必要があります。さらに、`with host` パラメータを指定した場合、検証操作によって、ローカル・ホスト・システムがメンバーとなっているプライマリ・トポロジと、スタンバイ・トポロジが比較され、それらが相互に一致しており、OracleAS Disaster Recovery の要件を満たしていることが確認されます。詳細は、第 6.15.1 項「トポロジの検証」および「`verify topology`」コマンドを参照してください。

`dump topology` と `verify topology` コマンドの両方で、ポリシー・ファイルを使用する場合は、ダンプおよび検証それぞれのポリシー・ファイル (`dump_policy.xml` および `verify_policy.xml`) を編集して使用します。このファイルを各コマンドの `using policy <file>` パラメータで指定し、それによって指定されたインスタンスのみをダンプまたは検証します。詳細は、第 6.7 項「いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。

## 6.7 いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用

OracleAS Disaster Recovery は、第 6.1.3 項「サポートされているトポロジ」に説明するように各種アプリケーション・サーバー・トポロジのサポートを提供しています。このサポートの一部として、一連の XML フォーマットのポリシー・ファイルが、`dump policies` コマンドを実行する OracleAS Guard クライアントにローカルに維持され、インスタンス別に、各 `asgctl` コマンド (`dump topology`、`verify topology`、`clone topology`、`failover`、`instantiate topology`、`switchover topology` および `sync topology`) に許可される実行操作のドメインが記録されます。

これらの `asgctl` コマンドに使用されているデフォルト・ポリシーを調べるには、`asgctl` プロンプトで次のコマンドを入力します。

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory "/private1/OraHome2/asr1012/dsa/conf/"
ASGCTL>
```

各 XML ポリシー・ファイルの各インスタンスのリストのエントリは、デフォルトでは、本番とスタンバイのピア組合せと、各コマンドの正常な操作のための成功要件を定義する特定の属性を論理的に結び付けます。この方法によって、サポートされているトポロジでこれらの各 OracleAS Guard 操作を正常に使用方法を柔軟に規定できます。詳細は、第 6.1.3 項「サポートされているトポロジ」を参照してください。

各 XML フォーマットのポリシー・ファイルを調べた後、それぞれのポリシー・ファイルを編集し、特定の `asgctl` コマンドのパラメータ構文 `using policy <file>` にそれを使用することによって、使用するポリシー・ファイルの名前を指定できます。このようにして、この章ですでに説明した OracleAS Guard の操作ごとに、インスタンス別の成功要件の属性値を定義する特定の障害時リカバリ・ポリシーを使用できます。

---

**注意：** カスタム・ポリシー・ファイルのセットを維持する場合は、それらをコピーして編集し、デフォルトの場所以外の場所に保管します。デフォルトの場所に保管すると、カスタム・ポリシー・ファイルのセットは、`discover topology` コマンドとその後に `dump policies` コマンドを実行するたびに上書きされます。

---

成功要件の属性値は、次のいずれかです。[`optional` | `mandatory` | `ignore` | `group <MinSucceeded=<number>>`] それぞれの値の意味は次のとおりです。

- **Optional:** そのインスタンスに障害がある場合、他のインスタンスの処理を続行します。
- **Mandatory:** このインスタンスにエラーが発生した場合、操作全体が失敗することを意味します。
- **Ignore:** インスタンスが操作の一部ではないことを意味します。

- Group <MinSucceeded=<number>: Oracle インスタンスのグループを結合することを意味し、指定された数のグループ・メンバーが正常に処理された場合、操作は成功します。指定された数よりも正常に処理されたグループ・メンバーの数が少ない場合、操作は失敗します。

各属性値は、そのピア・グループの成功要件を決定し、asgctl 操作に障害が発生した場合に参照され、OracleAS Guard の操作を続行するかどうか決定されます。たとえば、成功要件が必須であると指定された場合、OracleAS Guard の特定の操作は、本番とスタンバイのピア組合せに指定されたインスタンスに対して正常に実行される必要があります。正常に実行されなかった場合は、OracleAS Guard のその操作が中止され、実行はその開始時点までロールバックされ、エラー・メッセージが返されます。

たとえば、フェイルオーバー操作の非対称トポロジに使用されている次の XML ポリシー・ファイルでは、この asgctl 操作が infra インスタンスには必須であり、portal\_1 インスタンスおよび portal\_2 インスタンスにはオプションであり、portal\_3 インスタンスには無視でき、3つのインスタンス BI\_1、BI\_2 および BI\_3 のグループのうち、少なくともいずれか2つに対しては正常に実行される必要があると指定しています。

```
<policy>
  <instanceList successRequirement="Mandatory">
    <instance>infra</instance>
  </instanceList >
  <instanceList successRequirement="Optional">
    <instance>portal_1</instance>
    <instance>portal_2</instance>
  </instanceList >
  <instanceList successRequirement="Ignore">
    <instance>portal_3</instance>
  </instanceList >
  <instanceList successRequirement="Group" minSucceed="2">
    <instance>BI_1</instance>
    <instance>BI_2</instance>
    <instance>BI_3</instance>
  </instanceList >
</policy>
```

## 6.8 冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ内の OracleAS 10.1.3 インスタンスの検出

第 6.1.3.5 項「冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジ」で説明したように、冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジで、OracleAS 10.1.3 インスタンスの変更を検出できます。一般的な使用例では、OracleAS 10.1.3 J2EE は OC4J インスタンスを含むノードにインストールされており、既存の冗長な複数の OracleAS 10.1.3 ホーム J2EE トポロジにこのノードとインスタンスを追加します。次の手順で、一般的な使用例について説明します。

1. OracleAS Guard サーバーに接続します。

```
ASGCTL > connect asg prodinfra ias_admin/<adminpwd>
Successfully connected to prodinfra:7890
ASGCTL>
```

2. Disaster Recovery 管理者が、OC4J インスタンスを含む新しいノードに OracleAS 10.1.3 J2EE をインストールしたとします。動的なノードの検出により、クラスタは自己管理を行い、クラスタ内の各ノードにある各 OPMN 構成ファイル opmn.xml 内の情報を自動的に更新します。

- 次に、**discover topology within farm** コマンドを実行します。この操作により、この本番サイトの OracleAS 10.1.3 クラスタ内のインスタンスがすべて検出され、本番トポロジについて記述した Disaster Recovery のトポロジ XML ファイルが生成され、このトポロジ XML ファイルが Disaster Recovery 本番環境のインスタンスすべてに伝播されます。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ環境全体に本番トポロジ・ファイルが自動的に伝播されます。

```
ASGCTL> discover topology within farm
```

- 以前の作業で、oc4j3 という別の oc4j インスタンスを prodmid2 という既存の中間層ホスト・システムにインストールしたので、prodmid2 というホスト・システムを指定した、oc4j3 というインスタンスをローカル・トポロジ・ファイルに追加します。この更新済ローカル・トポロジ・ファイルをこの OracleAS 10.1.3 クラスタの Disaster Recovery 本番環境のインスタンスすべてに伝播するには、**to topology** キーワードを指定します。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ・トポロジ全体に本番トポロジ・ファイルが自動的に伝播されます。

```
ASGCTL> add instance oc4j3 on prodmid2 to topology
```

## 6.9 既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除

第 6.1.3.6 項「既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合された冗長な単一の OracleAS 10.1.3 Oracle ホーム J2EE トポロジ」で説明したように、既存の OID 10.1.2.0.2 トポロジでは OracleAS 10.1.3 単一ホーム J2EE インスタンスを追加または削除して、バージョンが混在する Disaster Recovery 環境を作成または変更できます。次の手順で、一般的な使用例について説明します。

- OracleAS Guard サーバーに接続します。

```
ASGCTL > connect asg prodinfra ias_admin/<adminpwd>  
Successfully connected to prodinfra:7890  
ASGCTL>
```

- 既存の OID 10.1.2.0.2 トポロジで Disaster Recovery 環境を最初に設定したときに、**discover topology** コマンドを実行し、Disaster Recovery 環境のインスタンスすべてに伝播されたトポロジ XML ファイルを作成したものと想定します。また、トポロジにはその他の変更は行われておらず、トポロジ XML ファイルは最新の状態になっているものと想定します。
- 4 つの個別のシステム上の単一の Oracle ホームにおいて、統合された Web サーバー、J2EE サーバーおよび OPMN インストール・オプションを使用して、OracleAS 10.1.3 のインストールを行ったものとします。これによって、冗長な単一の Oracle ホーム J2EE トポロジが作成されます。OracleAS Disaster Recovery ソリューションが、新しい各ノードに対して構成されているものとします。ここで、OC4J1 と OC4J2 という OracleAS 10.1.3 のインスタンスを既存の OID 10.1.2.0.2 トポロジに追加します。これを行うには、次の **asgctl** コマンドを実行します。

```
ASGCTL> add instance oc4j1 on prodinfra to topology  
ASGCTL> add instance oc4j2 on prodinfra to topology
```

各 `add instance` コマンドを実行すると、OracleAS Guard クライアントの接続先の OracleAS Guard サーバーでローカル・トポロジ・ファイルが更新されます。更新されたトポロジ・ファイルを Disaster Recovery の本番環境のインスタンスすべてに伝播するには、`to topology` キーワードを指定します。この操作によって、既存の OID 10.1.2.0.2 本番トポロジに統合された `oc4j1` および `oc4j2` インスタンスを含む、冗長な単一の OracleAS 10.1.3 Oracle ホーム J2EE トポロジが作成されます。検証、インスタンス化、同期化、スイッチオーバー、フェイルオーバーなど、スタンバイ・サイトに影響を及ぼす OracleAS Guard 操作を行うと、スタンバイ・トポロジ全体に本番トポロジ・ファイルが自動的に伝播されます。

## 6.10 OracleAS Guard 操作 : スタンバイ・システムへの1つ以上の本番インスタンスのスタンバイ・サイト・クローニング

スタンバイ・サイト・クローニングとは、1つの本番インスタンスを1つのスタンバイ・システムにクローンするか (`clone instance` コマンド)、2つ以上の本番インスタンスを複数のスタンバイ・システムにクローンする (`clone topology` コマンド) プロセスです。どちらの場合も、各インスタンスのトポロジ XML ファイルはクローン操作によりコピーされて、各インスタンスで一貫したものになり、そのスタンバイ・システムでローカルのみで更新されます。ただし、検証、インスタンス化、同期化、スイッチオーバー、フェイルオーバー操作など、スタンバイ・サイトに影響を及ぼす OracleAS Guard 操作が実行されると、本番トポロジ内のローカル・システムにある XML トポロジ・ファイルがスタンバイ・トポロジの全ノードのインスタンスすべてにただちに自動的に伝播されます。

### Clone Instance

`clone instance` コマンドを使用して、既存の本番インスタンス・ソースから新しいスタンバイ・インスタンス・ターゲットを作成します。

この操作を実行するために OracleAS Guard が使用する基礎となるテクノロジーの1つとして、ホストの破損に対する OracleAS バックアップおよびリストア機能があります。前提条件のリストなどの詳細は、『Oracle Application Server 管理者ガイド』のホストの破損の自動リカバリに関する項を参照してください。この機能は、ターゲット・マシンが新しく作成された Oracle 環境であることを想定しています。これは、それが Oracle ソフトウェア・レジストリを上書きするためです。さらに、基礎となる操作のいくつかには、UNIX 環境の場合は `root`、および Windows の場合は `Administrator` といった高い権限が必要です。Windows では、ユーザーは、クライアントと OracleAS Guard サーバーが `Administrator` 権限によって起動されていることを確認する必要があります。

クローンには次の2つのフェーズがあります。最初のフェーズは、Oracle ホームを作成し、それをシステム環境内に登録することです。2番目のフェーズは、OracleAS Guard インスタンス化操作を実行し、それを OracleAS Disaster Recovery 環境にリンクし、Oracle ホームをそれに対応する本番ホームと論理的に一致させることです。

様々なインスタンスでの一連のインスタンスのクローン操作は、1つのトポロジのクローン操作と同等です。

### Clone Topology

`clone topology` コマンドは、1つのグループのシステム全体にインスタンスのクローン操作を実行します。クローン操作は、データベースを含まないすべての OracleAS ホームで実行され、ポリシー・ファイルを使用してフィルタできます。データベースを含む OracleAS ホームの場合、トポロジのクローン操作によって、操作のインスタンス化フェーズが実行され、スタンバイ・サイトでの Oracle ホームの作成をスキップできます。この操作は、ポリシー・ファイルを使用することによってトポロジのサブセットに実行できます。

OracleAS Disaster Recovery のサイトの設定を計画する際には、次の3つの方法論を認識する必要があります。

- 純 OracleAS Disaster Recovery サイトの作成
- OracleAS Disaster Recovery が有効化されている既存のサイトへの OracleAS ホームの追加
- 既存のデータベース内の OracleAS Metadata Repository の統合

各操作では、新しくインストールした Oracle ホームを既存のサイトに統合したり、それらを本番サイトのスタンバイ・サイトに結合するための様々な方法が必要です。

### 純 OracleAS Disaster Recovery サイトの作成

OracleAS 10g リリース 2 (10.1.2.0.2) より前は、これが OracleAS Guard がサポートする唯一のタイプのサイトでした。OracleAS Disaster Recovery 構成は、デフォルトの Infrastructure および OracleAS 中間層のインストール・タイプに対してのみサポートされていました。このタイプの構成では、すべての OracleAS ホームは Oracle インストーラを使用して作成されました。OracleAS Guard のインスタンス化コマンドによって、本番およびスタンバイの Oracle ホームと基礎となっているスタンバイ Oracle データベース・リポジトリとの間の関係が作成されます。

### OracleAS Disaster Recovery が有効化されている既存のサイトへの OracleAS ホームの追加

OracleAS サイトで OracleAS Disaster Recovery が有効化された後、本番 Oracle ホームとスタンバイ Oracle ホームとの間の関係が作成されます。OracleAS 10g リリース 2 (10.1.2.0.2) の場合、新しいインスタンスをサイトに追加する唯一の方法は、スタンバイの関係を壊し、Oracle Installer を使用して本番サイトで新しいインスタンスを追加し、Oracle Installer を使用してその新しいインスタンスをスタンバイ・サイトに追加し、スタンバイ・サイトを再作成することでした。OracleAS 10g リリース 2 (10.1.2.0.2) では、clone instance コマンドを使用してインスタンスをスタンバイ・サイトに追加できます。

たとえば、中間層のサービスを増やすために新しい中間層が必要な場合、新しいインスタンスを本番サイトにインストールします。この操作によって、そのインスタンスの OracleAS ホームが作成され、OracleAS リポジトリ内に必要な関係が確立されます。

OracleAS 10g リリース 2 (10.1.2.0.2) の OracleAS Guard の非対称トポロジ・サポートでは、この Oracle ホームは、このサイトの OracleAS Disaster Recovery ソリューションについては無視することもできます。このインスタンスをスタンバイ・サイトに追加する場合は、clone topology コマンドでスタンバイ・ターゲット・ホストに OracleAS Oracle ホームを作成し、このインスタンスに対する本番とスタンバイの関係を確立します。このコマンドを発行する前に、OracleAS Guard のスタンドアロン・キットをターゲット・ホストにインストールして起動し (詳細は、Oracle Application Server のインストレーション・ガイドの OracleAS Disaster Recovery のインストール情報を参照)、サイトのトポロジ検出操作を実行して、本番トポロジの新しいインスタンスを検出する必要があります。

### 既存のデータベース内の OracleAS Metadata Repository の統合

OracleAS は、既存のデータベースに Metadata Repository スキーマを作成する機能をサポートしています。OracleAS Guard では、これらのデータベースが認識および管理され、Metadata Repository 構成データがサイトの残りの分散構成データと同期化されますが、スタンバイ・リポジトリも本番とスタンバイの関係も作成されません。この環境は、トポロジのクローン操作を使用してサポートします。

clone topology コマンドを利用するには、最初にスタンドアロン OracleAS Guard サーバーを各スタンバイ・ホストにインストールして起動します。さらに、スタンドアロン OracleAS Guard のインストールによって作成された Oracle ホームに OracleAS のバックアップおよびリストア・ユーティリティをインストールします。前提条件のリストなどの詳細は、『Oracle Application Server 管理者ガイド』のホストの破損の自動リカバリに関する項を参照してください。clone topology コマンドによって、スタンバイ・サイトにインスタンス Oracle ホームと構成情報が作成されます。Infrastructure インスタンスの場合は、暗黙的インスタンス化操作を実行して、OracleAS Disaster Recovery 環境を初期化します。トポロジのクローニング操作は、プロファイル・ファイルを使用して、非対称トポロジのインスタンスをフィルタで除外できます。

---

**警告：** クローン操作を、スタンドアロンの OracleAS Guard ホーム以外の既存の Oracle ホームを含むスタンバイ・システムに実行しないでください。実行すると既存の Oracle ホームが上書きされます。クローン操作は、Oracle ホーム (スタンドアロンの OracleAS Guard ホームを除く) がインストールされていないスタンバイ・システムにのみ実行してください。

---

クローン操作は、次のような状況で便利です。

- スタンバイ・ホスト・サイトに1つ以上のスタンバイ・ホスト・サイトを追加する場合。
- 1つの本番インスタンスをスタンバイ・ホスト・システムに追加する場合。

これらのクローン操作を実行する手順は、次の項で説明します。

## 6.10.1 スタンバイ・システムへの1つまたは複数の本番インスタンスのクローニング

スタンバイ・サイトに1つの本番インスタンスをクローニングする場合も、複数の本番インスタンスをクローニングする場合も、前提条件および従う手順は同じです。唯一の相違は、クローニング操作に使用する実際の `asgctl` コマンドです。このため、この項ではクローニング情報を統合しており、内容が若干異なる場合にのみその旨を示しています。

### スタンバイ・サイトへの1つの本番インスタンスのクローニング

例として、1つの本番インスタンスをスタンバイ・ホスト・システムに追加するとします。インスタンスのクローン操作を使用すると、Oracle インスタンスをスタンバイ中間層システムにインストールしてからインスタンス化操作を実行するという作業を行わずに済みます。

### スタンバイ・サイトへの複数の本番インスタンスのクローニング

例として、2つ以上の本番インスタンスをスタンバイ中間層ホスト・システムに追加するとします。トポロジのクローン操作を使用すると、Oracle インスタンスをスタンバイ中間層システムにインストールしてからインスタンス化操作を実行するという作業を行わずに済みます。

トポロジのクローン操作の一部として、本番インスタンスがクローンされ、OracleAS Metadata Repository がインスタンス化されます。ただし、OracleAS Metadata Repository Creation Assistant を使用して作成された OracleAS Metadata Repository 構成の場合、インスタンス化操作は実行されません。

ポリシー・ファイルを使用する場合、クローン・ポリシー・ファイル (`clone_policy.xml`) を編集して使用します。このファイルを `clone topology` コマンドの `using policy <file>` パラメータで指定し、それによって指定されたインスタンスに対してのみスタンバイ・トポロジをクローンします。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。

### 特別な考慮事項

複数の本番インスタンスをスタンバイ・サイトにクローニングする前に、次のことを考慮します。

- 各ノードは、ホスト・ファイル内で物理ホスト名の IP アドレスにマッピングされる独自の仮想ホスト名を持つ必要があります。たとえば、次のファイルを編集します。

Windows の場合 : `%SystemRoot%\system32\drivers\etc\hosts`

UNIX の場合 : `/etc/hosts`

`ip_address prodnode1.domain.com prodnode1 vhostdr.domain.com vhostdr`

- 非対称のノードとは、たとえばプライマリ・サイトの2つのノードにそれぞれ1つずつインスタンスがあり、スタンバイ・サイトの1つノードに2つのインスタンスがある場合などです。インスタンス数は同じですがノード数が異なります。
- 非対称のノードのクローニングはサポートされていません。

### 前提条件

本番インスタンスまたはクローニングするインスタンスは、スタンバイ・システムに存在してはなりません。

スタンバイ・サイト・システムにインスタンスとトポロジのクローン操作を実行するための前提条件は次のとおりです。

- スタンバイ・システムに OracleAS Guard スタンドアロン・キットがインストールされている必要があります。
- スタンバイ・システムの OracleAS Guard ホームにバックアップおよびリストアがインストールされている必要があります。
- Java Development Kit をその jar ユーティリティとともにスタンバイ・システムにインストールする必要があります。
- Windows システムの場合、サービス・キット (sc.exe) をスタンバイ・システムにインストールする必要があります。

### 手順

クローニングの基本手順は、次のクローニング前 (UNIX システムの場合のみ) およびクローニング中の各手順から構成されています。

---

---

**注意:** Windows システムの OracleAS Disaster Recovery 10g リリース 2 (10.1.2.0.2) については、OracleAS 10g リリース 2 (10.1.2.0.2) ドキュメント・セットの『Oracle Application Server 高可用性ガイド』の同じ項に記載されている、クローニング前とクローニング後の手順を参照してください。

---

---

#### クローニング前の手順 (UNIX システムの場合のみ)

本番サイトとスタンバイ・サイトの各インスタンスで、次の手順を実行します。

1. su - root でログインします。
2. インスタンス・ホームへ移動します。
3. OracleAS Guard サーバーを停止します。  

```
> <ORACLE_HOME>/opmn/bin/opmnctl stopproc ias-component=ASG
```
4. <ORACLE\_HOME>/dsa/bin 内の dsaServer.sh をすべてのユーザーが実行できることを確認します。実行できない場合は、現在の実行権限を書き留めてから、次のコマンドを実行して実行権限を変更します。  

```
chmod +x dsaServer.sh  
chmod u+x asgexec
```
5. asgctl を起動し、startup コマンドを実行します。  

```
> asgctl.sh startup
```
6. UNIX システムの root をログアウトします。

#### クローニング中の手順

本番サイトのインスタンスで、次の手順を実行します。

1. ユーザー (UNIX システムの root 以外のユーザー) としてログインします。
2. 本番インスタンスのホームへ移動します。

3. asgctl を起動してから clone instance コマンドを実行して、インスタンスをスタンバイ・トポロジのホスト・システムにクローニングします。

---

**注意：** コマンド出力には、多数の接続メッセージが表示されます。これは、これらの操作中に OracleAS Guard サーバーがリサイクルされるために発生する正常な動作です。

---

```
> asgctl.sh
Application Server Guard: Release 10.1.3.0.0
(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL> connect asg prodoc4j oc4jadmin/adminpwd
Successfully connected to prodoc4j:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb

CLONE INSTANCE -- CLONE AN INSTANCE EXAMPLE

ASGCTL> clone instance portal_2 to asmid2
Generating default policy for this operation
.
.
.

CLONE TOPOLOGY -- CLONE MULTIPLE INSTANCES EXAMPLE

# Command to use if you are using a policy file where <file>
# is the full path and file spec of the clone policy file.
ASGCTL> clone topology to standbyinfra using policy <file>
Generating default policy for this operation
.
.
.

ASGCTL> disconnect
ASGCTL> exit
>
```

4. システムからログアウトします。

---

**注意：** UNIX システムでは、OracleAS Guard が root として実行されていない場合、操作を続行するには、各インスタンス・ホームの基本的な操作を root として（手動で）実行する必要があることが OracleAS Guard クライアントによって示されます。

---

最後の手順で、インスタンスのクローニング操作が完了し、操作を開始する前の状態にシステムが戻ります。この時点で asgctl を起動し、本番システムに接続し、トポロジを検出し、検証操作を実行して、本番トポロジとスタンバイ・トポロジが有効で、予期したとおりに両方のトポロジが一致しているかどうかを調べます。

## 6.11 OracleAS Guard の操作 : スタンバイのインスタンス化とスタンバイの同期化

次の条件を満たしている場合、Oracle Application Server Guard を使用して、スタンバイのインスタンス化とスタンバイの同期化を実行できます。

- OracleAS Disaster Recovery ソリューションの実装に関する、第 6.1.1 項「OracleAS Disaster Recovery の要件」、第 6.1.3 項「サポートされているトポロジ」および第 6.2 項「OracleAS Disaster Recovery 環境の準備」の要件を満たしていること。
- 第 6.3 項「Oracle Application Server のインストールの概要」の説明に従って、OracleAS Disaster Recovery (DR) ソリューションをインストールしていること。

次の項目では、スタンバイのインスタンス化とスタンバイの同期化について説明します。

OracleAS Guard コマンドライン `asgctl` ユーティリティのリファレンス情報は、第 7 章「OracleAS Guard `asgctl` コマンドライン・リファレンス」を参照してください。

### 6.11.1 スタンバイのインスタンス化

スタンバイのインスタンス化操作では、スタンバイ・サイトで論理的な本番サイトのミラー化を設定し、維持するためのいくつかの操作を実行します。OracleAS Guard は、障害時リカバリの機能を実現するために、本番サイトとスタンバイ・サイト間での分散操作の調整に使用されます。設定操作は次のとおりです。

- トポロジの検出操作を実行して作成した前のトポロジ・ファイルを使用します。
- トポロジの定義を調べ、OracleAS Disaster Recovery 環境のルールを順守していることを確認します。
- Oracle Data Guard を構成して、データベース・リポジトリに対する OracleAS Disaster Recovery 環境を維持します。
- OracleAS トポロジ内のすべての Oracle ホームの構成情報を、スタンバイ・サイトの対応する Oracle ホームにミラー化します。
- ポリシー・ファイルを使用する場合、インスタンス化ポリシー・ファイル (`instantiate_policy.xml`) を編集して使用します。このファイルを `instantiate topology` コマンドの `using policy <file>` パラメータで指定し、それによって指定されたインスタンスに対してのみスタンバイ・トポロジをインスタンス化します。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。
- 修正が必要なエラーを報告します。

スタンバイのインスタンス化操作を実行する手順では、次の例を使用します。この例では、OracleAS Guard クライアントを起動して、`discover topology` コマンドを実行してトポロジ・ファイルを作成してあると想定しています。

CFC 環境に OracleAS Disaster Recovery 構成があり、インスタンス化操作を実行しようとしている場合は、第 7.2.1.1 項「CFC 環境でインスタンス化およびフェイルオーバー操作を実行するときの特別な考慮事項」を参照してください。

1. OracleAS Guard サーバーに接続します。

```
ASGCTL > connect asg prodinfra ias_admin/<adminpwd>
Successfully connected to prodinfra:7890
ASGCTL>
```

2. プライマリ OracleAS Metadata Repository データベースを指定します。詳細は、第 6.17.1.2 項「プライマリ・データベースの指定」を参照してください。トポロジに OracleAS Metadata Repository が複数ある場合は、`set primary database` コマンドを使用して、それぞれを認証する必要があります。

```
ASGCTL> set primary database sys/testpwd@asdb
```

3. ポリシーをダンプし (`dump policies` コマンド)、検証ポリシー・ファイル (`verify_policy.xml`) およびインスタンス化ポリシー・ファイル (`instantiate_policy.xml`) を編集して使用し、ファイル内に各インスタンスの成功要件の属性を指定します。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory
"/private1/OraHome2/asr1012/dsa/conf/"
```

4. トポロジを検証します。ネットワーク・ホスト名 `standbyinfra` が使用されています。

```
ASGCTL> verify topology with standbyinfra
```

5. セカンダリ・サイトでトポロジをインスタンス化します。ネットワーク・ホスト名 `standbyinfra` が使用されています。このコマンドは、すべての Oracle ホームが Oracle インストーラ・ソフトウェアを使用してインストールされていると想定しています。`using policy <file>` パラメータを指定します。ここで `<file>` は、`instantiate_policy.xml` ファイルのパスとファイルの指定を表します。

```
ASGCTL> instantiate topology to standbyinfra using policy <file>
```

`asgctl instantiate topology` コマンドを使用して、スタンバイのインスタンス化を実行するたびに、同期化操作も実行されます。したがって、インスタンス化操作の後、すぐに別の同期化操作を実行する必要はありません。インスタンス化操作の後に一定期間経過した場合は、プライマリ・サイトとスタンバイ・サイトが一致していることを確認します。その後、トポロジの同期化操作を実行し、プライマリ・サイトで行われたすべての変更がセカンダリ・サイトに適用されるようにします。

## 6.11.2 スタンバイの同期化

OracleAS Guard の同期化操作は、スタンバイ・サイトとプライマリ・サイトを同期化し、2 つのサイトを論理的に一致させます。この操作は、次のような状況の場合に必要なになります。

- 新しいアプリケーションの配置、または既存のアプリケーションの再配置: 新しいアプリケーションを配置したり、既存のアプリケーションを再配置する場合は、メタデータ・リポジトリのスキーマ・ベースの情報を変更し、OracleAS トポロジ内の Oracle ホームに分散しているコンポーネント構成情報を変更する必要があります。これと同じ情報をスタンバイ・サイトで反映する必要があります。
- 構成の変更: 構成を変更する場合は、その変更の規模にかかわらず、スタンバイ・サイトに反映させる必要があります。
- ユーザーのプロビジョニング: デフォルトの **Infrastructure** のインストールでは、Oracle **Internet Directory** のデータベースが維持されます。新規ユーザーをデータベースに追加した場合は、ビジネスの可用性の要件を満たすスケジュールで、スタンバイ・サイトと同期化する必要があります。
- 定期的な完全同期化: デフォルトの同期化の操作は、最後の同期化操作から変更された構成部分のみが同期化されます。テスト・サイクルや特別に複雑な構成変更を行う場合は、管理者がスタンバイ・サイトの構成情報を完全にリフレッシュして、変更内容を完全にミラー化することができます。

完全な同期化を指定することも増分の同期化を指定することもできます。デフォルトでは、増分の同期化が実行されます。これは最も高いパフォーマンスを提供します。ただし、次の 3 つの状況では、完全な同期化を指定する必要があります。

- スタンバイ・サイトとプライマリ・サイトを完全に一致させるなど、なんらかの理由で完全な同期化を強制する必要がある場合。
- セカンダリ・サイトと同期するプライマリ・サイトで、短時間に多数のトランザクションの変更が発生することがわかっている場合。

- セカンダリ・サイトと同期するプライマリ・サイトで、長時間にわたって多数のトランザクションの変更が蓄積していることがわかっている場合。

同期化操作の際に検証操作が実行されて、必要な OracleAS Disaster Recovery 環境が維持されていることが確認されます。また、OracleAS トポロジに新しい OracleAS インスタンスをインストールした場合は、OracleAS Guard によってこれらのインストールが検出されます。

ポリシー・ファイルを使用する場合、同期化ポリシー・ファイル (sync\_policy.xml) を編集して使用します。このファイルを `sync topology` コマンドの `using policy <file>` パラメータで指定し、それによって指定されたインスタンスに対してのみスタンバイ・トポロジを同期化します。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンブとポリシー・ファイルの使用」を参照してください。

次の例では、OracleAS Guard クライアントを起動して、`discover topology` コマンドを実行してトポロジ・ファイルを作成してあると想定しています。

スタンバイの同期化を実行する手順は次のとおりです。

1. OracleAS Guard サーバーに接続します。

```
ASGCTL > connect asg prodinfra ias_admin/<adminpwd>
Successfully connected to prodinfra:7890
ASGCTL>
```

2. プライマリ・データベースを指定します。詳細は、第 6.17.1.2 項「プライマリ・データベースの指定」を参照してください。

```
ASGCTL> set primary database sys/testpwd@asdb
```

3. セカンダリ・サイトとプライマリ・サイトを同期化します。

```
ASGCTL> sync topology to standbyinfra
```

## 6.12 実行時操作 : OracleAS Guard のスイッチオーバーおよびフェイルオーバー操作

実行時操作には、スケジューリングした停止と計画外停止を含む停止の操作（第 6.12.1 項「停止」を参照）、`asgctl` コマンドライン・ユーティリティを使用した実行中の OracleAS Guard 操作を監視する操作およびトラブルシューティング（第 6.15 項「OracleAS Guard 操作の監視とトラブルシューティング」を参照）があります。

### 6.12.1 停止

停止には、スケジューリングした停止と計画外停止の 2 つのカテゴリがあります。

次のサブセクションではこれらの停止について説明します。

#### 6.12.1.1 スケジューリングした停止

スケジューリングした停止とは、計画停止のことです。この停止は、ビジネス・アプリケーションをサポートするテクノロジ・インフラストラクチャの定期的なメンテナンスのために必要で、この間に、ハードウェアのメンテナンスや修復とアップグレード、ソフトウェアのメンテナンスとパッチ適用、アプリケーションの変更とパッチ適用、およびシステムのパフォーマンスと管理機能強化を目的とした変更などのタスクが実行されます。スケジューリングした停止は、本番サイトまたはスタンバイ・サイトのどちらでも実行できます。次に、本番サイトまたはスタンバイ・サイトに影響を与えるスケジューリングした停止について説明します。

- サイトレベルのメンテナンス

現行の本番環境があるサイト全体が使用できなくなります。サイトレベルのメンテナンスの例には、スケジューリングした停電、サイトのメンテナンス、定期的に計画されたスイッチオーバー操作などがあります。

- OracleAS Cold Failover Cluster クラスタレベルのメンテナンス

これは、ハードウェア・メンテナンスのためにスケジューリングした、OracleAS Cold Failover Cluster の停止時間です。この停止時間は、ハードウェア・クラスタ全体に影響を与えます。クラスタレベルのメンテナンスの例には、クラスタ・インターコネクトの修復、クラスタ管理ソフトウェアのアップグレードなどがあります。

- スタンバイ・サイトのテストおよび検証 (OracleAS Disaster Recovery が万全かどうかをテストする)

スケジューリングした停止では、次の項で説明するサイト・スイッチオーバー操作を実行する必要があります。

### サイト・スイッチオーバー操作

サイト・スイッチオーバーは、本番サイトの計画停止の際に実行されます。スイッチオーバー時には、本番サイトとスタンバイ・サイトの両方が使用可能である必要があります。データベース REDO ログは、中間層および OracleAS Infrastructure インストールの構成ファイルのバックアップおよびリストアと同時に適用されます。

---

**注意：** スイッチオーバー操作中に、opmn.xml ファイルがプライマリ・サイトからスタンバイ・サイトにコピーされます。このため、TMP 変数の値はプライマリ・サイトとスタンバイ・サイトの両方の opmn.xml ファイルで同じ値に定義する必要があります。そうしないと、ディレクトリが見つからないというメッセージが表示され、このスイッチオーバー操作は失敗します。したがって、TMP 変数が同じ値に定義され、スイッチオーバー操作を試行する前に両方のサイトの同じディレクトリ構造に解決されることを確認してください。

---

サイト・スイッチオーバー時は、DNS 情報が長時間キャッシュされないように配慮する必要があります。そのため、サイトの DNS 情報の変更、具体的には Time-To-Live (TTL) 値の変更が必要です。手順は、第 6.16.2 項「DNS 名の手動変更」を参照してください。

ポリシー・ファイルを使用する場合、スイッチオーバー・ポリシー・ファイル (switchover\_policy.xml) を編集して使用します。このファイルを `switchover topology` コマンドの `using policy <file>` パラメータで指定し、それによって指定されたインスタンスに対してのみスタンバイ・トポロジにスイッチオーバーします。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。この例では、ポリシー・ファイルの使用例は示しません。

CFC 環境に OracleAS Disaster Recovery 構成があり、スイッチオーバー操作を計画している場合は、第 7.2.1.3 項「CFC 環境でスイッチオーバー操作を実行するときの特別な考慮事項」を参照してください。

本番サイトからスタンバイ・サイトにスイッチオーバーする手順は次のとおりです。

1. サイト用のワイド・エリア DNS の TTL 値を減らします。詳細は、第 6.16.2 項「DNS 名の手動変更」を参照してください。
2. プライマリ Infrastructure システムで、`emagent` プロセスが停止していることを確認します。停止していない場合は、`emagent` がデータベースと接続されているため、スイッチオーバー操作の実行時に次のエラーが発生することがあります。

```
prodinfra: -->ASG_DGA-13051: Error performing a physical standby switchover.
prodinfra: -->ASG_DGA-13052: The primary database is not in the proper state to
perform a switchover. State is "SESSIONS ACTIVE"
```

UNIX システムの場合、次のようにして Application Server Control を停止し (`iasconsole`)、`emagent` プロセスを停止します。

```
> <ORACLE_HOME>/bin/emctl stop iasconsole
```

UNIX システムの場合、**emagent** プロセスが実行されていないかを確認するために、次のコマンドを入力します。

```
> ps -ef | grep emagent
```

UNIX システムの場合、**iasconsole** の停止操作の後で **emagent** プロセスが依然として実行されている場合は、前の **ps** コマンドで示したようにプロセス ID (PID) を取得し、次のようにして **emagent** プロセスを停止します。

```
> kill -9 <emagent-pid>
```

Windows システムの場合、「サービス」コントロールパネルを開きます。OracleAS10gASControl サービスを見つけて、このサービスを停止します。

- OracleAS Guard クライアントのコマンドライン・ユーティリティ **asgctl** を起動し、OracleAS Guard サーバーに接続します。UNIX システムでは、**asgctl.sh** は **<ORACLE\_HOME>/dsa/bin** にあり、Windows システムでは、**asgctl.bat** は **<ORACLE\_HOME>%dsa%bin** にあります。

```
> asgctl.sh
Application Server Guard: Release 10.1.2.0.2
(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL> connect asg prodinfra ias_admin/<adminpwd>
```

- プライマリ・データベースを指定します。詳細は、[第 6.17.1.2 項「プライマリ・データベースの指定」](#)を参照してください。

```
ASGCTL> set primary database sys/testpwd@asdb
```

- Oracle RAC の障害時リカバリの配置では、スイッチオーバーを実行する前にすべてのインスタンスを停止します。

---

**注意：** この例では、Oracle ホームにインストールして、分散された ASG のスクリプティング機能を利用するスクリプトを作成します。これにより、システム管理者は **asgctl** ユーティリティからすべてのスイッチオーバー操作を実行できます。**srvctl** ユーティリティで、クラスタ内のすべてのインスタンスを停止します。

---

- shutdown\_asdb\_instance.sh** スクリプトを作成し、スクリプト内で指定している場所にコピーします。これは Oracle ホーム内の場所である必要があります。

```
#shutdown_asdb_instance.sh for asdb instance
#in /private/oracle/product/10.1.0/asdb on db_site1_node1 & db_site2_node1
export ORACLE_HOME=/private/oracle/product/10.1.0/asdb
$ORACLE_HOME/bin/srvctl stop instance -d asdb -i asdb2
```

- asgctl** の **run** コマンドを使用して、インスタンスを停止するスクリプトを実行します。

```
ASGCTL> run at instance asdb shutdown_asdb_instance.sh
```

- トポロジをセカンダリ・サイトにスイッチオーバーします。ポリシー・ファイルを使用する場合、**using policy <file>** パラメータを指定します。**<file>** は、**switchover\_policy.xml** ファイルのパスとファイルの指定を表します。

```
ASGCTL> switchover topology to standbyinfra
```

---

**注意：** OracleAS Guard のスイッチオーバー操作の際に、**implicit sync topology** 操作が実行され、2 つのトポロジが同一であるか確認されます。さらに、本番サイトですべての OPMN サービスを停止し、再起動します。

---

- 古いプライマリ・サイトの OracleAS Guard サーバーを切断します。

```
ASGCTL> disconnect
ASGCTL>
```

- 第 6.16 項「ワイド・エリア DNS の操作」のオプションの 1 つに基づいたワイド・エリア DNS スwitchオーバーを実行して、リクエストが新しい本番サイトへと送信されるようにします。
- ワイド・エリア DNS の TTL を適切な値に調整します。

### スイッチオーバー操作に関する特別な考慮事項

この項では、スイッチオーバー操作に関する次のような特別な考慮事項について説明します。

- 2 つの Oracle Identity Management インスタンスが稼動しているプライマリ・サイトから、Oracle Identity Management インスタンスが 1 つだけ稼動している非対称トポロジのスタンバイ・サイトへのスイッチオーバー操作を実行する場合、つまり、スイッチオーバー・サイトでもう 1 つのノードを無視する場合は、このスイッチオーバー操作を成功させるには、システム管理者が `switchover_policy.xml` ポリシー・ファイルを編集してノードを `Ignore` に設定するだけでなく、このノードで実行中のすべてのプロセスを停止する必要があります。たとえば、プライマリ・サイトで実行されている 2 つの Oracle Identity Management インスタンスが `im.machineA.us.oracle.com` と `im.machineB.us.oracle.com` であり、もう 1 つのノード (`im.machineB.us.oracle.com`) がスイッチオーバー・サイトで無視される場合、スイッチオーバー操作が正常に実行されるためには、システム管理者はそのノード (`im.machineB.us.oracle.com`) 上で実行されているすべてのプロセスを停止することも必要です。
- スイッチオーバー操作の後に `discover topology` コマンドが実行される際に、非対称スタンバイ・サイト・トポロジに存在する中間層 (例: `instA`、`instB`) が元の本番サイト・トポロジにある中間層 (例: `instA`、`instB`、`instC`) より少ない場合、存在していない中間層のインスタンスごとに警告エラー・メッセージが表示されます (この場合、`instC` に対して表示される)。このメッセージの表示は、予期されている動作であるため無視できます。スイッチオーバー操作の後に `discover topology` コマンドを発行した場合、OracleAS Server Guard によって Oracle Internet Directory 情報が読み取られます。この情報は、この新しいプライマリ・サイト (前のスタンバイ・サイト) 上にある、元のプライマリ・サイトの Oracle Internet Directory 情報の完全なコピーです。この Oracle Internet Directory 情報は、元のプライマリ・サイトの Oracle Internet Directory 情報と完全に同一であるため、OracleAS Server Guard がこれらの中間層の各インスタンスのホストまたはホームでそれらの存在を検証すると、いくつかの中間層が存在しないことが検出され、警告が発行されません。

#### 6.12.1.2 計画外停止

本番サイトに影響を与える計画外停止は、そのサイトが使用不能になり、本番サイトがリストアされて妥当な期間内にサービスを提供する可能性がない場合に発生します。これには、火事、洪水、地震、停電などの本番サイトでのサイトレベルの停止があります。

計画外停止が発生した場合は、本番サイトからスタンバイ・サイトへのフェイルオーバー操作の実行が保証されます。

#### サイト・フェイルオーバー操作

サイト・フェイルオーバー操作は、本番サイトの計画外停止の際に実行されます。フェイルオーバー操作では、構成データと Infrastructure データを、一貫性のあった時点までリストアする必要があります。OracleAS Guard では、最後に `sync` 操作を行った時点から、一貫性のある方法でサイトのサービスが開始されます。フェイルオーバー操作によって、最後の同期化の時点までリストアされます。

ポリシー・ファイルを使用する場合、フェイルオーバー・ポリシー・ファイル (failover\_policy.xml) を編集して使用します。このファイルを `failover` コマンドの `using policy <file>` パラメータで指定し、それによって指定されたインスタンスのみがスタンバイ・トポロジにフェイルオーバーされるようにします。詳細は、第 6.7 項「いくつかの `asgctl` コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。

CFC 環境に OracleAS Disaster Recovery 構成があり、フェイルオーバー操作を実行しようとしている場合は、第 7.2.1.1 項「CFC 環境でインスタンス化およびフェイルオーバー操作を実行するときの特別な考慮事項」を参照してください。

本番サイトからスタンバイ・サイトにフェイルオーバーする手順は次のとおりです。

1. スタンバイ・サイトの OracleAS Guard サーバーに接続します。ネットワーク名は `standbyinfra` です。

```
ASGCTL> connect asg standbyinfra ias_admin/<adminpwd>
Successfully connected to stanfbyinfra:7890
```

2. Oracle RAC の障害時リカバリの配置を行うには、データベースを起動します。

---

**注意：** この例では、Oracle ホームにインストールして、分散された ASG のスクリプティング機能を利用するスクリプトを作成します。これにより、システム管理者は `asgctl` ユーティリティからすべてのフェイルオーバー操作を実行できます。`srvctl` ユーティリティで、クラスタ内のすべてのインスタンスを起動します。

---

- a. `start_asdb_db.sh` スクリプトを作成し、スクリプト内で指定している場所にコピーします。これは Oracle ホーム内の場所である必要があります。

```
#start_asdb_db.sh for asdb database
#in /private/oracle/product/10.1.0/asdb on db_site1_node1 & db_site2_node1
export ORACLE_HOME=/private/oracle/product/10.1.0/asdb
$ORACLE_HOME/bin/srvctl start database -d asdb
```

- b. `asgctl` の `run` コマンドを使用して、データベースを起動するスクリプトを実行します。

```
ASGCTL> run at instance asdb start_asdb_db.sh
```

3. スタンバイ・サイトのプライマリ OracleAS Metadata Repository データベースを新しい本番サイトの新しいプライマリ・データベースに指定します。次の例では、重要なキーワードであることを示すため、**new** を太字のテキストで示しています。トポロジに複数の OracleAS Metadata Repository がある場合、`set new primary database` コマンドを使用して各 OracleAS Metadata Repository を認証する必要があります。

```
ASGCTL> set new primary database sys/testpwd@asdb
```

4. `asgctl failover` 操作を実行します。

```
ASGCTL> failover
```

5. トポロジを検出します。この操作を実行して、この本番サイトの新しいトポロジ・ファイルを作成する必要があります。

```
ASGCTL> discover topology oidpassword=oidpwd
```

## 6.13 Real Application Clusters データベースを使用しない OracleAS Disaster Recovery の構成

この項では、プライマリ・サイトでもスタンバイ・サイトでも Oracle Real Application Clusters データベースを使用しない OracleAS Disaster Recovery トポロジの設定方法について説明します。これは基本的に、OracleAS Disaster Recovery の最も単純な構成です。Real Application Clusters データベースを使用した OracleAS Disaster Recovery トポロジについては、次の項の第 6.14 項「Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery」で説明します。

この項の項目は次のとおりです。

- 第 6.13.1 項「前提条件」
- 第 6.13.2 項「構成手順」
- 第 6.13.3 項「スイッチオーバー手順」
- 第 6.13.4 項「スイッチバック手順」
- 第 6.13.5 項「フェイルオーバー手順」

### 6.13.1 前提条件

次の前提条件に注意してください。

- プライマリ・サイトおよびスタンバイ・サイトの両方のデータベースの Oracle ホームに、OracleAS Guard のスタンドアロン・バージョンをインストールする必要があります。OracleAS Guard のスタンドアロン・バージョンは、Oracle Application Server Companion CD Disk 2 にあります。

スタンドアロンの OracleAS Guard インストーラの実行方法の詳細は、使用しているプラットフォームの Oracle Application Server のインストレーション・ガイドで、「高可用性環境へのインストール」の章を参照してください。

- スタンバイ・サイトのデータベースを停止し、その SID を削除する必要があります。これらの手順（第 6.13.2 項「構成手順」を参照）は、後の手順で create standby database コマンドを実行する前に実行する必要があります。
- スタンバイ・サイトのデータベースは、create standby database コマンドによって上書きされます。
- 表 6-5 は、次に説明する手順で使用するホスト名とデータベース名を示しています。

**表 6-5 プライマリ・サイトおよびスタンバイ・サイトのホスト名とデータベース名**

	プライマリ・サイト	スタンバイ・サイト
物理ホスト名	prodnode1	standbynode1
仮想ホスト名	vhostdr	vhostdr
データベース名	orcl.oracle.com	orcl.oracle.com
データベース SID	orcl	orcl

- 仮想ホスト名 vhostdr は、マシンの IP アドレスにマッピングする必要があります。このマッピングは、各マシンの hosts ファイルで設定します。次に、例を示します。

プライマリ・サイトで、%SystemRoot%\system32\drivers\etc\hosts (Windows) または /etc/hosts (UNIX) を編集し、次のようなエントリを追加します。

```
ip_address prodnode1.domain.com prodnode1 vhostdr.domain.com vhostdr
```

スタンバイ・サイトで、同じ hosts ファイルを編集し、次のようなエントリを追加します。

```
ip_address standbynode1.domain.com standbynode1 vhostdr.domain.com vhostdr
```

## 6.13.2 構成手順

次の手順を実行して、プライマリ・サイトでもスタンバイ・サイトでも Real Application Clusters データベースを使用しない OracleAS Disaster Recovery トポロジを設定します。

1. スタンバイ・サイトのデータベースを停止します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> shutdown;
```

2. standbynode1 で、そのデータベースの SID を削除します。SID を削除しないと、create standby database コマンドの実行時にエラーが発生します。

Windows の場合は、次の oradim コマンドを実行して Oracle SID を削除します。

```
> oradim -delete -sid orcl
```

UNIX の場合は、`oratab` ファイル内でデータベース SID/ データベース名のエントリをコメント・アウトします。Real Application Clusters 以外のデータベースでは、`oratab` ファイル内のこのエントリの形式は次のとおりです。

```
DBSID:oracle_home
```

3. prodnode1 でデータベースを起動します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

4. prodnode1 で次の ASGCTL コマンドを実行して、スタンバイ・ノードにスタンバイ・データベースを作成します。

create standby database コマンドは、standbynode1 に同じ名前の既存のデータベースがある場合、それを上書きします。

```
ASGCTL> connect asg prodnode1 oc4jadmin/<adminpwd>
ASGCTL> set trace on all
```

```
ASGCTL> add instance orcl on vhostdr
```

```
ASGCTL> dump topology
ASGCTL> verify topology
```

```
ASGCTL> set noprompt
ASGCTL> set primary database sys/<passwd>@orcl
```

```
ASGCTL> create standby database orcl on standbynode1
```

```
ASGCTL> verify topology with standbynode1
ASGCTL> instantiate topology to standbynode1
ASGCTL> sync topology to standbynode1
```

## 6.13.3 スイッチオーバー手順

プライマリ・サイトでスケジューリングした停止を行う場合は、ASGCTL の `switchover` コマンドを実行してスタンバイ・サイトにスイッチオーバーします。計画外停止がある場合は、[第 6.13.5 項「フェイルオーバー手順」](#) の説明に従ってフェイルオーバーの手順を実行する必要があります。

prodnode1 で、次のように `switchover` コマンドを実行します。

```
ASGCTL> connect asg oc4jadmin/<adminpwd>
ASGCTL> verify topology with standbynode1
ASGCTL> set primary database sys/<passwd>@orcl
ASGCTL> switchover topology to standbynode1
ASGCTL> disconnect
```

### 6.13.4 スイッチバック手順

スケジューリングした停止が終了したら、スタンバイ・サイトからプライマリ・サイトにスイッチバックします。

standbynode1 で次のコマンドを実行して、プライマリ・サイトにスイッチバックします。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
ASGCTL> verify topology with prodsnodel
ASGCTL> set primary database sys/<passwd>@orcl
ASGCTL> switchover topology to prodsnodel
```

### 6.13.5 フェイルオーバー手順

プライマリ・サイトに予期しない障害が発生した場合は、フェイルオーバーの手順を実行してスタンバイ・サイトにフェイルオーバーします。スケジューリングした停止については、かわりに第 6.13.3 項「スイッチオーバー手順」の手順を実行してください。

スタンバイ・サイトにフェイルオーバーするには、次のコマンドをスタンバイ・サイトで実行して、新しいプライマリとしてアクティブにします。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
ASGCTL> set primary database sys/<passwd>@orcl
ASGCTL> set new primary database sys/<passwd>@orcl
ASGCTL> failover
```

## 6.14 Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery

この項では、OracleAS Metadata Repository で Real Application Clusters データベースを使用する OracleAS Disaster Recovery トポロジの構成方法について説明します。Real Application Clusters データベースは、プライマリ・サイトとスタンバイ・サイトの両方で使用することも、プライマリ・サイトのみで使用する（スタンバイ・サイトでは Real Application Clusters 以外のデータベースを使用する）こともできます。次のサブセクションでは、これらのケースについて説明します。

- 第 6.14.1 項「プライマリ・サイトおよびスタンバイ・サイトの両方で Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の構成」
- 第 6.14.2 項「プライマリ・サイトのみで Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の構成（スタンバイ・サイトでは Real Application Clusters 以外のデータベースを使用）」

### 6.14.1 プライマリ・サイトおよびスタンバイ・サイトの両方で Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の構成

この項では、プライマリ・サイトとスタンバイ・サイトの両方で Oracle Real Application Clusters データベースを使用するトポロジで OracleAS Disaster Recovery を設定する方法について説明します。

この項の項目は次のとおりです。

- 第 6.14.1.1 項「前提条件」
- 第 6.14.1.2 項「構成手順」
- 第 6.14.1.3 項「スイッチオーバー手順」
- 第 6.14.1.4 項「スイッチバック手順（プライマリ・サイトへのスイッチバック）」
- 第 6.14.1.5 項「フェイルオーバー手順」

### 6.14.1.1 前提条件

次の前提条件に注意してください。

- Real Application Clusters ソフトウェアがプライマリ・サイトとスタンバイ・サイトの両方にインストールされているものとします。
- プライマリ・サイトとスタンバイ・サイトのすべての Real Application Clusters ノード上にあるデータベースの Oracle ホームに、OracleAS Guard のスタンドアロン・バージョンをインストールする必要があります。OracleAS Guard のスタンドアロン・バージョンは、Oracle Application Server Companion CD Disk 2 にあります。  
スタンドアロンの OracleAS Guard インストーラの実行方法の詳細は、使用しているプラットフォームの Oracle Application Server のインストール・ガイドで、「高可用性環境へのインストール」の章を参照してください。
- スタンバイ・サイトのデータベースを停止し、その SID を削除する必要があります。これらの手順（第 6.14.1.2 項「構成手順」を参照）は、後の手順で create standby database コマンドを実行する前に実行する必要があります。
- スタンバイ・サイトのデータベースは、create standby database コマンドによって上書きされます。
- 表 6-6 は、次に説明する手順で使用するホスト名とデータベース名を示しています。この手順では、各サイトに 2 つのノードを持つ Real Application Clusters があることが前提となります。

**表 6-6 プライマリ・サイトおよびスタンバイ・サイトのホスト名とデータベース名**

	プライマリ・サイト	スタンバイ・サイト
物理ホスト名	prodnod1、prodnod2	standbynode1、standbynode2
仮想ホスト名	vracnode1、vracnode2	vracnode1、vracnode2
データベース名	orcl.oracle.com	orcl.oracle.com
データベース SID	orcl1 (prodnod1) orcl2 (prodnod2)	orcl1 (standbynode) orcl2 (standbynode2)

- 仮想ホスト名 vracnode1 は、マシンの IP アドレスにマッピングする必要があります。このマッピングは、各マシンの hosts ファイルで設定します。次に、例を示します。

プライマリ・サイトで、%SystemRoot%\system32\drivers\etc\hosts (Windows) または /etc/hosts (UNIX) を編集し、次のようなエントリを追加します。

```
ip_address prodnode1.domain.com prodnode1 vracnode1.domain.com vracnode1
```

スタンバイ・サイトで、同じ hosts ファイルを編集し、次のようなエントリを追加します。

```
ip_address standbynode1.domain.com standbynode1 vracnode1.domain.com vracnode1
```

### 6.14.1.2 構成手順

次の手順を実行して、プライマリ・サイトとスタンバイ・サイトで Real Application Clusters データベースを使用する OracleAS Disaster Recovery トポロジを構成します。

1. スタンバイ・サイトの Real Application Clusters データベースを停止し、CRS を介したデータベースの自動再起動を無効にします。
  - > DBHOME/bin/srvctl stop database -d orcl
  - > DBHOME/bin/srvctl disable database -d orcl

- standbynode1 で、そのデータベースの SID を削除します。SID を削除しないと、create standby database コマンドの実行時にエラーが発生します。

Windows の場合は、次の oradim コマンドを実行して Oracle SID を削除します。

```
> oradim -delete -sid orcl1
```

UNIX の場合は、oratab ファイル内でデータベース SID/ データベース名のエントリをコメント・アウトします。Real Application Clusters データベースでは、oratab ファイル内のこのエントリの形式は次のとおりです。

```
DBUniqueName:oracle_home
```

- prodnode1 で Real Application Clusters データベースを停止して、CRS を介したデータベースの自動再起動を無効にします。

```
> DBHOME/bin/srvctl stop database -d orcl
> DBHOME/bin/srvctl disable database -d orcl
```

- prodnode1 のみでデータベースを起動します。他の Real Application Clusters ノード上のデータベース・インスタンスが停止していることを確認します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

- prodnode1 で次の ASGCTL コマンドを実行して、standbynode1 にデータベースを作成します。

これらのコマンドについては、次の点に注意してください。

- create standby database コマンドは、standbynode1 に同じ名前の既存のデータベースがある場合、それを上書きします。
- UNIX の場合、add instance コマンドは orcl (データベース名) を使用して oratab エントリを検出します。  
Windows の場合は、orcl1 (データベース SID) を使用してレジストリ・エントリを検出します。
- UNIX の場合、set primary database コマンドおよび create standby database コマンドは orcl (データベース名) を使用しますが、Windows の場合、orcl1 (データベース SID) を使用します。

```
ASGCTL> connect asg prodnode1 oc4jadmin/<adminpwd>
ASGCTL> set trace on all
```

**UNIX の場合のみ:** ASGCTL> add instance orcl on vracnode1

**Windows の場合のみ:** ASGCTL> add instance orcl1 on vracnode1

```
ASGCTL> dump topology
```

```
ASGCTL> verify topology
```

```
ASGCTL> set noprompt
```

**UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl

**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

**UNIX の場合のみ:** ASGCTL> create standby database orcl on standbynode1

**Windows の場合のみ:** ASGCTL> create standby database orcl1 on standbynode1

```
ASGCTL> verify topology with standbynode1
```

```
ASGCTL> instantiate topology to standbynode1
```

6. orcl\_remote1 または orcl1\_remote1 エントリを、prodnode1 からプライマリ・サイトの他のノードに伝播します。

- a. prodnode1 の tnsnames.ora にある orcl\_remote1 (UNIX) または orcl1\_remote1 (Windows) のエントリを、プライマリ・サイト上のその他すべての Real Application Clusters ノードにコピーします。

UNIX では、このエントリにデータベース名 (orcl)、Windows では、データベース SID (orcl1) が使用され、それぞれのエントリ名に \_remote<n> が追加されています (<n> は数字)。

場合によっては、<n> の番号が上がり、LOG\_ARCHIVE\_DEST\_<n> パラメータの SERVICE 属性で指定されたエントリ \_remote<n> も同様に伝播する必要があります。

- b. prodnode2 で、CRS を使用してリスナーを再起動します。

```
> CRSHOME/bin/crs_stop ora.prodnode2.LISTENER_PRODNODE2.lsnr
> CRSHOME/bin/crs_start ora.prodnode2.LISTENER_PRODNODE2.lsnr
```

- c. リモート・エントリに指定されたスタンバイ・データベースが TNS を使用して ping 可能であることを確認します。

**UNIX の場合のみ:** > tnsping orcl\_remote1  
**Windows の場合のみ:** > tnsping orcl1\_remote1

7. prodnode2 でデータベースを起動して、spfile を作成します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

**UNIX の場合のみ:** SQL> create spfile='<ORADATASHAREDLOCATION>/orcl/spfileorcl.ora'  
 from pfile='<DBHOME>/dbs/initORCL2.ora';

**Windows の場合のみ:** SQL> create spfile='<ORADATASHAREDLOCATION>\orcl\spfileorcl.ora'  
 from pfile='<DBHOME>/database/initORCL2.ora';

```
SQL> shutdown immediate;
```

すべての Real Application Clusters インスタンスを起動するために、データベースを停止します。

8. プライマリ・サイトのすべての Real Application Clusters データベース・インスタンスを起動するために、prodnode1 で稼動している単一のデータベース・インスタンスを停止します。sync topology コマンドを使用するには、すべてのインスタンスが稼動している必要があります。

- a. prodnode1 でデータベースを停止します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> shutdown immediate;
```

- b. Real Application Clusters データベース・インスタンスを起動します。

```
> srvctl enable database -d orcl
> srvctl start database -d orcl
```

9. prodnode1 で、ASGCTL の sync topology コマンドを実行します。

```
ASGCTL> connect asg oc4jadmin/<adminpwd>
```

**UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl

**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

```
ASGCTL> sync topology to standbynode1
```

### 6.14.1.3 スイッチオーバー手順

この項では、プライマリ・サイトのスケジューリングした停止に対する準備を行うため、ASGCTL の `switchover` コマンドを実行してプライマリ・サイトからスタンバイ・サイトに切り替える方法について説明します。

スケジューリングした停止が終了したら、プライマリ・サイトにスイッチバックできます。詳細は、第 6.14.1.4 項「スイッチバック手順 (プライマリ・サイトへのスイッチバック)」を参照してください。

計画外停止については、かわりに第 6.14.1.5 項「フェイルオーバー手順」の手順を実行してください。

スケジューリングした停止を実行するためにスタンバイ・サイトにスイッチオーバーする手順は次のとおりです。

1. プライマリ・サイトでデータベースを停止して、CRS を介したデータベースの自動再起動を無効にします。これは、データベース・インスタンスを 1 つのみ起動するためです。これは、今後のスイッチオーバー操作で必要になります。

`prodnode1` で次のコマンドを実行します。

```
> srvctl stop database -d orcl
> srvctl disable database -d orcl
```

`prodnode1` で単一のデータベース・インスタンスを起動します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

2. `prodnode1` で `switchover` コマンドを実行します。

```
ASGCTL> connect asg oc4jadmin/<adminpwd>
ASGCTL> verify topology with standbynode1
```

**UNIX の場合のみ:** `ASGCTL> set primary database sys/<passwd>@orcl`  
**Windows の場合のみ:** `ASGCTL> set primary database sys/<passwd>@orcl1`

```
ASGCTL> switchover topology to standbynode1
ASGCTL> disconnect
```

プライマリ・データベースは排他的にマウントする必要があるというメッセージがコンソールに表示される場合は、複数の Real Application Clusters データベース・インスタンスが稼動していることを意味します。データベース・インスタンスを 1 つのみ起動する方法については、前述の手順を参照してください。

3. `standbynode1` で、スイッチオーバー後に `orcl1` インスタンスを停止します。

```
SQL> shutdown immediate;
```

4. `standbynode1` で `initorcl1.ora` ファイル内のパラメータを変更します。

- a. `standbynode1` で、ファイル `DBHOME/dbs/initorcl1.ora` (UNIX) または `DBHOME¥database¥initorcl1.ora` (Windows) のバックアップ・コピーを作成します。このファイルは次の手順で編集します。

- b. `DBHOME/dbs/initorcl1.ora` (UNIX) または `DBHOME¥database¥initorcl1.ora` (Windows) で、パラメータの値が次のように設定されていることを確認します。

```
*.cluster_database_instances=2
*.cluster_database=TRUE
*.remote_listener='LISTENERS_ORCL'
```

- c. `initorcl1.ora` を `standbynode1` から `standbynode2` の対応するディレクトリ (UNIX では `DBHOME/dbs`、Windows では `DBHOME¥database`) にコピーします。

- d. `standbynode2` で、このファイルの名前を `initorcl2.ora` に変更します。

- e. standbynode2 で、`initorc12.ora` ファイルを更新してインスタンス固有のパラメータを修正します。たとえば、次のような行があるとします。
- ```
*.service_names=orc11
*.instance_name=orc11
```
- これを次のように修正します。
- ```
*.service_names=orc12
*.instance_name=orc12
```
5. `orcl_remote1` または `orcl1_remote1` エントリを、`standbynode1` からスタンバイ・サイトの他の Real Application Clusters ノードに伝播します。
- a. `standbynode1` の `tnsnames.ora` にある `orcl_remote1` (UNIX) または `orcl1_remote1` (Windows) のエントリを、スタンバイ・サイト上のその他すべての Real Application Clusters ノードにコピーします。
- UNIX ではこのエントリにデータベース名 (`orc1`)、Windows ではデータベース SID (`orc11`) が使用され、それぞれのエントリ名に `_remote<n>` が追加されています (`<n>` は数字)。
- 場合によっては、`<n>` の番号が上がり、`LOG_ARCHIVE_DEST <n>` パラメータの `SERVICE` 属性で指定されたエントリ `_remote<n>` も同様に伝播する必要があります。
- b. `standbynode2` で、CRS を使用してリスナーを再起動します。
- ```
> CRSHOME/bin/crs_stop ora.standbynode2.LISTENER_STANDBYNODE2.lsnr
> CRSHOME/bin/crs_start ora.standbynode2.LISTENER_STANDBYNODE2.lsnr
```
- c. リモート・エントリに指定されたスタンバイ・データベースが TNS を使用して ping 可能であることを確認します。
- UNIX の場合のみ: `> tnsping orcl_remote1`  
Windows の場合のみ: `> tnsping orcl1_remote1`
6. `standbynode2` でデータベースを起動し、`spfile` を作成して、データベースを停止します。
- ```
SQL> startup;
```
- UNIX の場合のみ: `SQL> create spfile='<ORADATASHAREDLOCATION>/orcl/spfileorcl.ora'`  
`from pfile='<DBHOME>/dbs/initORCL2.ora';`
- Windows の場合のみ: `SQL> create spfile='<ORADATASHAREDLOCATION>%orcl%spfileorcl.ora'`  
`from pfile='<DBHOME>/database/initORCL2.ora';`
- ```
SQL> shutdown immediate;
```
7. `standbynode1` で、CRS を使用してスタンバイ・サイトの Real Application Clusters データベース・インスタンスを再起動します。
- ```
> srvctl enable database -d orcl
> srvctl start database -d orcl
```

#### 6.14.1.4 スイッチバック手順 (プライマリ・サイトへのスイッチバック)

プライマリ・サイトのスケジューリングした停止が終了したら、次の手順を実行してプライマリ・サイトにスイッチバックします。

スイッチバック先のプライマリ・サイトでは、1つのノードのみで Real Application Clusters データベースが起動している必要があることに注意してください。これを行うには、CRS ではなく、SQL\*Plus を使用して手動でデータベースを起動します。この方法は、次の手順 1 で説明します。

これは、sync topology コマンドにおいて、同期対象のデータベース（つまり prodnode1）に対するメディア・リカバリの実行が必要となる場合があるためです。メディア・リカバリは、データベースの排他的なマウントを必要とします。

1. スイッチバック先のプライマリ・サイトの 1 つのノードのみで Real Application Clusters データベースを実行します。ここで説明する手順では、prodnode1 のみでデータベースを実行します。

prodnode1 で Real Application Clusters データベースを停止して、CRS を介したデータベースの自動再起動を無効にします。

```
> DBHOME/bin/srvctl stop database -d orcl
> DBHOME/bin/srvctl disable database -d orcl
```

prodnode1 で、単一のデータベース・インスタンスを起動します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

2. スタンバイ・サイトのデータベースをプライマリ・サイトのデータベースと同期します。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
ASGCTL> verify topology with prodnode1
ASGCTL> set trace on all
ASGCTL> sync topology to prodnode1
```

3. スタンバイ・サイトですべてのデータベース・インスタンスを停止し、データベースの自動再起動を無効にした後、standbynode1 上のデータベースのみを起動します。

```
> srvctl stop database -d orcl
> srvctl disable database -d orcl
```

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

4. standbynode1 で、switchover コマンドを実行します。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
ASGCTL> verify topology with prodnode1
```

**UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl  
**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

```
ASGCTL> switchover topology to prodnode1
ASGCTL> disconnect
```

5. prodnode1 で、単一のデータベース・インスタンスを停止します。CRS を使用してすべての Real Application Clusters インスタンスを起動し、sync topology を実行します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> shutdown immediate;
```

```
> DBHOME/bin/srvctl enable database -d ORCL
> DBHOME/bin/srvctl start database -d ORCL
```

6. sync topology コマンドを実行します。

```
ASGCTL> connect asg prodnode1 oc4jadmin/<adminpwd>
```

**UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl  
**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

```
ASGCTL> sync topology to standbynode1
ASGCTL> disconnect
```

### 6.14.1.5 フェイルオーバー手順

この項では、スタンバイ・サイトへのフェイルオーバー手順について説明します。ここで説明する手順は、プライマリ・サイトの計画外停止に対して使用してください。スケジューリングした停止については、第 6.14.1.3 項「スイッチオーバー手順」の手順を参照してください。

1. スタンバイ・サイトで failover コマンドを実行し、新しいプライマリ・サイトとしてアクティブにします。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
```

```
UNIX の場合のみ: ASGCTL> set primary database sys/<passwd>@orcl
```

```
UNIX の場合のみ: ASGCTL> set new primary database sys/<passwd>@orcl
```

```
Windows の場合のみ: ASGCTL> set primary database sys/<passwd>@orcl1
```

```
Windows の場合のみ: ASGCTL> set new primary database sys/<passwd>@orcl1
```

```
ASGCTL> set trace on all
```

```
ASGCTL> failover
```

```
ASGCTL> disconnect
```

2. スタンバイ・サイトで、CRS を使用してデータベースを停止し、起動します。

```
> DBHOME¥bin¥sqlplus / as sysdba
SQL> shutdown immediate;
```

```
> srvctl enable database -d rac
```

```
> srvctl start database -d rac
```

## 6.14.2 プライマリ・サイトのみで Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の構成（スタンバイ・サイトでは Real Application Clusters 以外のデータベースを使用）

この項では、プライマリ・サイトのみで Oracle Real Application Clusters データベースを使用する OracleAS Disaster Recovery の設定方法について説明します。スタンバイ・サイトでは、標準の Oracle データベースを使用します。

この項の項目は次のとおりです。

- 第 6.14.2.1 項「前提条件」
- 第 6.14.2.2 項「構成手順」
- 第 6.14.2.3 項「スイッチオーバー手順」
- 第 6.14.2.4 項「スイッチバック手順」
- 第 6.14.2.5 項「フェイルオーバー手順」

### 6.14.2.1 前提条件

次の前提条件に注意してください。

- Real Application Clusters ソフトウェアがプライマリ・サイトにインストールされているものとします。
- プライマリ・サイトのすべての Real Application Clusters ノード上にあるデータベースの Oracle ホーム、およびスタンバイ・サイトのデータベースの Oracle ホームに OracleAS Guard のスタンドアロン・バージョンをインストールする必要があります。OracleAS Guard のスタンドアロン・バージョンは、Oracle Application Server Companion CD Disk 2 にあります。

スタンドアロンの OracleAS Guard インストーラの実行方法の詳細は、使用しているプラットフォームの Oracle Application Server のインストールレーション・ガイドで、「高可用性環境へのインストール」の章を参照してください。

- スタンバイ・サイトのデータベースを停止し、その SID を削除する必要があります。これらの手順（第 6.14.2.2 項「構成手順」を参照）は、後の手順で `create standby database` コマンドを実行する前に実行する必要があります。
- スタンバイ・サイトのデータベースは、`create standby database` コマンドによって上書きされます。
- 表 6-7 は、次に説明する手順で使用するホスト名とデータベース名を示しています。この手順では、プライマリ・サイトに 2 つのノードを持つ Real Application Clusters があることが前提となります。

表 6-7 プライマリ・サイトおよびスタンバイ・サイトのホスト名とデータベース名

	プライマリ・サイト	スタンバイ・サイト
物理ホスト名	prodnode1、prodnode2	standbynode1
仮想ホスト名	vracnode1、vracnode2	vracnode1
データベース名	orcl.oracle.com	orcl.oracle.com
データベース SID	orcl1 (prodnode1) orcl2 (prodnode2)	orcl1 (standbynode)

- 仮想ホスト名 `vracnode1` は、マシンの IP アドレスにマッピングする必要があります。このマッピングは、各マシンの `hosts` ファイルで設定します。次に、例を示します。

プライマリ・サイトで、`%SystemRoot%\system32\drivers\etc\hosts` (Windows) または `/etc/hosts` (UNIX) を編集し、次のようなエントリを追加します。

```
ip_address prodnode1.domain.com prodnode1 vracnode1.domain.com vracnode1
```

スタンバイ・サイトで、同じ `hosts` ファイルを編集し、次のようなエントリを追加します。

```
ip_address standbynode1.domain.com standbynode1 vracnode1.domain.com vracnode1
```

### 6.14.2.2 構成手順

次の手順を実行して、プライマリ・サイトでは Real Application Clusters データベースを使用し、スタンバイ・サイトでは Real Application Clusters 以外のデータベースを使用する OracleAS Disaster Recovery トポロジを構成します。

1. スタンバイ・サイトの Real Application Clusters データベースを停止します。

```
> DBHOME/bin/srvctl stop database -d orcl
```

2. `standbynode1` で、そのデータベースの SID を削除します。SID を削除しないと、`create standby database` コマンドの実行時にエラーが発生します。

Windows の場合は、次の `oradim` コマンドを実行して Oracle SID を削除します。

```
> oradim -delete -sid orcl1
```

UNIX の場合は、`oratab` ファイル内でデータベース SID/ データベース名のエントリをコメント・アウトします。Real Application Clusters 以外のデータベースでは、`oratab` ファイル内のこのエントリの形式は次のとおりです。

```
DBSID:oracle_home
```

3. `prodnode1` で Real Application Clusters データベースを停止して、CRS を介したデータベースの自動再起動を無効にします。

```
> DBHOME/bin/srvctl stop database -d orcl
```

```
> DBHOME/bin/srvctl disable database -d orcl
```

4. prodnode1 のみでデータベースを起動します。他の Real Application Clusters ノード上のデータベース・インスタンスが停止していることを確認します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

5. prodnode1 で次の ASGCTL コマンドを実行して、standbynode1 にデータベースを作成します。

これらのコマンドについては、次の点に注意してください。

- create standby database コマンドは、standbynode1 に同じ名前の既存のデータベースがある場合、それを上書きします。
- UNIX の場合、add instance コマンドは orcl (データベース名) を使用して oratab エントリを検出します。

Windows の場合、orcl1 (データベース SID) を使用してレジストリ・エントリを検出します。

- UNIX の場合、set primary database コマンドおよび create standby database コマンドは orcl (データベース名) を使用しますが、Windows の場合、orcl1 (データベース SID) を使用します。

```
ASGCTL> connect asg prodnode1 oc4jadmin/<adminpwd>
ASGCTL> set trace on all
```

UNIX の場合のみ: ASGCTL> add instance orcl on vracnode1

Windows の場合のみ: ASGCTL> add instance orcl1 on vracnode1

```
ASGCTL> dump topology
```

```
ASGCTL> verify topology
```

```
ASGCTL> set noprompt
```

UNIX の場合のみ: ASGCTL> set primary database sys/<passwd>@orcl

Windows の場合のみ: ASGCTL> set primary database sys/<passwd>@orcl1

UNIX の場合のみ: ASGCTL> create standby database orcl on standbynode1

Windows の場合のみ: ASGCTL> create standby database orcl1 on standbynode1

```
ASGCTL> verify topology with standbynode1
```

```
ASGCTL> instantiate topology to standbynode1
```

6. orcl\_remote1 または orcl1\_remote1 エントリを、prodnode1 からプライマリ・サイトの他のノードに伝播します。

- a. prodnode1 の tnsnames.ora にある orcl\_remote1 (UNIX) または orcl1\_remote1 (Windows) のエントリを、プライマリ・サイト上のその他すべての Real Application Clusters ノードにコピーします。

UNIX では、このエントリにデータベース名 (orcl)、Windows では、データベース SID (orcl1) が使用され、それぞれのエントリ名に \_remote<n> が追加されています (<n> は数字)。

場合によっては、<n> の番号が上がり、LOG\_ARCHIVE\_DEST <n> パラメータの SERVICE 属性で指定されたエントリ \_remote<n> も同様に伝播する必要があります。

- b. prodnode2 で、CRS を使用してリスナーを再起動します。

```
> CRSHOME/bin/crs_stop ora.prodnode2.LISTENER_PRODNODE2.lsnr
```

```
> CRSHOME/bin/crs_start ora.prodnode2.LISTENER_PRODNODE2.lsnr
```

- c. リモート・エントリに指定されたスタンバイ・データベースが TNS を使用して ping 可能であることを確認します。
- UNIX の場合のみ:** > tnsping orcl\_remotel  
**Windows の場合のみ:** > tnsping orcl1\_remotel
7. プライマリ・サイトのすべての Real Application Clusters データベース・インスタンスを起動するために、prodnode1 で稼動している単一のデータベース・インスタンスを停止します。sync topology コマンドを使用するには、すべてのインスタンスが稼動している必要があります。
- a. prodnode1 でデータベースを停止します。
- ```
> DBHOME/bin/sqlplus / as sysdba
SQL> shutdown immediate;
```
- b. Real Application Clusters データベース・インスタンスを起動します。
- ```
> srvctl enable database -d orcl
> srvctl start database -d orcl
```
8. prodnode1 で、ASGCTL の sync topology コマンドを実行します。
- ```
ASGCTL> connect asg oc4jadmin/<adminpwd>
```
- UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl  
**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1
- ```
ASGCTL> sync topology to standbynode1
```

### 6.14.2.3 スイッチオーバー手順

この項では、プライマリ・サイトのスケジューリングした停止に対する準備を行うため、ASGCTL の switchover コマンドを実行してプライマリ・サイトからスタンバイ・サイトに切り替える方法について説明します。

スケジューリングした停止が終了したら、プライマリ・サイトにスイッチバックできます。詳細は、第 6.14.2.4 項「スイッチバック手順」を参照してください。

計画外停止については、かわりに第 6.14.2.5 項「フェイルオーバー手順」の手順を実行してください。

スケジューリングした停止を実行するためにスタンバイ・サイトにスイッチオーバーする手順は次のとおりです。

1. プライマリ・サイトでデータベースを停止して、CRS を介したデータベースの自動再起動を無効にします。これは、データベース・インスタンスを 1 つのみ起動するためです。これは、今後のスイッチオーバー操作で必要になります。
- prodnode1 で次のコマンドを実行します。
- ```
> srvctl stop database -d orcl
> srvctl disable database -d orcl
```
- prodnode1 で単一のデータベース・インスタンスを起動します。
- ```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```
2. prodnode1 で switchover コマンドを実行します。
- ```
ASGCTL> connect asg oc4jadmin/<adminpwd>
ASGCTL> verify topology with standbynode1
```
- UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl  
**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

```
ASGCTL> switchover topology to standbynode1
ASGCTL> disconnect
```

プライマリ・データベースは排他的にマウントする必要があるというメッセージがコンソールに表示される場合は、複数の Real Application Clusters データベース・インスタンスが稼働していることを意味します。データベース・インスタンスを1つのみ起動する方法については、前述の手順を参照してください。

#### 6.14.2.4 スイッチバック手順

この項では、スケジューリングした停止が終了したときにプライマリ・サイトにスイッチバックする手順について説明します。

スイッチバック先のプライマリ・サイトでは、1つのノードのみで Real Application Clusters データベースが起動している必要があることに注意してください。これを行うには、CRS ではなく、SQL\*Plus を使用して手動でデータベースを起動します。この方法は、次の手順 1 で説明します。

これは、sync topology コマンドにおいて、同期対象のデータベース（つまり prodnode1）に対するメディア・リカバリの実行が必要となる場合があるためです。メディア・リカバリは、データベースの排他的なマウントを必要とします。

1. スイッチバック先のプライマリ・サイトの1つのノードのみで Real Application Clusters データベースを実行します。ここで説明する手順では、prodnode1 のみでデータベースを実行します。

prodnode1 で Real Application Clusters データベースを停止して、CRS を介したデータベースの自動再起動を無効にします。

```
> DBHOME/bin/srvctl stop database -d orcl
> DBHOME/bin/srvctl disable database -d orcl
```

prodnode1 で、単一のデータベース・インスタンスを起動します。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> startup
```

2. standbynode1 で次のコマンドを実行して、プライマリ・サイトにスイッチバックします。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
ASGCTL> verify topology with prodnode1
```

**UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl

**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

```
ASGCTL> switchover topology to prodnode1
```

3. prodnode1 で、CRS を使用して単一のデータベース・インスタンスを停止し、すべての Real Application Clusters インスタンスを起動します。sync topology コマンドを使用するには、すべてのインスタンスが稼働している必要があります。

```
> DBHOME/bin/sqlplus / as sysdba
SQL> shutdown immediate;
```

```
> DBHOME/bin/srvctl enable database -d ORCL
> DBHOME/bin/srvctl start database -d ORCL
```

4. sync topology コマンドを実行します。

```
ASGCTL> connect asg prodnode1 oc4jadmin/<adminpwd>
```

**UNIX の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl

**Windows の場合のみ:** ASGCTL> set primary database sys/<passwd>@orcl1

```
ASGCTL> sync topology to standbynode1
ASGCTL> disconnect
```

### 6.14.2.5 フェイルオーバー手順

この項では、スタンバイ・サイトへのフェイルオーバー手順について説明します。ここで説明する手順は、プライマリ・サイトの計画外停止に対して使用してください。スケジューリングした停止については、第 6.14.2.3 項「スイッチオーバー手順」の手順を参照してください。

スタンバイ・サイトで failover コマンドを実行し、新しいプライマリ・サイトとしてアクティブにします。

```
ASGCTL> connect asg standbynode1 oc4jadmin/<adminpwd>
```

```
UNIX の場合のみ: ASGCTL> set primary database sys/<passwd>@orcl
```

```
UNIX の場合のみ: ASGCTL> set new primary database sys/<passwd>@orcl
```

```
Windows の場合のみ: ASGCTL> set primary database sys/<passwd>@orcl1
```

```
Windows の場合のみ: ASGCTL> set new primary database sys/<passwd>@orcl1
```

```
ASGCTL> set trace on all
```

```
ASGCTL> failover
```

```
ASGCTL> disconnect
```

## 6.15 OracleAS Guard 操作の監視とトラブルシューティング

OracleAS Disaster Recovery ソリューションを設定し、スタンバイ・トポロジをインスタンス化し、スタンバイ・トポロジを同期化したら、OracleAS Guard クライアントのコマンドライン・ユーティリティの asgctl を使用して、OracleAS Guard 調整サーバーから asgctl 操作の監視およびトラブルシューティング・タスクを実行するコマンドを発行できます。通常の OracleAS Guard の監視またはトラブルシューティング・セッションでは、次のようなタスクを実行します。

1. 第 6.15.1 項「トポロジの検証」
2. 第 6.15.2 項「現在の操作の表示」
3. 第 6.15.3 項「完了した操作のリストの表示」
4. 第 6.15.4 項「オペレーションの停止」
5. 第 6.15.5 項「タスクのトレース」
6. 第 6.15.6 項「トポロジに関する情報のファイルへの書込み」

OracleAS Guard クライアントから asgctl コマンドを発行すると、OracleAS Guard 調整サーバーにリクエストが送られます。OracleAS Guard 調整サーバーはこのリクエストを本番トポロジとスタンバイ・トポロジの OracleAS Guard サーバーに転送します。その後、ステータス・メッセージが OracleAS Guard クライアントに返され、特定のタスクに問題が発生した場合はエラー・メッセージも返されます。エラー・メッセージの詳細は、第 6.15.7 項「エラー・メッセージ」を参照してください。

### 6.15.1 トポロジの検証

プライマリ・トポロジが実行状態にあり、構成が有効であることを確認するには、asgctl プロンプトで次の asgctl コマンドを入力します。

```
ASGCTL> connect asg ias_admin/iastest2
Successfully connected to prodinfra:7890
ASGCTL> discover topology oidpassword=oidpwd
ASGCTL> verify topology
Generating default policy for this operation
prodinfra:7890
  HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
  HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
  HA directory exists for instance asmid1.asmid1.us.oracle.com
ASGCTL>
```

ポリシー・ファイルを使用する場合は、検証ポリシー・ファイル (verify\_policy.xml) を編集して使用し、ファイル内に各インスタンスの成功要件の属性を指定します。verify コマンドに using policy <file> パラメータを指定します。ここで <file> は、verify\_policy.xml ファイルのパスとファイルの指定を表します。詳細は、第 6.7 項「いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照してください。

ローカル・ホストがメンバーであるプライマリ・トポロジとスタンバイ・トポロジを比較して、両方のトポロジが一致していること、および両方のトポロジが OracleAS Disaster Recovery の要件を満たしていることを確認するには、asgctl プロンプトで次の asgctl コマンドを入力して、スタンバイ・ホスト・システムの名前を指定します。

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory "/private1/OraHome2/asr1012/dsa/conf/"

ASGCTL> verify topology with standbyinfra
Generating default policy for this operation
prodinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
prodinfra:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
ASGCTL>

# Command to use if you want to use a policy file
# verify topology with standbyinfra using policy <file>
```

## 6.15.2 現在の操作の表示

OracleAS Guard クライアントが接続しているトポロジのすべてのノードで、現在実行されているすべての操作のステータスを表示するには、asgctl プロンプトで次の asgctl コマンドを入力します。

```
ASGCTL> show operation
*****
OPERATION: 19
Status: running
Elapsed Time: 0 days, 0 hours, 0 minutes, 28 secs
TASK: syncFarm
    TASK: backupFarm
        TASK: fileCopyRemote
        TASK: fileCopyRemote
    TASK: restoreFarm
        TASK: fileCopyLocal
```

### 6.15.3 完了した操作のリストの表示

完了した操作（現行セッションで OracleAS Guard クライアントが接続されているトポロジーのノード上で実行されていない操作）のみを表示するには、`asgctl` プロンプトで次の `asgctl` コマンドを入力します。

```
ASGCTL> show operation history
*****
OPERATION: 7
  Status: success
  Elapsed Time: 0 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
*****
OPERATION: 16
  Status: success
  Elapsed Time: 0 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
*****
OPERATION: 19
  Status: success
  Elapsed Time: 0 days, 0 hours, 1 minutes, 55 secs
  TASK: syncFarm
    TASK: backupFarm
      TASK: fileCopyRemote
      TASK: fileCopyRemote
    TASK: restoreFarm
      TASK: fileCopyLocall
```

### 6.15.4 オペレーションの停止

サーバーで実行されている特定の操作を停止するには、`asgctl` プロンプトで次の `asgctl` コマンドを入力し、停止する操作番号を指定します。停止する操作番号は、`asgctl show operation full` コマンドを入力すると表示されます。

```
ASGCTL> show operation full
*****
OPERATION: 19
  Status: running
  Elapsed Time: 0 days, 0 hours, 0 minutes, 28 secs
  Status: running
  .
  .
  .
ASGCTL> stop operation 19
```

### 6.15.5 タスクのトレース

特定のイベントにトレース・フラグを設定し、出力を `asgctl` ログ・ファイルに記録するには、`asgctl` プロンプトで次の `asgctl` コマンドを入力し、`on` キーワードを指定して、有効にするトレース・フラグを入力します。この場合、トレース・フラグ DB は、Oracle Database 環境の処理に関するトレース情報が表示されることを示します。有効にすることができるその他のトレース・フラグの詳細は、「[set trace](#)」コマンドを参照してください。設定できるすべてのトレース・フラグの一覧も、「[set trace](#)」コマンドを参照してください。

```
ASGCTL> set trace on db
```

## 6.15.6 トポロジに関する情報のファイルへの書込み

ローカル・ホストが接続されているトポロジに関する詳細情報を書き込むには、`asgctl` プロンプトで次の `asgctl` コマンドを入力し、出力される情報を書き込むファイルのパスと名前を指定します。この出力では、`dump topology` コマンドを実行した場合と同じ結果が表示されますが、この場合はファイルに書き込まれるので、将来参照するために保存しておくことが可能です。

```
ASGCTL> dump topology to c:\%dump_mid_1.txt
```

## 6.15.7 エラー・メッセージ

OracleAS Disaster Recovery ソリューションの実行時に表示されるエラー・メッセージは、付録 B「OracleAS Guard エラー・メッセージ」に分類して説明しています。

## 6.16 ワイド・エリア DNS の操作

クライアント・リクエストが本番サイトのエントリ・ポイントに送信されるようにするには、DNS 解決を使用します。サイトのスイッチオーバーまたはフェイルオーバーが実行される際には、クライアント・リクエストが本番の役割を果たす新しいサイトに透過的にリダイレクトされる必要があります。このリダイレクションを可能にするには、本番サイトへのリクエストを解決するワイド・エリア DNS をスタンバイ・サイトにスイッチオーバーする必要があります。DNS スイッチオーバーは、ワイド・エリア・ロード・バランサを使用するか手動で DNS 名を変更することによって行えます。

---

**注意：** ハードウェア・ロード・バランサは、各サイトのフロントエンドにあると想定します。サポートされているロード・バランサについては、<http://metalink.oracle.com> を参照してください。

---

次のサブセクションでは DNS スイッチオーバー操作について説明します。

### 6.16.1 ワイド・エリア・ロード・バランサの使用

ワイド・エリア・ロード・バランサ（グローバル・トラフィック・マネージャ）が本番サイトとスタンバイ・サイトの前面に配置されているときは、両サイトに対して、障害検出サービスとパフォーマンススペースのルーティング・リダイレクションが提供されます。さらに、このロード・バランサは、信頼できる DNS ネーム・サーバーと同等の機能を提供できます。

通常の運用時は、本番サイトのロード・バランサの名前と IP の対応付けにより、ワイド・エリア・ロード・バランサを構成できます。DNS スイッチオーバーが必要な際には、ワイド・エリア・ロード・バランサのこの対応付けが変更され、スタンバイ・サイトのロード・バランサの IP に対応付けられます。これにより、本番の役割を引き継いだスタンバイ・サイトにリクエストが転送されるようになります。

DNS スイッチオーバーのこの方法は、サイトのスイッチオーバーとフェイルオーバーの両方に対応できます。ワイド・エリア・ロード・バランサを使用する利点の 1 つは、名前と IP の新しい対応付けがすぐに有効になるという点です。不利な点は、ワイド・エリア・ロード・バランサへの追加投資が必要になることです。

### 6.16.2 DNS 名の手動変更

DNS スイッチオーバーのこの方法には、元は本番サイトのロード・バランサの IP アドレスに対応付けられていた名前と IP の対応付けを手動で変更することが関係しています。この対応付けを変更して、スタンバイ・サイトのロード・バランサの IP アドレスに対応付けます。スイッチオーバーを実行する手順は次のとおりです。

1. 本番サイトのロード・バランサの対応付けに対する現行の TTL 値を書き留めます。この対応付けは DNS キャッシュにあり、TTL が終了するまで保持されます。例として、TTL が 3600 秒であると想定します。

2. TTL 値をより短い時間 (60 秒など) に変更します。
3. 元の TTL 間隔が 1 回経過するまで待ちます。これは、手順 1 で書き留めた元の TTL 値である 3600 秒です。
4. スタンバイ・サイトがリクエストを受信するようにスイッチオーバーされていることを確認します。
5. DNS の対応付けを変更して、スタンバイ・サイトのロード・バランサ用に設定します。また、通常の運用に適切な TTL 値 (たとえば、3600 秒) を指定します。

DNS スwitchオーバーのこの方法は、計画的なサイトのスイッチオーバー操作にのみ対応できます。手順 2 の TTL 値は、クライアント・リクエストを処理しきれない時間に設定する必要があります。TTL の変更とは、実質的にはアドレス解決時間を短縮させるためにキャッシュ・セマンティックを変更することを表しているからです。そのため、キャッシュ時間が短縮されたことにより、DNS リクエストが増加する可能性があります。

## 6.17 OracleAS Guard コマンドライン・ユーティリティ (asgctl) の使用

この項は、次のサブセクションで構成されています。

- 第 6.17.1 項「asgctl を使用する一般的な OracleAS Guard セッション」
- 第 6.17.2 項「OracleAS Guard asgctl スクリプトの定期的なスケジュール」
- 第 6.17.3 項「Enterprise Manager Job System を使用した OracleAS Guard ジョブの実行」

### 6.17.1 asgctl を使用する一般的な OracleAS Guard セッション

asgctl を使用した一般的な OracleAS Guard セッションには、次のようなタスクがあります。これらのタスクについては、次のサブセクションで説明します。

- 第 6.17.1.1 項「ヘルプの参照」
- 第 6.17.1.2 項「プライマリ・データベースの指定」
- 第 6.17.1.3 項「トポロジーの検出」

asgctl コマンドライン・インタフェースがサポートされている利点の 1 つは、これらの asgctl コマンドを、第 6.17.1.4 項「asgctl スクリプトの作成と実行」の説明に従って適切な順序でスクリプトに入力し、第 6.17.2 項「OracleAS Guard asgctl スクリプトの定期的なスケジュール」および第 6.17.3 項「Enterprise Manager Job System を使用した OracleAS Guard ジョブの実行」に従ってスクリプトとして実行できることです。

#### 6.17.1.1 ヘルプの参照

特定のコマンドのヘルプを表示するには、asgctl プロンプトに asgctl コマンドを入力し、ヘルプ情報を表示するコマンド名を指定します。また、すべてのコマンドのヘルプを表示するには、asgctl プロンプトで次の asgctl コマンドを入力します。

```
ASGCTL> help
  connect asg [<host>] [<ias_administrator_account>/<password>]
  disconnect
  exit
  quit
  add instance <instance_name> on <instance_host> [to topology]
  clone topology to <standby_topology_host> [using policy <file>] [no standby]
  clone instance <instance> to <standby_topology_host> [no standby]
  discover topology [oidhost=<host>] [oidsslport=<sslport>] [oiduser=<user>]
  oidpassword=<pass>
  discover topology within farm
  dump farm [to <file>] (Deprecated)
  dump topology [to <file>] [using policy <file>]
  dump policies
  failover [using policy <file>]
```

```

help [<command>]
instantiate farm to <standby_farm_host> (Deprecated)
instantiate topology to <standby_topology_host> [using policy <file>]
remove instance <instance_name> [from topology]
set asg credentials <host> <ias_administrator_account>/<password> [for topology]
set asg credentials <host> ias_admin/<password> [for farm] (Deprecated)
set primary database <username>/<password>@<servicename> [pfile <filename> | spfile
<filename>]
set new primary database <username>/<password>@<servicename> [pfile <filename> | spfile
<filename>]
set noprompt
set trace on|off <traceflags>
sync farm to <standby_farm_host> [full | incr[emental]] (Deprecated)
sync topology to <standby_topology_host> [full | incr[emental]] [using policy <file>]
startup [asg]
startup farm (Deprecated)
startup topology
shutdown [local]
shutdown farm (Deprecated)
shutdown topology
show op[eration] [full] [[his]tory]
show env
stop op[eration] <op#>
switchover farm to <standby_farm_host> (Deprecated)
switchover topology to <standby_topology_host> [using policy <file>]
verify farm [with <host>] (Deprecated)
verify topology [with <host>] [using policy <file>]
ASGCTL>

```

### 6.17.1.2 プライマリ・データベースの指定

プライマリ・トポロジの OracleAS Infrastructure データベースを指定するには、asgctl プロンプトで次の asgctl コマンドを入力し、OracleAS Infrastructure データベースにアクセスする sysdba 権限があるデータベース・アカウントのユーザー名とパスワード、および OracleAS Infrastructure データベースの TNS サービス名を指定します。

```

ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>

```

スタンバイ・サイトの場合も、プライマリ・データベースで指定する値と同じ値を使用します。これは、プライマリとスタンバイの OracleAS Infrastructure データベースのサービス名とパスワードを同じにする必要があるためです。プライマリ・データベースは、インスタンス化、同期化、スイッチオーバーまたはフェイルオーバー操作を実行する前に設定する必要があります。

トポロジに OracleAS Metadata Repository が複数ある場合は、set primary database コマンドを使用して、それぞれを認証する必要があります。

### 6.17.1.3 トポロジの検出

トポロジ XML ファイルを内部的に作成するために、OracleAS Disaster Recovery 環境を最初に設定するとき、discover topology コマンドを実行する必要があります。したがって、本番サイトに別の Oracle ホームを配置した場合や、スイッチオーバーまたはフェイルオーバー操作によって本番サイトからスタンバイ・サイトに役割を変更した場合は、常にトポロジの検出操作を実行する必要があります。discover topology コマンドは、Oracle Internet Directory に、本番サイトの同じ Oracle Internet Directory を共有する、トポロジ内のすべてのインスタンスについて問い合わせます。トポロジ・ファイルの詳細は、第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。asgctl プロンプトで次の asgctl コマンドを入力し、トポロジを検出します。

```

ASGCTL> discover topology oidpassword=oidpwd
Discovering topology on host "infra" with IP address "123.1.2.111" prodinfra:7890
Connecting to the OID server on host "infra.us.oracle.com" using SSL port "636" and
username "orcladmin"

```

```

Getting the list of databases from OID
Gathering database information for SID "asdb" from host "infra.us.oracle.com"
Getting the list of instances from OID
Gathering instance information for "asr1012.infra.us.oracle.com" from host
"infra.us.oracle.com"
Gathering instance information for "asmid1.asmid1.us.oracle.com" from host
"asmid1.us.oracle.com"
Gathering instance information for "asmid2.asmid2.us.oracle.com" from host
"asmid2.us.oracle.com"
The topology has been discovered. A topology.xml file has been written to each home in
the topology.

ASGCTL>

```

本番トポロジが本番サイトの OracleAS Guard に認識された後、後続のコマンドのいずれかを実行し、スタンバイ・サイトに関連する後続の asgctl 操作を実行できます。詳細は、[discover topology](#) を参照してください。

#### 6.17.1.4 asgctl スクリプトの作成と実行

一連の asgctl コマンド名と引数を含むスクリプトを作成するには、適当なエディタを開いて、実行する操作に応じた正しい順序で asgctl コマンドを入力します。スクリプト・ファイルを保存し、asgctl を起動するときに次のようにスクリプトを実行します。

```
> ASGCTL @myasgctlscript.txt
```

一連の asgctl コマンドを指定したスクリプトの例は、「[set echo](#)」コマンドを参照してください。

asgctl スクリプトのコマンドの実行に使用するために、後ですべての対話型プロンプトが無視される非プロンプト状態を設定できます。詳細は、asgctl の「[set noprompt](#)」コマンドを参照してください。

### 6.17.2 OracleAS Guard asgctl スクリプトの定期的なスケジュール

定期的なトポロジの同期化操作を実行してスタンバイ・トポロジとプライマリ・トポロジの同期化を維持するなど、OracleAS Guard 操作を定期的に行うために、OracleAS Guard の asgctl スクリプトの定期的な実行を自動化できます。

UNIX システムの場合は、cron ジョブを設定して asgctl スクリプトを実行できます。asgctl スクリプトを /etc の下の適切なサブディレクトリ、cron.hourly、cron.daily、cron.weekly または cron.monthly にコピーします。スクリプトのコピー先に選択したディレクトリの名前に応じて、スクリプトは時間、日、週または月単位で実行されます。または、crontab を編集して、asgctl スクリプトを実行する時間を指定したエントリを作成することもできます。詳細は、cron および crontab に関するマニュアルのページを参照してください。

Windows システムの場合は、「コントロール パネル」のタスク・スケジューラまたはスケジューラされたタスクを使用して、asgctl スクリプトを毎日、毎週、または毎月実行したり、特定の時間に実行したりできます。また、様々なオプションの付いたサード・パーティ製のスケジューラ・ソフトウェアを購入して、asgctl スクリプトを実行する時間や頻度を設定することもできます。詳細は、Windows オペレーティング・システムのヘルプを参照してください。

### 6.17.3 Enterprise Manager Job System を使用した OracleAS Guard ジョブの実行

Enterprise Manager Job System を使用して、asgctl スクリプトを特定の時間間隔または特定の日時（あるいはその両方）に実行するように自動化したり、さらに別のカスタム設定に設定したりすることができます。その場合は、Enterprise Manager の「[ジョブ・アクティビティ](#)」ページにアクセスして、asgctl スクリプトを実行するユーザー自身のホスト・コマンド・ジョブ、つまりジョブ・タスクを作成します。ジョブ・タスク（スクリプト）は asgctl を起動し、指定された順序に従って asgctl コマンドを実行します。OracleAS Guard ジョブを作成したら、EM ジョブ・ライブラリに保存します。このライブラリは、頻繁に使用されるジョブのリポジトリであり、ここで、カスタム設定と選択した時間に従ってジョブが実行されます。詳細は、Enterprise Manager のオンライン・ヘルプ、および『Oracle Enterprise Manager 概要』を参照してください。

## 6.18 いくつかの OracleAS Metadata Repository 構成の特別な考慮事項

この項では、OracleAS Metadata Repository Creation Assistant を使用して作成した複数の OracleAS Metadata Repository と OracleAS Metadata Repository の特別な考慮事項について説明します。

### 6.18.1 複数の OracleAS Metadata Repository 構成の特別な考慮事項

デフォルトでは、`asgctl connect` コマンドで指定した資格証明は、OracleAS Guard サーバーが他の OracleAS Guard サーバーに接続する際に必ず使用されます。しかし、次のような使い方が必要になる場合もあります。

- サイトのシステムごとに異なる資格証明を使用する場合。詳細は、第 6.18.1.1 項「[asgctl 資格証明の設定](#)」を参照してください。
- プライマリ・トポロジで使用されているものと同じ資格証明の共通のセットをスタンバイ・トポロジで使用する場合。詳細は、第 6.18.1.2 項「[プライマリ・データベースの指定](#)」を参照してください。

ホスト・システムの資格証明が `asgctl connect` コマンドで使用されているものと異なる場合は、OracleAS Guard サーバーが構成内の各ホスト・システムに接続できるように、OracleAS Guard 資格証明を設定する必要があります。

#### 6.18.1.1 asgctl 資格証明の設定

同じトポロジに属する、すべてのホスト・システムに異なる資格証明を設定するには、`asgctl` プロンプトに次の `asgctl` コマンドを入力します。資格証明が適用されるホスト・システムのノード名、Oracle Application Server のインストール時に作成した `ias_admin` アカウント名と `ias_admin` アカウントのパスワード、およびキーワードの `for topology` を指定します。

---

**注意：** OracleAS 10.1.3 インストールの場合、ユーザー名は `oc4jadmin` を使用し、この `oc4jadmin` アカウントのパスワードは Oracle Application Server 10.1.3 のインストール時に作成されたものを使用する必要があります。

---

これらの設定は、現行セッションに有効です。

```
ASGCTL> set asg credentials standbyinfra ias_admin/<iasadminpwd> for topology
```

キーワードの `for topology` を指定した場合は、資格証明は、指定したシステムと同じトポロジに属するすべてのホスト・システムに設定されます。キーワードを指定しない場合、資格証明は指定したホスト・システムにのみ適用されます。

`set asg credentials` コマンドは、トポロジ内の特定のサーバーに異なる資格証明を適用する場合にも使用できます。前述の例では、スタンバイ・トポロジ内のすべてのノードに同じ資格証明を適用し、プライマリ・トポロジに使用する資格証明とは異なる資格証明を設定しました。次のコマンドでは、スタンバイ・トポロジ内の特定のノードである `standbyinfra` ノードに、資格証明を設定します。

```
ASGCTL> set asg credentials standbyinfra ias_admin/<iasadminpwd>
```

つまり、あるトポロジに資格証明を設定した場合、この資格証明はトポロジ全体に継承されます。トポロジ上の個別のホストに資格証明を設定した場合、(そのホストの) 資格証明が、トポロジに設定されたデフォルトの資格証明よりも優先されます。

また、Oracle Internet Directory と OracleAS Metadata Repository が同じ場所にあつて、Portal の OracleAS Metadata Repository が別の場所にあるような、複数の Infrastructure が存在するトポロジでは、OracleAS Guard でインスタンス化、同期化、スイッチオーバー、フェイルオーバーなどの重要な操作を実行する前に、Infrastructure が存在する各システムに対して OracleAS Guard で資格証明を設定する必要があります。「[set asg credentials](#)」に、例を示します。

### 6.18.1.2 プライマリ・データベースの指定

プライマリ・トポロジ内の OracleAS Infrastructure データベースを指定するには、`asgctl` プロンプトに次の `asgctl` コマンドを入力します。プライマリ・トポロジ内の OracleAS Infrastructure データベースにアクセスする `sysdba` 権限のあるデータベース・アカウントのユーザー名とパスワードを指定し、OracleAS Infrastructure データベースの TNS サービス名を指定します。

```
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>
```

スタンバイ・サイトの場合も、プライマリ・データベースで指定する値と同じ値を使用します。これは、プライマリとスタンバイの OracleAS Infrastructure データベースのサービス名とパスワードが必ず同じであるためです。

本番サイトまたはスタンバイ・サイトに複数の OracleAS Metadata Repository インスタンスがインストールされており、インスタンス化、同期化、スイッチオーバーまたはフェイルオーバー操作を実行する場合、それらの操作を実行する前に、各 OracleAS Metadata Repository インスタンスに対して `set primary database` コマンドを実行することによって、OracleAS Metadata Repository インスタンスすべてを識別する必要があります。「[set asg credentials](#)」に、例を示します。

### 6.18.1.3 OracleAS Guard ポート番号の設定

OracleAS Guard は、デフォルトのポート (port) 番号である 7890 を使用します。たとえば、`port=7890` として設定されます。システムに追加の Oracle ホームがインストールされている場合、各 Oracle ホームは一意の OracleAS Guard ポート番号を持つ必要があります。その番号は、通常は `port=7891` などのように 1 ずつ増加します。詳細は、[第 6.4.6 項「サポートされる OracleAS Disaster Recovery 構成」](#) を参照してください。

## 6.18.2 OracleAS Metadata Repository Creation Assistant を使用して作成した OracleAS Metadata Repository 構成の特別な考慮事項

次の項目は、OracleAS Metadata Repository Creation Assistant を使用して作成した OracleAS Metadata Repository 構成の特別な考慮事項です。これらのメタデータ・リポジトリ・データベースは、ユーザー・データを含むスキーマとともに Oracle ホームにインストールされます。この理由により、OracleAS Disaster Recovery に関するいくつかの特別な考慮事項があります。

- スタンバイ・サイトでは、OracleAS Disaster Recovery によって作成されるメタデータ・リポジトリはありません。システム管理者は、スタンバイ・サイトで OracleAS Metadata Repository Creation Assistant を使用して、このメタデータ・リポジトリを作成する必要があります。
- スタンバイ・サイトへのトポロジのクローン操作中に、メタデータ・リポジトリでインスタンス化操作は実行されません。
- **警告:** クローン操作を既存の Oracle ホームを含むスタンバイ・システムに実行しないでください。実行すると既存の Oracle ホームが上書きされます。クローン操作は、Oracle ホームがインストールされていないスタンバイ・システムにのみ実行してください。
- OracleAS Disaster Recovery ソリューションは、ユーザー・スキーマが Oracle Data Guard に対してすでに構成されていることを前提としています。
- OracleAS Disaster Recovery ソリューションは、Oracle Data Guard を使用するとき Metadata Repository が管理リカバリ・モードになっていないことを前提としています。
- OracleAS Disaster Recovery は、Metadata Repository に対して Oracle Data Guard のリカバリ・モードが、管理リカバリ・モードであることが判明してもそれを変更しません。そのかわり、OracleAS Guard によってデータベースが管理リカバリ・モードになっており、この機能の設定を変える必要があることを示す警告が発行されます。

- OracleAS Guard は、OracleAS Disaster Recovery ソリューションのために構成された本番トポロジおよびスタンバイ・トポロジの一部であるすべてのシステムのすべての Oracle ホームにインストールする必要があります。OracleAS Guard は、OracleAS Companion CD 2 にあるスタンドアロン・インストール・キットとしてインストールできます。詳細は、Oracle Application Server のインストーション・ガイドの OracleAS Disaster Recovery のインストール情報を参照してください。

## 6.19 OracleAS Disaster Recovery 環境の特別な考慮事項

次の項では、OracleAS Disaster Recovery 環境のいくつかの追加の特別な考慮事項について説明します。

### 6.19.1 いくつかの OracleAS Disaster Recovery サイトを設定する際に注意する必要がある特別な考慮事項

サイトに OracleAS Disaster Recovery を設定する際に注意する必要がある特別な考慮事項は次のとおりです。

- 中間層 CFC 構成
- OracleAS Guard 10g (9.0.4) のクローニング

どちらの場合も、Oracle Internet Directory に格納されているインスタンス名は、本番サイトのインストールが実行された元のホスト名で構成されます。OracleAS Disaster Recovery サイトに対称トポロジがある場合、OracleAS Disaster Recovery のスタンバイ・ピアは、本番サイトと同じようにインストールする必要があります。OracleAS Guard 10g リリース 2 (10.1.2.0.2) 以上のインスタンスのクローニング操作またはトポロジのクローニング操作の場合、操作は構成をミラー化するように実行する必要があります。

スタンバイが非対称トポロジで、本番サイトの物理ホストがスタンバイ・サイトに存在しない場合、ポリシー・ファイル機能を使用して、そのインスタンス名をトポロジからフィルタで除外する必要があります (詳細は第 6.7 項「いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」を参照)。トポロジの検出操作を実行するホストのホスト・ファイルは、元のホスト名を、クローンされた新しいホスト・システムの対応する IP に対応付けしている必要があります。

### 6.19.2 非対称トポロジの構成ファイル ons.conf および dsa.conf の処理

OracleAS Guard 操作で、スタンバイ・サイトの構成ファイルが、プライマリ・サイトのバックアップ操作によって本番サイトの構成ファイルと同期化され、スタンバイ・サイトにリストアされます。

非対称トポロジの場合、スタンバイ・サイトは本番サイトよりノードの数が少なく、ons.conf 構成ファイルのノード名リストは、本番サイトのノード名リストと異なります。したがって、ons.conf 構成ファイルは、ファイルのバックアップ・リストから除外し、スタンバイ・サイトにリストアされないようにします。除外しない場合、ons.conf 構成ファイルにリストされているノードは、本番サイトのノード・リストを反映し、スタンバイ・サイトの実際のノード・リストは反映しなくなります。これによって、存在しないノードに OPMN が ping を実行し続けるという非効率が発生します。

さらに、非対称トポロジの場合、Oracle ホームの dsa.conf 構成ファイルには、スタンバイ・サイトとは異なる特別な設定が本番サイトに含まれていることもあります。たとえば、inventory\_location パラメータ設定は、スタンバイ・サイトとプライマリ・サイトでは異なることがあります。この場合、dsa.conf 構成ファイルも、ファイルのバックアップ・リストから除外し、スタンバイ・サイトにリストアされないようにします。除外しない場合、この例では、スイッチオーバーまたはフェイルオーバー操作の後の OraInventory の場所はスタンバイ・サイトでは正しくありません。

どちらの場合も、次のようにバックアップおよびリストアの除外ファイルを変更し、これらのどちらの構成ファイルも、ファイルのバックアップ・リストから除外し、どちらもスタンバイ・サイトにリストアされないようにします。

```
# Exclude Files
# - Add additional files to this list that you want to be ignored
# - during the configuration file backup/restore
c:¥oracle¥ias1012¥opmn¥conf¥ons.conf
c:¥oracle¥ias1012¥dsa¥dsa.conf
```

dsa.conf ファイルで設定されたディレクティブが、現在、本番サイトとして動作しているサイトで必要な場合は、同期化に dsa.conf ファイルも含め、スイッチオーバーまたはフェイルオーバー後の手順を追加して、物理サイト固有のディレクティブを編集したほうがよいこともあります。

### 6.19.3 OracleAS Disaster Recovery 環境における他の特別な考慮事項

いくつかの追加の特別な考慮事項の詳細は、[第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#)を参照してください。



# OracleAS Guard asgctl コマンドライン・リファレンス

この章では、asgctl コマンドについてのリファレンス情報を記載します。表 7-1 に、すべての asgctl コマンドの要約を示します。表 7-2 には、OracleAS 10g リリース 2 (10.1.2.0.2) で廃止されたすべての asgctl コマンドの要約を示します。その後の項では、多くのコマンドで共通なリファレンス情報と各コマンドについてのリファレンス情報を詳細に記載します。

表 7-1 asgctl コマンドの要約

| コマンド                          | 説明                                                                                                                                                              |
|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| add instance                  | 指定されているインスタンス名と、このインスタンスがインストールされているホスト・システムの名前をローカル・トポロジ・ファイルに追加し、指定に応じてこの更新済トポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播します。                                |
| asgctl                        | OracleAS Guard クライアントのコマンドライン・ユーティリティ asgctl を起動します。UNIX システムでは、asgctl.sh は <ORACLE_HOME>/dsa/bin にあり、Windows システムでは、asgctl.bat は <ORACLE_HOME>%dsa%\bin にあります。 |
| clone instance                | 単一の本番インスタンスをスタンバイ・システムにクローニングします。                                                                                                                               |
| clone topology                | 複数の本番の中間層インスタンスをスタンバイの中間層システムにクローニングします。                                                                                                                        |
| connect asg                   | OracleAS Guard クライアントを OracleAS Guard サーバーに接続します。                                                                                                               |
| create standby database       | スタンバイ・データベースをリモート・ホスト・システムに作成します。                                                                                                                               |
| disconnect                    | OracleAS Guard クライアントを OracleAS Guard サーバーから切断します。                                                                                                              |
| discover topology             | 本番サイトの同じ Oracle Internet Directory を共有するトポロジ内の Oracle Internet Directory のすべてのインスタンスを問い合せて検出し、トポロジについて記述するトポロジ XML ファイルを生成します。                                  |
| discover topology within farm | Oracle Internet Directory が使用できないとき、サイトのファーム内のトポロジを検出します。この場合、OracleAS Guard サーバーは OPMN を使用して、ファーム内のトポロジを検出します。                                                 |
| dump policies                 | 一部の asgctl コマンドに対して、詳細なデフォルトのポリシー情報をそれぞれの XML 形式ファイルに書き込むように OracleAS Guard サーバーに指定します。各ポリシーファイルを編集して、各管理コマンドで使用するトポロジの障害時リカバリ・ポリシーを定義するように後で指定することもできます。       |

**表 7-1 asgctl コマンドの要約 (続き)**

| コマンド                     | 説明                                                                                                                                                        |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| dump topology            | トポロジに関する詳細情報を画面に表示するように OracleAS Guard サーバーに指定します。また、ファイルに書き込むように指定することもできます。                                                                             |
| exit                     | OracleAS Guard クライアントの既存の接続をすべて切断し、OracleAS Guard クライアントを終了します。これは、quit コマンドと同じ結果になります。                                                                   |
| failover                 | 本番サイトの計画外停止の際に、スタンバイ・サイトを本番サイトに切り替えます。                                                                                                                    |
| help                     | ヘルプ情報をコマンドラインに表示します。                                                                                                                                      |
| instantiate topology     | スタンバイ・サイトにトポロジを作成します (プライマリ・サイトとスタンバイ・サイトが OracleAS Disaster Recovery に対して有効であることが検証された後)。また、プライマリ・サイトとスタンバイ・サイトで一貫性が維持されるように、プライマリ・サイトとスタンバイ・サイトを同期化します。 |
| quit                     | OracleAS Guard クライアントの既存の接続をすべて切断し、OracleAS Guard クライアントを終了します。これは、exit コマンドと同じ結果になります。                                                                   |
| remove instance          | 指定されたインスタンス名をローカル・トポロジ・ファイルから削除し、指定に応じて、この更新されたトポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播します。                                                         |
| run                      | OracleAS Guard がインストールされている任意のホームにあるスクリプトまたはプログラムをリモートで実行します。                                                                                             |
| set asg credentials      | OracleAS Guard クライアントから OracleAS Guard サーバーへの接続、および OracleAS Guard サーバーと特定のホスト間の接続を認証するための資格証明を設定します。                                                     |
| set echo                 | asgctl スクリプトのコマンドエコーのオンおよびオフを設定します。                                                                                                                       |
| set new primary database | スタンバイ・トポロジの OracleAS Infrastructure データベースを新しいプライマリ OracleAS Infrastructure データベースとして指定します。                                                               |
| set noprompt             | asgctl スクリプトで、プロンプトなしの状態を設定します。この後のすべての対話プロンプトが無視されます。                                                                                                    |
| set primary database     | プライマリ・トポロジの OracleAS Infrastructure データベースを指定します。                                                                                                         |
| set trace                | トレース・フラグを指定したトレースを有効または無効にします。フラグによるトレースをオンにすると、トレースの出力は OracleAS Guard ログ・ファイルに書き込まれます。                                                                  |
| show env                 | OracleAS Guard クライアントが接続している OracleAS Guard サーバーの現在の環境を示します。                                                                                              |
| show operation           | 現在の操作を表示します。                                                                                                                                              |
| shutdown                 | OPMN が実行されていないシステムのオペレーティング・システムのコマンドラインのプロンプトで、OracleAS Guard サーバーを停止します。このコマンドは、インスタンスまたはトポロジのクローニングにのみ使用します。                                           |
| shutdown topology        | 稼働中のトポロジを停止します。                                                                                                                                           |
| startup                  | OPMN が実行されていないシステムのオペレーティング・システムのコマンドラインのプロンプトで、OracleAS Guard サーバーを起動します。このコマンドは、インスタンスまたはトポロジのクローニングにのみ使用します。                                           |
| startup topology         | 停止したトポロジを起動します。                                                                                                                                           |

表 7-1 asgctl コマンドの要約 (続き)

| コマンド                | 説明                                                                                                                                   |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| stop operation      | 指定された操作を停止します。                                                                                                                       |
| switchover topology | 本番サイトのスケジューリングした停止の間に、本番サイトの役割をスタンバイ・サイトに移行し、スタンバイ・サイトを本番サイトに切り替えます。                                                                 |
| sync topology       | スタンバイ・サイトとプライマリ・サイトを同期化し、プライマリ・サイトとスタンバイ・サイトの間で一貫性を保ちます。                                                                             |
| verify topology     | トポロジが実行中であること、およびその構成が有効であることを検証します。スタンバイ・トポロジが指定された場合は、プライマリ・トポロジとスタンバイ・トポロジを比較し、両者が OracleAS Disaster Recovery の要件を満たしていることを検証します。 |

表 7-2 廃止された asgctl コマンドの要約

| コマンド                   | 説明                                                                                                                                                        |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| dump farm (廃止済)        | ファームに関する詳細情報を画面に表示するように OracleAS Guard サーバーに指定します。また、ファイルに書き込むように指定することもできます。                                                                             |
| instantiate farm (廃止済) | スタンバイ・サイトにファームを作成します (プライマリ・サイトとスタンバイ・サイトが OracleAS Disaster Recovery に対して有効であることが検証された後)。また、プライマリ・サイトとスタンバイ・サイトで一貫性が維持されるように、プライマリ・サイトとスタンバイ・サイトを同期化します。 |
| shutdown farm (廃止済)    | 稼働中のファームを停止します。                                                                                                                                           |
| startup farm (廃止済)     | 停止したファームを起動します。                                                                                                                                           |
| switchover farm (廃止済)  | 本番サイトのスケジューリングした停止の間に、本番サイトの役割をスタンバイ・サイトに移行し、スタンバイ・サイトを本番サイトに切り替えます。                                                                                      |
| sync farm (廃止済)        | スタンバイ・サイトとプライマリ・サイトを同期化し、プライマリ・サイトとスタンバイ・サイトの間で一貫性を保ちます。                                                                                                  |
| verify farm (廃止済)      | ファームが実行中であること、およびその構成が有効であることを検証します。スタンバイ・ファームが指定された場合は、プライマリ・ファームとスタンバイ・ファームを比較し、両者が OracleAS Disaster Recovery の要件を満たしていることを検証します。                      |

## 7.1 OracleAS Guard asgctl コマンドに共通な情報

この項では、OracleAS Guard asgctl コマンドに共通な情報について説明します。

### 一般的な情報

すべての asgctl コマンドでは、システムのホスト名を指定できるパラメータがある場合、システムの物理ホスト名を指定して実行する必要があります。

起動および停止コマンドの例外とともに asgctl コマンドを実行する場合は、OracleAS Guard クライアントが OracleAS Guard サーバーに接続している必要があります。

OracleAS Guard サーバーは、構成している各システムで実行されるすべての操作の調整サーバーとして動作します。デフォルトでは、これが connect asg コマンドを実行しているローカル・システムになります。このシステムは、本番サイトのトポロジのメンバーである必要があります。

**OracleAS Guard サーバー情報**

OracleAS Guard サーバーは、スタンバイ・ホスト・システムで起動する必要があります (<standby\_topology\_host>)。OracleAS Guard サーバーは、次のように opmnctl コマンドライン・ユーティリティを使用して停止および起動できます。

On UNIX systems:

```
<ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=ASG
```

On Windows systems:

```
<ORACLE_HOME>%opmn%bin\opmnctl stopproc ias-component=ASG
```

**7.2 OracleAS Guard コマンドの一部に特有の情報**

この項では、インスタンス化、同期化、フェイルオーバー、スイッチオーバー、トポロジのダンプ、トポロジの検出、トポロジのクローニング、トポロジの検証、プライマリ・データベースの設定、ASG 資格証明の設定などの OracleAS Guard 操作の一部に特有の情報を説明します。

本番サイトまたはスタンバイ・サイトに複数の OracleAS Metadata Repository インスタンスがインストールされているときに、インスタンス化、同期化、スイッチオーバー、またはフェイルオーバー操作を実行する場合、実行前に `set primary database` コマンドを実行して、すべての OracleAS Metadata Repository インスタンスを特定する必要があります。

OracleAS Guard 操作を実行する場合は、事前に `emagent` を停止する必要があります。この操作は、フェイルオーバーやスイッチオーバーなど、OracleAS サービスをリサイクルする OracleAS Guard コマンドにおいて必要です。スクリプトで `asgctl` の `run` コマンドを発行して、OracleAS Guard からこの操作を実行できます。この操作を実行しないと、「ORA-01093: ALTER DATABASE CLOSE は接続中のセッションがない場合にのみ実行できます」などのエラー・メッセージが表示される場合があります。`emagent` の停止については、スイッチオーバー操作に関する記述でのみ説明しています。ただし、前述したように、これはすべての OracleAS Guard 操作に対して適用されます。

インスタンス化、同期化、スイッチオーバー、およびフェイルオーバーなどの重要な OracleAS Guard の操作を実行する前に、接続している OracleAS Guard サーバーと異なる資格証明を持つトポロジの OracleAS Guard サーバーに対して、OracleAS Guard で資格証明を設定する必要があります。「`set asg credentials`」に、例を示します。

最初にトポロジ XML ファイルを作成するために、最初に OracleAS Disaster Recovery 環境を設定するとき、`discover topology` コマンドを実行する必要があります。そこで、後ほど本番サイトで別の Oracle ホームを取得するときや、フェイルオーバーまたはスイッチオーバー操作で本番サイトからスタンバイ・サイトに役割を変更するとき、トポロジの検出操作を実行する必要があります。

OracleAS Guard の管理者は、次に `asgctl` の `sync topology` 操作、`switchover topology` 操作または `instantiate topology` 操作を実行するまで 1 分程度の時間を置く必要があります。特に `incremental` を指定して `sync topology` 操作を実行する場合に重要です。これは、基になる一連のイベントをできるかぎり順番どおりに実行するための既知の制限です。

次の情報とシナリオは、トポロジ・ファイルの利用方法を理解するのに役立ちます。

- OracleAS 10g リリース 2 (10.1.2) 以前を使用している場合は、`discover topology` コマンドを使用します。Oracle Internet Directory がトポロジに含まれていない場合は、`discover topology within farm` コマンドを使用します。
- OracleAS 10g リリース 2 (10.1.2) およびリリース 3 (10.1.3) を使用している場合は、(OracleAS 10.1.2 または 10.1.3 のホームから) OracleAS Guard 10g リリース 2 (10.1.2) サーバーに接続し、`discover topology` コマンドを実行します。OracleAS Guard によって、検出されたトポロジの各 OracleAS ホームに `topology.xml` ファイルが書き込まれます。次に、インスタンスを追加する場合 (`add instance` コマンド) は、OracleAS 10g リリース 3 (10.1.3) のホームからこの操作を実行して、既存のトポロジ内の任意の OracleAS 10g リリース 2 (10.1.2) のシステムに接続し、`add instance` コマンドを実行します。

- OracleAS 10g リリース 3 (10.1.3) を使用している場合は、OracleAS Guard 10g リリース 3 (10.1.3) サーバーに接続し、そのインスタンスをトポロジ (まだ存在しない) に追加します。次に、OracleAS Guard の同じ接続を使用して、インスタンスをさらに追加します。

以上の 2 つのシナリオで強調する必要がある重要な点は、トポロジに追加するインスタンスの OracleAS ホームに topology.xml ファイルが存在しない場合、OracleAS Guard によって新しい topology.xml ファイルが作成され、このファイルによって実質的に新しいトポロジが定義されることです。

ポリシー・ファイルを使用する場合、XML ポリシー・ファイルの内容を編集して、asgctl コマンド ([clone topology](#)、[dump topology](#)、[failover](#)、[instantiate topology](#)、[switchover topology](#)、[sync topology](#)、および [verify topology](#)) のいずれかに許可する実行操作のドメインをインスタンスごとに定義します。この XML ポリシー・ファイルのそれぞれのインスタンス・リストのエントリ (clone\_policy.xml、dump\_policy.xml、failover\_policy.xml、instantiate\_policy.xml、switchover\_policy.xml、sync\_policy.xml、および verify\_policy.xml) では、本番とスタンバイの対応する組合せを、コマンド操作の成功を判断するためのコマンドの成功要件を定義する特定の属性とともに、論理的に結び付けます。XML ポリシー・ファイルの詳細と例は、[第 6.7 項「いくつかの asgctl コマンドでのポリシー・ファイルのダンプとポリシー・ファイルの使用」](#) を参照してください。

## 7.2.1 CFC 環境の OracleAS Disaster Recovery 構成に関する特別な考慮事項

プライマリ・トポロジまたはスタンバイ・トポロジ、またはその両方で CFC を使用する OracleAS Disaster Recovery 構成では、asgctl の clone、instantiate topology、switchover topology、failover コマンドを実行する前に次の情報について考慮する必要があります。メタデータ・リポジトリ・データベースのコールド・バックアップまたはリストアの前に、OracleAS Recovery Manager はまずデータベースを停止します。

たとえば、Windows CFC 環境では、Oracle Fail Safe がデータベース・ポーリングを実行し、データベースが停止すると再起動します。したがって、管理者は、クローニング、インスタンス化、スイッチオーバー、またはフェイルオーバーの操作を実行する前に Oracle Fail Safe のデータベース・ポーリングを無効にする必要があります。そして、バックアップやリストアの操作後 (クローニング、インスタンス化、スイッチオーバー、またはフェイルオーバー操作の完了後) に再度有効にします。この一連の操作を実行する手順は、[第 7.2.1.1 項「CFC 環境でインスタンス化およびフェイルオーバー操作を実行するときの特別な考慮事項」](#) および [第 7.2.1.3 項「CFC 環境でスイッチオーバー操作を実行するときの特別な考慮事項」](#) の「注意」に記載されています。

### 7.2.1.1 CFC 環境でインスタンス化およびフェイルオーバー操作を実行するときの特別な考慮事項

プライマリ・トポロジまたはスタンバイ・トポロジ、またはその両方で CFC を使用する OracleAS Disaster Recovery 構成では、asgctl の clone、instantiate、switchover、または failover 操作を実行する前に次の情報について考慮する必要があります。

メタデータ・リポジトリ・データベースのコールド・バックアップまたはリストアの前に、OracleAS Recovery Manager はまずデータベースを停止します。

たとえば、Windows CFC 環境では、Oracle Fail Safe がデータベース・ポーリングを実行し、データベースが停止すると再起動します。したがって、管理者は、インスタンス化、スイッチオーバー、またはフェイルオーバーの操作を実行する前に Oracle Fail Safe のデータベース・ポーリングを無効にする必要があります。そして、バックアップやリストアの操作後 (クローニング、インスタンス化、スイッチオーバー、またはフェイルオーバー操作の完了後) に再度有効にします。

この一連の操作を実行する手順は次のとおりです。

1. Microsoft クラスタ アドミニストレータを使用して、Application Server のリソースを含む クラスタ・グループを開きます。Oracle Process Manager、Oracle Database、Oracle Listener の順序で、リソースをオフラインにします。
2. Windows の「コントロールパネル」の「サービス」を使用して、Fail Safe Listener、Oracle Database の順に、サービスを開始します。

3. Windows のコマンド・プロンプトで、sqlplus コマンドライン・ユーティリティを使用してデータベースを起動します。
4. Windows の「コントロールパネル」の「サービス」を使用して、Oracle Process Manager を開始します。
5. clone、instantiate、switchover、failover などの asgctl コマンドを実行します。
6. Microsoft クラスタ アドミニストレータを使用して、Application Server のリソースが含まれるクラスタ・グループを起動し、Oracle Listener、Oracle Database、Oracle Process Manager の順序でリソースを開始します。

### 7.2.1.2 CFC 環境でインスタンス化操作を実行するときの特別な考慮事項と回避方法

インスタンス化操作を実行するとき、OracleAS Guard では本番サイトとスタンバイ・サイトの両方で tnsnames.ora ファイルにリモート・データベースのエントリを追加します。このエントリのサービス名は、データベースのサービス名に `_REMOTE1` を付けた名前になります (たとえば、`ORCL_REMOTE1`)。エントリには、データベースが実行されているターゲット・ホストの IP アドレスが含まれます。本番サイトの IP はスタンバイ・システムを参照し、スタンバイ・サイトの IP は本番システムを参照します。

CFC 環境では、物理 IP ではなく仮想 IP を使用してデータベースにアクセスします。OracleAS Guard で tnsnames.ora エントリを作成するときに、仮想 IP を使用する必要がありますが、実際は物理 IP を使用してしまいます。この問題は、OracleAS Guard の今後のリリースで修正する予定です。対処方法として、この環境でインスタンス化操作を実行する際は、インスタンス化操作の後に tnsnames.ora ファイルを編集して、エントリの物理 IP をデータベースのアクセスに使用する仮想 IP に置き換えます。

### 7.2.1.3 CFC 環境でスイッチオーバー操作を実行するときの特別な考慮事項

プライマリ・トポロジまたはスタンバイ・トポロジ、またはその両方で CFC を使用する OracleAS Disaster Recovery 構成では、asgctl の instantiate topology、switchover topology、または failover コマンドを実行する前に次の情報について考慮する必要があります。

メタデータ・リポジトリ・データベースのコールド・バックアップまたはリストアの前に、OracleAS Recovery Manager はまずデータベースを停止します。

たとえば、Windows CFC 環境では、Oracle Fail Safe がデータベース・ポーリングを実行し、データベースが停止すると再起動します。したがって、管理者は、インスタンス化、スイッチオーバー、またはフェイルオーバーの操作を実行する前に Oracle Fail Safe のデータベース・ポーリングを無効にする必要があります。そして、バックアップやリストアの操作後 (インスタンス化、スイッチオーバー、またはフェイルオーバー操作の完了後) に再度有効にします。

この一連の操作を実行する手順は次のとおりです。

1. Microsoft クラスタ アドミニストレータを使用して、Application Server のリソースを含むクラスタ・グループを開きます。Oracle Process Manager、Oracle Database、Oracle Listener の順序で、リソースをオフラインにします。
2. Windows の「コントロールパネル」の「サービス」を使用して、Fail Safe Listener、Oracle Database の順に、サービスを開始します。
3. Windows のコマンド・プロンプトで、sqlplus を使用してデータベースを起動します。
4. `instantiate topology`、`switchover topology`、`failover` などの asgctl コマンドを実行します。
5. Microsoft クラスタ アドミニストレータを使用して、Application Server のリソースが含まれるクラスタ・グループを起動し、Oracle Listener、Oracle Database、Oracle Process Manager の順序でリソースを開始します。

## 7.2.2 OracleAS Disaster Recovery 環境における他の特別な考慮事項

OracleAS Disaster Recovery 環境についての特別な考慮事項の詳細は、[第 6.18 項「いくつかの OracleAS Metadata Repository 構成の特別な考慮事項」](#) および [第 6.19 項「OracleAS Disaster Recovery 環境の特別な考慮事項」](#) を参照してください。

## add instance

指定されているインスタンス名と、このインスタンスがインストールされているホスト・システムの名前をローカル・トポロジ・ファイルに追加し、指定に応じて、更新済トポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播します。

### 書式

```
add instance <instance_name> on <instance_host> [to topology]
```

### パラメータ

#### instance\_name

トポロジ・ファイルに追加されるインスタンスの名前。

#### instance\_host

このインスタンスがインストールされているホストの名前。DR 環境では、このホストに別名または仮想ホスト名がある場合、トポロジ・ファイルにはこの値が格納されるため、仮想ホスト名を使用する必要があります。

#### to topology

キーワード。コマンドラインにこのキーワードを指定した場合、更新されたトポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播するように OracleAS Guard を指定します。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ環境全体に本番トポロジ・ファイルが自動的に伝播されます。

### 使用方法

このコマンドは、OracleAS Guard クライアントが接続されている OracleAS Guard サーバー上のインスタンスの管理に役立ちます。たとえば、このローカル・ホスト・システム上のインスタンスに問題があるために、[remove instance](#) コマンドを使用して、ローカル・トポロジ・ファイルから、またはトポロジ内部のすべてのトポロジ・ファイルから、そのインスタンスを削除する場合があります。そこで、ローカル・トポロジ・ファイルまたはトポロジ内のトポロジ・ファイルすべてに、適切なインスタンスを追加することにします。この場合、[asgctl](#) コマンドを起動してトポロジ全体に対して実行したとき、ポリシー・ファイルではこのインスタンスについて成功要件属性が `Ignore` に設定された可能性があるため、そのポリシー・ファイルは不適切なインスタンスの管理に使用するには好ましくないことがあります。

このコマンドは、OID が利用できない Disaster Recovery ファーム、つまり OracleAS 10g リリース 3 (10.1.3) のみのトポロジの管理に特に役立ちます。このファーム内の各インスタンスにトポロジ・ファイルを最初に作成するには、[discover topology within farm](#) コマンドを使用する必要があります。その後、[add instance](#) および [remove instance](#) コマンドを使用して、ローカル・トポロジ・ファイルで個別のインスタンスを追加または削除することで、インスタンスを管理できるようになります。to topology または from topology キーワードを指定した場合、更新済のローカル・トポロジ・ファイルの変更が Disaster Recovery 環境におけるインスタンスすべてに伝播します。トポロジ・ファイルの詳細は、[第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

このコマンドは、OracleAS 10.1.3 J2EE インスタンスを OID ベースの 10.1.2.0.2 トポロジに追加して、バージョンが混在した Disaster Recovery 環境をサポートする場合に役立ちます。たとえば、[add instance](#) コマンドを使用して、OracleAS 10.1.3 J2EE インスタンスを OID ベースの 10.1.2.0.2 トポロジに追加できます。ユースケースについては、[第 6.9 項「既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除」](#) を参照してください。トポロジ・ファイルの詳細は、[第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

またこのコマンドは、リリース 9.0.4 からリリース 10.1.3 までの任意の OracleAS リリースに対して、バージョンが混在したトポロジをサポートするためにも役立ちます。唯一の要件は、リリース 10.1.3 の OracleAS Guard スタンドアロン・キットが、リリース 9.0.4 の Disaster Recovery 環境におけるすべてのシステムにインストールされていることです。詳細は、[第 9 章「OracleAS Disaster Recovery サイトのアップグレード手順」](#)を参照してください。

## 例

次の例では、`prodfra` というローカル・ホスト・システムにインストールされている `oc4j1` というインスタンスを、ローカル・トポロジ・ファイルのみに追加します。

```
ASGCTL> connect asg prodfra ias_admin/adminpwd
Successfully connected to prodfra:7890
ASGCTL> discover topology within farm
ASGCTL> add instance oc4j1 on prodfra
```

---

## asgctl

オペレーティング・システムのコマンドライン・プロンプトから OracleAS Guard クライアントを起動します。スクリプトのパス名が指定された場合はスクリプトを実行します。

### 書式

```
asgctl@[filename]
```

### パラメータ

**filename = <file-path>**

スクリプトとして実行する asgctl コマンドを含んだファイルへのパス。

### 使用方法

UNIX システムでは、asgctl.sh は <ORACLE\_HOME>/dsa/bin にあり、Windows システムでは、asgctl.bat は <ORACLE\_HOME>%dsa%\bin にあります。

### 例

```
> asgctl.sh
Application Server Guard: Release 10.1.3.0.0

(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
ASGCTL>
```

## clone instance

単一の本番インスタンスをスタンバイ・システムにクローニングします。

### 書式

```
clone instance <instance> to <standby_topology_host> [no standby]
```

### パラメータ

#### instance

インスタンス名。

#### standby\_topology\_host

インスタンスのクローニング先となるスタンバイ・トポロジのホストの名前。

#### no standby

キーワード。コマンドラインにこのキーワードを指定した場合、Disaster Recovery 内にインスタンスを設定することなく (Data Guard なし)、インスタンスをクローニングするように OracleAS Guard を指定します。MR ホームでのみ、インスタンス化操作が通常行われますが、この場合は実行されません。

### 使用方法

このコマンドは、中間層の本番インスタンスをスタンバイ中間層のホスト・システムにクローニングする際に有用です。インスタンスのクローニング操作により、Oracle インスタンスをスタンバイ中間層システムにインストールしてインスタンス化操作を実行する手間が省けます。

クローニングする本番インスタンスは、スタンバイ・システムに存在してはなりません。

スタンバイ・サイト・システムにインスタンスのクローニング操作を実行するための要件は、次のとおりです。

- スタンバイ・システムに OracleAS Guard スタンドアロン・キットがインストールされている必要があります。
- スタンバイ・システムの OracleAS Guard ホームに Backup and Restore がインストールされている必要があります。
- Java Development Kit とその jar ユーティリティがスタンバイ・システムにインストールされている必要があります。
- Windows システムの場合、サービス・キット (sc.exe) がスタンバイ・システムにインストールされている必要があります。

コマンドラインに no standby キーワードが指定されている場合、エントリ <nodes list = "nodes"/>、<discover list = "nodes"/>、および <gateway list = "nodes"/> を含むトポロジ・エントリ (<topology> </topology>) は、元の構成と競合しないように、opmn.xml ファイルから削除されます。通常、このトポロジ・エントリはスイッチオーバー操作またはフェイルオーバー操作によって opmn.xml ファイルにリライトされますが、どちらの操作も発生しないため、スタンバイ・サイトの opmn.xml ファイルにはこの情報が記述されません。

詳細は、[第 6.10 項「OracleAS Guard 操作: スタンバイ・システムへの 1 つ以上の本番インスタンスのスタンバイ・サイト・クローニング」](#)を参照してください。

クローニングの基本手順は、次のクローニング前 (UNIX システムの場合のみ) およびクローニング中の各手順から構成されています。

---



---

**注意：** Windows システムの OracleAS Disaster Recovery 10g リリース 2 (10.1.2.0.2) については、OracleAS 10g リリース 2 (10.1.2.0.2) ドキュメント・セットの『Oracle Application Server 高可用性ガイド』の同じ項に記載されている、クローニング前とクローニング後の手順を参照してください。

---



---

### クローニング前の手順 (UNIX システムの場合のみ)

本番サイトとスタンバイ・サイトの各インスタンスで、次の手順を実行します。

1. su - root でログインします。
2. インスタンス・ホームへ移動します。
3. OracleAS Guard サーバーを停止します。  

```
> <ORACLE HOME>/opmn/bin/opmnctl stopproc ias-component=ASG
```
4. <ORACLE\_HOME>/dsa/bin 内の dsaServer.sh をすべてのユーザーが実行できることを確認します。実行できない場合は、現在の実行権限を書き留めてから、次のコマンドを実行して実行権限を変更します。  

```
chmod +x dsaServer.sh
chmod u+x asgexec
```
5. asgctl を起動し、**startup** コマンドを実行します。  

```
> asgctl.sh startup
```
6. UNIX システムの root をログアウトします。

### クローニング中の手順

本番サイトのインスタンスで、次の手順を実行します。

1. ユーザー (UNIX システムの root 以外のユーザー) としてログインします。
2. 本番インスタンスのホームへ移動します。
3. asgctl を起動してから clone instance コマンドを実行して、インスタンスをスタンバイ・トポロジのホスト・システムにクローニングします。

---



---

**注意：** コマンド出力には、多数の接続メッセージが表示されます。これは、これらの操作中に OracleAS Guard サーバーがリサイクルされるために発生する正常な動作です。

---



---

4. システムからログアウトします。

---



---

**注意：** UNIX システムでは、OracleAS Guard が root として実行されていない場合、操作を続行するには、各インスタンス・ホームの基本的な操作を root として (手動で) 実行する必要があることが OracleAS Guard クライアントによって示されます。

---



---

この最後の手順でインスタンスのクローニング操作が完了し、システムがクローニング操作開始前の状態に戻ります。この時点で、asgctl を起動してから本番システムに接続して、トポロジを検出できます。その後、検証操作を実行して、本番トポロジとスタンバイ・トポロジが有効で一致しているかどうかを調べられます。

## 例

次の例では、portal\_2 という名前のインスタンスが asmid2 という名前のスタンバイ・トポロジのホスト・システムにクローニングされます。

1. Check the prerequisites as described in the Usage Notes.
2. Perform the Pre-Clone steps as described in the Usage Notes.
3. Perform the Clone steps as described in the Usage Notes.
  - a. Log in as user to any production system.
  - b. CD to any production instance home.
  - c. Invoke asgctl and perform the clone instance command.

```
> asgctl.sh
```

```
Application Server Guard: Release 10.1.3.0.0
```

```
(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
```

```
ASGCTL> connect asg prodoc4j oc4jadmin/adminpwd
```

```
Successfully connected to prodoc4j:7890
```

```
ASGCTL> set primary database sys/testpwd@asdb
```

```
Checking connection to database asdb
```

```
ASGCTL> clone instance prodoc4j2 to asmid2
```

```
Generating default policy for this operation
```

```
.
```

```
.
```

```
.
```

```
ASGCTL> disconnect
```

```
ASGCTL> exit
```

```
>
```

- d. Log off the system

## clone topology

複数の本番の中間層インスタンスをスタンバイの中間層システムにクローニングします。

### 書式

```
clone topology to <standby_topology_host> [using policy <file>] [no standby]
```

### パラメータ

#### standby\_topology\_host

スタンバイ・トポロジ・ホスト・システムの名前。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

#### no standby

キーワード。コマンドラインにこのキーワードを指定した場合、Disaster Recovery 内にインスタンスを設定することなく (Data Guard なし)、インスタンスをクローニングするように OracleAS Guard を指定します。MR ホームでのみ、インスタンス化操作が通常行われますが、この場合は実行されません。

### 使用方法

このコマンドは、中間層システムの複数の本番インスタンスをスタンバイ中間層のホスト・システムにクローニングする際に有用です。トポロジのクローニング操作により、Oracle インスタンスをスタンバイ中間層システムにインストールしてインスタンス化操作を実行する手間が省けます。

トポロジのクローニング操作の一部として、インスタンスをクローニングして OracleAS Metadata Repository をインスタンス化しますが、OracleAS Metadata Repository Creation Assistant を使用して作成された OracleAS Metadata Repository 構成では、インスタンス化操作は実行しません。

クローニングする本番インスタンスは、スタンバイ・システムに存在してはなりません。

スタンバイ・サイト・システムにトポロジのクローニング操作を実行するための要件は、次のとおりです。

- OracleAS Guard スタンドアロン・キットが各スタンバイ・システムにインストールされている必要があります。
- 各スタンバイ・システムの各 OracleAS Guard ホームに Backup and Restore がインストールされている必要があります。
- Java Development Kit とその jar ユーティリティが各スタンバイ・システムにインストールされている必要があります。
- Windows システムのみの場合、サービス・キット (sc.exe) が各スタンバイ・システムにインストールされている必要があります。

コマンドラインに no standby キーワードが指定されている場合、エントリ <nodes list = "nodes"/>、<discover list = "nodes"/>、および <gateway list = "nodes"/> を含むトポロジ・エントリ (<topology> </topology>) は、元の構成と競合しないように、opmn.xml ファイルから削除されます。通常、このトポロジ・エントリはスイッチオーバー操作またはフェイルオーバー操作によって opmn.xml ファイルにリライトされますが、どちらの操作も発生しないため、スタンバイ・サイトの opmn.xml ファイルにはこの情報が記述されません。

詳細は、第 6.10 項「OracleAS Guard 操作: スタンバイ・システムへの 1 つ以上の本番インスタンスのスタンバイ・サイト・クローニング」を参照してください。

クローニングの基本手順は、次のクローニング前（UNIX システムの場合のみ）およびクローニング中の各手順から構成されています。

---

---

**注意：** Windows システムの OracleAS Disaster Recovery 10g リリース 2 (10.1.2.0.2) については、OracleAS 10g リリース 2 (10.1.2.0.2) ドキュメント・セットの『Oracle Application Server 高可用性ガイド』の同じ項に記載されている、クローニング前とクローニング後の手順を参照してください。

---

---

#### クローニング前の手順（UNIX システムの場合のみ）

本番サイトとスタンバイ・サイトの各インスタンスで、次の手順を実行します。

1. su - root でログインします。
2. インスタンス・ホームへ移動します。
3. OracleAS Guard サーバーを停止します。  

```
> <ORACLE_HOME>/opmn/bin/opmnctl stopproc ias-component=ASG
```
4. **UNIX システムの場合のみ：** <ORACLE\_HOME>/dsa/bin の dsaServer.sh をすべてのユーザーが実行できることを確認します。実行できない場合は、現在の実行権限を書き留めてから、次のコマンドを実行して実行権限を変更します。  

```
chmod +x dsaServer.sh  
chmod u+x asgexec
```
5. asgctl を起動し、**startup** コマンドを実行します。  

```
> asgctl.sh startup
```
6. UNIX システムの root をログアウトします。

#### クローニング中の手順

本番サイトのインスタンスで、次の手順を実行します。

1. ユーザー（UNIX システムの root 以外のユーザー）としてログインします。
2. 本番インスタンスのホームへ移動します。
3. asgctl を起動してから clone topology コマンドを実行して、トポロジをスタンバイ・トポロジのホスト・システムにクローニングします。

---

---

**注意：** コマンド出力には、多数の接続メッセージが表示されます。これは、これらの操作中に OracleAS Guard サーバーがリサイクルされるために発生する正常な動作です。

---

---

4. システムからログアウトします。

---

---

**注意：** UNIX システムでは、OracleAS Guard が root として実行されていない場合、操作を続行するには、各インスタンス・ホームの基本的な操作を root として（手動で）実行する必要があることが OracleAS Guard クライアントによって示されます。

---

---

この最後の手順でトポロジのクローニング操作が完了し、システムがクローニング操作開始前の状態に戻ります。この時点で、asgctl を起動してから本番システムに接続して、トポロジを検出できます。その後、検証操作を実行して、本番トポロジとスタンバイ・トポロジが有効で一致しているかどうかを調べられます。

詳細は、第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」および第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。

## 例

次のコマンドは、OracleAS Guard クライアントでトポロジをスタンバイ・トポロジ・ホスト・システム standbyinfra にクローニングする例を示しています。

1. Check the prerequisites as described in the Usage Notes.
2. Perform the Pre-Clone steps as described in the Usage Notes.
3. Perform the Clone steps as described in the Usage Notes.
  - a. Log in as user to any production system.
  - b. CD to any production instance Oracle home.
  - c. Invoke asgctl and perform the clone instance command.

```
> asgctl.sh
```

```
Application Server Guard: Release 10.1.3.0.0
```

```
(c) Copyright 2004, 2005 Oracle Corporation. All rights reserved
```

```
ASGCTL> connect asg prodoc4j oc4jadmin/adminpwd
```

```
Successfully connected to prodoc4j:7890
```

```
ASGCTL> set primary database sys/testpwd@asdb
```

```
Checking connection to database asdb
```

```
ASGCTL> clone topology to standbyinfra
```

```
Generating default policy for this operation
```

```
.
```

```
.
```

```
.
```

```
# Command to use if you are using a policy file
```

```
# clone topology to standbyinfra using policy <file>
```

```
.
```

```
.
```

```
.
```

```
ASGCTL> disconnect
```

```
ASGCTL> exit
```

```
>
```

- d. Log off the system

## connect asg

Oracle Application Server サービスが実行されているシステムの OracleAS Guard サーバーに OracleAS Guard クライアントを接続します。

### 書式

```
connect asg [<host-name>[:<port>]] <ias_administrative_account>/<password>
```

### パラメータ

#### host-name = <host-name>

OracleAS Guard クライアントを接続する OracleAS Guard サーバーのホスト・システムの名前。この OracleAS Guard サーバーは、構成している各システムで実行されるすべての操作の調整サーバーになります。OracleAS Guard クライアントと OracleAS Guard サーバーが同じノードに存在する場合は、ホスト名の指定は省略可能です。

#### port

Oracle ホーム内の OracleAS Guard サーバーのポート番号。

#### <ias\_administrative\_account>/password

OracleAS 10.1.3 インストールの場合、ユーザー名は oc4jadmin を使用し、この oc4jadmin アカウントのパスワードは Oracle Application Server のインストール時に作成されたものを使用する必要があります。OracleAS 10.1.2.0.2 以前のインストールの場合、ユーザー名は ias\_admin を使用し、この ias\_admin アカウントのパスワードは Oracle Application Server のインストール時に作成されたものを使用する必要があります。

---

**注意：** OracleAS 10.1.3 インストールの場合、ユーザー名は oc4jadmin を使用し、この oc4jadmin アカウントのパスワードは Oracle Application Server 10.1.3 のインストール時に作成されたものを使用する必要があります。

---

### 使用方法

- OracleAS Guard クライアント・システムは、host-name パラメータで指定された OracleAS Guard ホスト・システムにネットワークを通してアクセスできる必要があります。
- OracleAS Guard ホスト・システムは、OracleAS Disaster Recovery 構成のすべてのシステムにネットワークを介してアクセスできる必要があります。
- 指定する ias\_admin アカウント名には、OracleAS Disaster Recovery サイトで操作を行うために必要な権利および権限（すべての必要なファイルおよびディレクトリへの読取り / 書き込みアクセスなど）が構成されている必要があります。
- ホスト名には、IP アドレスを使用することができます。
- ias\_admin または oc4jadmin アカウントのパスワードを connect コマンドで指定しないと、パスワードを入力するよう求められます。

### 例

次のコマンドは、OracleAS Guard クライアントが prodinfra というホストで実行されている OracleAS Guard サーバーに、ユーザー名 ias\_admin、パスワード adminpwd を使用して接続する例を示しています。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
```

## create standby database

スタンバイ・データベースをリモート・ホスト・システムに作成します。

### 書式

```
create standby database <database_name> on <remote_host>
```

### パラメータ

#### database\_name

リモート・ホスト・システムでのスタンバイ・データベースの作成に使用する、プライマリ・データベースの一意的な名前。

#### remote\_host

スタンバイ・データベースを作成するホスト・システムの名前。

### 使用方法

- Oracle ソフトウェアおよび OracleAS Guard ソフトウェアを、<remote\_host> に指定したノードにインストールする必要があります。
- スタンバイ・データベース用に生成される `init.ora` パラメータ・ファイルは、非 Oracle RAC 対応のスタンバイ・データベースを想定して構成されています。スタンバイ・データベースを Oracle RAC 対応にする場合は、次の初期化パラメータを適切に定義する必要があります。

```
- cluster_database
- cluster_database_instances
- remote_listener
```

- このコマンドは、必要な場合のみ慎重に使用してください。
- スタンバイ・サイトのデータベースが実行中の場合は、`asgctl` の `create standby database` コマンドを発行する前にそれを停止することをお勧めします。停止しないと次のようなエラーが返されます。

```
ASG_DGA-12500: Standby database instance "<instance_name>" already exists on host
"<hostname>"
```

- Windows では、`create standby database` コマンドを実行する前に、スタンバイ・データベースのサイトで次のコマンドを実行することをお勧めします。

```
$ORACLE_HOME%bin%oradim -delete -sid <Oracle DB SID>
```

- CRS 環境では、障害時にデータベースを自動的に起動するように CRS が構成されています。ただし、Oracle AS Guard の操作時には、この CRS の自動オンライン機能をすべての Oracle RAC インスタンスで無効にする必要があります。データベースの自動起動を無効化するには、次の 2 つのコマンドを実行します。

```
$DBHOME%bin%srvctl disable database -d ORCL
$DBHOME%bin%srvctl stop database -d ORCL
```

また、CRS 全体のデーモンは停止しないようにしてください。停止するとデータベースが起動せず、次のようなエラーが返されます。

```
SQL> startup
ORA-29702: error occurred in Cluster Group Service operation
SQL> exit
```

- Oracle RAC 環境で、2つのノード（1つのノード上に oracle1 インスタンス、もう1つのノード上に orcl2 インスタンスを持つ）を含む RAC インスタンス内の1つのプライマリ・インスタンス（たとえば orcl2）から、単一のスタンバイ・ノードである非 RAC データベースに対して asgctl の create standby database コマンドを実行すると、別の Oracle RAC インスタンスのアラート・ログにハートビート・エラー・ログが記録されます。
- Oracle RAC 環境で、プライマリ・インスタンスの1つから単一のスタンバイ・ノードである非 RAC データベースに対して asgctl の create standby database コマンドを実行すると、別の Oracle RAC インスタンスのアラート・ログに次のようなハートビートのエラー・ログが記録されます。

```
-----
PING[ARC0]: Heartbeat failed to connect to standby 'orcl2_remote1'. Error is 12154.
Mon Aug 28 09:53:43 2006
Error 12154 received logging on to the standby
Mon Aug 28 09:53:43 2006
Errors in file c:\oracle\product\10.2.0\admin\orcl1\bdump\orcl1_arc0_2752.trc:
ORA-12154: TNS:could not resolve the connect identifier specified
-----
```

このエラーは、asgctl の create standby database コマンドでホストが構成されなかったことが原因で発生します。この問題に対処するには、適切なリモート・サービス名のエントリを Oracle RAC の残りの tnsnames.ora ファイル・インスタンスに追加して、このコマンドで構成されなかった残りのホストを手動で構成する必要があります。次に、例を示します。

```
ORCL2_REMOTE1 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 140.87.23.35) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl2)
    )
  )
```

- Oracle RAC 環境では、create standby database コマンドを実行すると、Oracle RAC がクラスタリング・オプションを無効にしたスタンドアロンの非 Oracle RAC 構成に設定されます。ただし、OracleAS Guard のフェイルオーバーまたはスイッチオーバー操作後、スタンバイ（新しい本番）データベースをクラスタの一部として起動する場合は init.ora ファイルのパラメータを次のように変更します。

- cluster\_database=false を cluster\_database=true に変更します。
- cluster\_database\_instances=1 を値 n に変更します。n はクラスタ内のインスタンス数を表します。
- remote\_listener パラメータを設定します。

これらの変更は、スクリプトを使用し、asgctl の run コマンドを呼び出して設定できます。詳細は、「run」を参照してください。

## 例

次の例では、プライマリ・データベースに orcl という一意の名前を使用し、asmid1 という名前のリモート・ホスト・システムにスタンバイ・データベースを作成します。

```
ASGCTL> create standby database orcl on asmid1
```

---

## disconnect

OracleAS Guard クライアントを、現在接続している OracleAS Guard サーバーから切断します。

### 書式

```
disconnect
```

### 使用方法

このコマンドを実行する場合は、OracleAS Guard クライアントが OracleAS Guard サーバーに接続されている必要があります。

### 例

次のコマンドは、OracleAS Guard クライアントを、現在接続している OracleAS Guard サーバーから切断する例を示しています。

```
ASGCTL> disconnect  
ASGCTL>
```

## discover topology

asgctl に Oracle Internet Directory を問い合わせるように指定し、本番サイトの同じ Oracle Internet Directory を共有するトポロジ内のすべてのインスタンスを確認して、トポロジについて記述するトポロジ XML ファイルを生成します。

### 書式

```
discover topology [oidhost=<host>] [oidsslport=<sslport>] [oiduser=<user>] oidpassword=<pass>
```

### パラメータ

#### host

Oracle Internet Directory がインストールされているホスト・システムの名前。

#### sslport

Oracle Internet Directory と SSL がインストールされているホスト・システムのポート番号。

#### user

Oracle Internet Directory ユーザー名。

#### pass

指定した Oracle Internet Directory ユーザー名のパスワード。

### 使用方法

本番サイトで別の Oracle ホームを取得するときや、スイッチオーバーやフェイルオーバー操作で本番サイトからスタンバイ・サイトに役割を変更するときに、トポロジの検出操作を実行する必要があります。

トポロジの検出により、OracleAS Guard のすべての操作が実行されるトポロジが作成されます (topology.xml に保存されます)。このコマンドは、Oracle Internet Directory の情報を使用して、トポロジに含まれるインスタンスを定義します。さらに、各インスタンスについてのローカルの情報を収集します。そのため、すべての本番サイトのインスタンスで OPMN が実行されている必要があります。DCM フェームを使用して管理されていないインスタンスの場合、Oracle ホームの OracleAS Guard サービスを起動する必要があります。ローカルでサービスが起動されていない場合、警告が発せられ、topology.xml ファイルには検出されたインスタンスしか含まれなくなります。

詳細は、[第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」](#) および [第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

### 例

次のコマンドは、本番サイトの同じ Oracle Internet Directory を共有するトポロジ内のすべてのインスタンスを検出し、トポロジについて記述するトポロジ XML ファイルを生成する例を示しています。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> discover topology oidpassword=oidpwd
Discovering topology on host "infra" with IP address "123.1.2.111" prodinfra:7890
  Connecting to the OID server on host "infra.us.oracle.com" using SSL port "636" and
  username "orcladmin"
    Getting the list of databases from OID
    Gathering database information for SID "asdb" from host "infra.us.oracle.com"
    Getting the list of instances from OID
    Gathering instance information for "asr1012.infra.us.oracle.com" from host
  "infra.us.oracle.com"
    Gathering instance information for "asmid1.asmid1.us.oracle.com" from host
```

```
"asmid1.us.oracle.com"
```

```
    Gathering instance information for "asmid2.us.oracle.com" from host
```

```
"asmid2.us.oracle.com"
```

```
The topology has been discovered. A topology.xml file has been written to each home in  
the topology.
```

```
ASGCTL>
```

---

## discover topology within farm

ファームで使用可能な Oracle Internet Directory がないという特殊なケースでは、本番サイトのファーム内でトポロジを検出するように asgctl に指定します。

---

**注意：** サイトのトポロジを検出する場合は、必ず `discover topology` コマンドを使用する必要があります。このコマンドは Oracle Internet Directory を使用してトポロジのすべてのインスタンスを検出するためです。Oracle Internet Directory が使用できないような特殊なケースでのみ、`discover topology within farm` コマンドが有用です。この場合、OracleAS Guard は OPMN を使用して、ファーム内のトポロジを検出します。

---

### 書式

`discover topology within farm`

### パラメータ

なし。

### 使用方法

このコマンドを実行する場合は、OracleAS Guard クライアントが OracleAS Guard サーバーに接続されている必要があります。

トポロジ・ファイルの詳細は、[第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#)を参照してください。

### 例

次のコマンドは、Oracle Internet Directory が使用できない特殊なケースで、OPMN を使用して OracleAS Guard クライアントが現在接続している OracleAS Guard サーバーのファーム内のアプリケーション・サーバーのトポロジを検出する例を示しています。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> discover topology within farm
Warning: If OID is part of your environment, you should use it for discovery
Discovering topology on host "infra" with IP address "123.1.2.111"
prodinfra:7890
    Discovering instances within the topology using OPMN
    Gathering instance information for "asr1012.infra.us.oracle.com" from host
"infra.us.oracle.com"
The topology has been discovered. A topology.xml file has been written to each home in
the topology.
ASGCTL>
```

## dump policies

asgctl の各種コマンドの詳細なデフォルト・ポリシー情報を XML 形式出力で一連のポリシー・ファイルに書き込むように OracleAS Guard サーバーに指定します。ポリシー・ファイルは、UNIX システムの場合 <ORACLE\_HOME>/dsa/conf ディレクトリ、Windows システムの場合 <ORACLE\_HOME>%dsa%conf ディレクトリのローカル・ホストにあります。

### 書式

dump policies

### パラメータ

なし。

### 使用方法

clone topology、dump topology、failover、instantiate topology、sync topology、switchover topology、および verify topology の各 asgctl コマンドに対して、XML 形式のポリシー・ファイルが書き込まれます。それぞれのコマンドのポリシー・ファイルを編集して、当該のコマンドに対して using policy <file> 句でこれを指定できます。このパラメータにより、OracleAS Guard の各操作に対してトポロジの障害時リカバリ・ポリシーを定義できます。

ダンプのポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンス（中間層および OracleAS Metadata Repository）でオプションに設定されています。

フェイルオーバーのポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンス（中間層および OracleAS Metadata Repository）でオプションに設定され、Oracle Internet Directory ホームでは必須に設定されています。

インスタンス化のポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンスで必須に設定されています。

スイッチオーバー・ポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンス（中間層および OracleAS Metadata Repository）でオプションに設定され、Oracle Internet Directory ホームでは必須に設定されています。

同期化のポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンスで必須に設定されています。

検証のポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンス（中間層および OracleAS Metadata Repository）でオプションに設定され、Oracle Internet Directory ホームでは必須に設定されています。

### 例

次の例では、asgctl の各種コマンドの詳細なデフォルト・ポリシー情報を XML 形式出力でローカル・ホストにある一連の各ポリシー・ファイルに書き込みます。

```
ASGCTL> dump policies
Generating default policy for this operation
Creating policy files on local host in directory "/private1/OraHome2/asr1012/dsa/conf/"
ASGCTL>
```

## dump topology

asgctl に、トポロジに関する詳細な情報を、指定したファイルに書き込むように指示します。

### 書式

```
dump topology [to <file>] [using policy <file>]
```

### パラメータ

#### to <file>

詳細情報の出力の書き込み先である、OracleAS Guard クライアント・ノード内のファイルの名前。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

### 使用方法

ダンプのポリシー・ファイルでは、デフォルトで成功要件属性はすべての OracleAS ホーム（中間層および OracleAS Metadata Repository）でオプションに設定されています。

### 例

次の例では、トポロジに関する詳細情報をローカル・ファイルに書き込んでいます。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> dump topology to c:\%dump_mid_1.txt
```

Contents of file c:\%dump\_mid\_1.txt are:

Generating default policy for this operation

```
Instance: asr1012.infra.us.oracle.com
Type: Infrastructure
Oracle Home Name: asr1012
Oracle Home Path: /private1/OraHome
Version: 10.1.2.0.2
OidHost: infra.us.oracle.com
OidPort: 389
VirtualHost: infra.us.oracle.com
Host: prodinfra
Ip: 123.1.2.111
Operation System Arch: sparc
Operation System Version: 5.8
Operation System Name: SunOS
```

```
Instance: asmid2.asmid2.us.oracle.com
Type: Core
Oracle Home Name: asmid2
Oracle Home Path: /private1/OraHome2
Version: 10.1.2.0.2
OidHost: infra.us.oracle.com
OidPort: 389
VirtualHost: asmid2.us.oracle.com
Host: asmid2
Ip: 123.1.2.333
```

```
Operation System Arch: sparc
Operation System Version: 5.8
Operation System Name: SunOS
```

```
Instance: asmid1.asmid1.us.oracle.com
Type: Core
Oracle Home Name: asmid1
Oracle Home Path: /private1/OraHome
Version: 10.1.2.0.2
OidHost: infra.us.oracle.com
OidPort: 389
VirtualHost: asmid1.us.oracle.com
Host: asmid1
Ip: 123.1.2.334
Operation System Arch: sparc
Operation System Version: 5.8
Operation System Name: SunOS
ASGCTL>
```

次の例では、トポロジに関する詳細情報をローカル・ファイルに書き込んでいます。出力する必要のないインスタンスはポリシー・ファイルで指定できます。

```
# Command to use if you are using a policy file
ASGCTL> dump topology to c:\%dump_mid_1.txt using policy <file>
```

## exit

OracleAS Guard サーバーとの既存の接続を切断し、OracleAS Guard クライアントを終了します。

### 書式

exit

### パラメータ

なし。

### 使用方法

なし。

### 例

```
ASGCTL> exit  
>
```

## failover

本番サイトの計画外停止の間に、スタンバイ・サイトでフェイルオーバー操作を実行してこれを本番サイトにします。

### 書式

```
failover [using policy <file>]
```

### パラメータ

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

### 使用方法

フェイルオーバー操作を実行する前に、OracleAS Infrastructure データベースがスタンバイ・トポロジで実行されていることを確認する必要があります。また、asgctl の set new primary database コマンドによって、OracleAS Infrastructure データベースの情報が設定されている必要があります。

グローバル DNS 名を使用してフェイルオーバーを実行することもできます。この場合、OracleAS Disaster Recovery 環境で利用されている高可用性のネーミングと異なることとなります。検出メカニズムが、ローカルの名前解決に従って自動的にトポロジを対応するピアにマップします。

フェイルオーバーのポリシー・ファイルでは、デフォルトで成功要件属性はすべての OracleAS ホーム（中間層および OracleAS Metadata Repository）でオプションに設定され、Oracle Internet Directory ホームでは必須に設定されています。

詳細は、[第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」](#) および [第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

### 例

次の例では、スタンバイ・サイトにフェイルオーバー操作を実行しています。

```
ASGCTL> connect asg standbyinfra ias_admin/adminpwd
Successfully connected to standbyinfra:7890
ASGCTL> set new primary database sys/testpwd@asdb
ASGCTL> failover
Generating default policy for this operation
standbyinfra:7890
    Failover each instance in the topology from standby to primary topology
standbyinfra:7890 (home /private1/OraHome2/asr1012)
    Shutting down each instance in the topology
.
.
.
    Executing opmnctl startall command
standbyinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
ASGCTL>

# Command to use if you are using a policy file
# failover using policy <file>
```

## help

ヘルプ情報を表示します。

## 書式

help [<command>]

## パラメータ

### command

ヘルプ情報を表示するコマンドの名前。

## 使用方法

なし。

## 例

次の例では、すべてのコマンドに関するヘルプ情報を表示しています。

```
ASGCTL> help
connect asg [<host>] [<ias_administrator_account>/<password>]
disconnect
exit
quit
add instance <instance_name> on <instance_host> [to topology]
clone topology to <standby_topology_host> [using policy <file>] [no standby]
clone instance <instance> to <standby_topology_host> [no standby]
discover topology [oidhost=<host>] [oidsslport=<sslport>] [oiduser=<user>] oidpassword=<pass>
discover topology within farm
dump farm [to <file>] (Deprecated)
dump topology [to <file>] [using policy <file>]
dump policies
failover [using policy <file>]
help [<command>]
instantiate farm to <standby_farm_host> (Deprecated)
instantiate topology to <standby_topology_host> [using policy <file>]
remove instance <instance_name> [from topology]
set asg credentials <host> <ias_administrator_account>/<password> [for topology]
set asg credentials <host> ias_admin/<password> [for farm] (Deprecated)
set primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]
set new primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]
set noprompt
set trace on|off <traceflags>
sync farm to <standby_farm_host> [full | incr[emental]] (Deprecated)
sync topology to <standby_topology_host> [full | incr[emental]] [using policy <file>]
startup [asg]
startup farm (Deprecated)
startup topology
shutdown [local]
shutdown farm (Deprecated)
shutdown topology
show op[eration] [full] [[his]tory]
show env
stop op[eration] <op#>
switchover farm to <standby_farm_host> (Deprecated)
switchover topology to <standby_topology_host> [using policy <file>]
verify farm [with <host>] (Deprecated)
verify topology [with <host>] [using policy <file>]
ASGCTL>
```

## instantiate topology

スタンバイ・インスタンスと本番インスタンス間の関係を確立し、構成をミラー化して、スタンバイ Infrastructure を作成した後、スタンバイ・サイトをプライマリ・サイトに同期化して、トポロジをスタンバイ・サイトにインスタンス化します。

### 書式

```
instantiate topology to <standby_topology_host>[:<port>] [with cloning] [using policy <file>]
```

### パラメータ

#### standby\_topology\_host

スタンバイ・ホスト・システムの名前。このパラメータは、OracleAS Guard 調整サーバー・インスタンスを指定して、スタンバイ・サイトを構成するインスタンスを検出するために必要です。ホスト・システムは、スタンバイ・トポロジのメンバーである必要があります。

#### port

Oracle ホーム内の OracleAS Guard サーバーのポート番号。

#### with cloning

クローニングを使用してインスタンス化操作を実行するためのディレクティブ。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

### 使用方法

トポロジのインスタンス化操作を実行する前に、OracleAS Infrastructure データベースがプライマリ・トポロジで実行されていることを確認する必要があります。また、`asgctl` の `set primary database` コマンドによって、OracleAS Infrastructure データベースの情報が設定されている必要があります。

グローバル DNS 名を使用してインスタンス化を実行することもできます。この場合、OracleAS Disaster Recovery 環境で利用されている高可用性のネーミングと異なることとなります。検出メカニズムが、ローカルの名前解決に従って自動的にトポロジを対応するピアにマップします。

インスタンス化操作により、暗黙の検証操作が実行されます。

インスタンス化のポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンスで必須に設定されています。

詳細は、[第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」](#) および [第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

### 例

次の例では、スタンバイ・トポロジをインスタンス化しています。具体的には、OracleAS Guard 調整サーバーへの接続、本番サイトとスタンバイ・サイトのトポロジの検出、およびサイトの検証を行い、DNS 名が `standbyinfra` のスタンバイ・トポロジ・ホストを含むトポロジに OracleAS Disaster Recovery 環境を確立しています。操作の途中で、データベースを停止するかどうかについて質問されます。返答として、`y` または `yes` を入力します。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> instantiate topology to standbyinfra
Generating default policy for this operation
prodinfra:7890
```

```
Instantiating each instance in the topology to standby topology
  HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
  HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
  HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
  HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
  HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
  HA directory exists for instance asmid1.asmid1.us.oracle.com
asmid2:7890
  Verifying that the topology is symmetrical in both primary and standby configuration
.
.
.
This operation requires the database to be shutdown. Do you want to continue? Yes or No
Y
.
.
.
asmid2:7890 (home /private1/oracle/asr1012)
  Starting backup/synchronization of database "orcl.us.oracle.com"
  Starting restore/synchronization of database "orcl.us.oracle.com"
  Synchronizing topology completed successfully
asmid2:7890
  Synchronizing topology completed successfully

ASGCTL>

# Command to use if you are using a policy file
# instantiate topology to standbyinfra using policy <file>
```

---

## quit

OracleAS Guard クライアントに、既存の接続を切断し、asgctl を終了することを指示します。

### 書式

quit

### パラメータ

なし。

### 使用方法

なし。

### 例

次の例では、asgctl を終了しています。

```
ASGCTL> quit  
>
```

## remove instance

指定されたインスタンス名をローカル・トポロジ・ファイルから削除し、指定に応じて、この更新されたトポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播します。

### 書式

```
remove instance <instance_name> [from topology]
```

### パラメータ

#### **instance\_name**

トポロジ・ファイルから削除されるインスタンスの名前。

#### **from topology**

キーワード。コマンドラインにこのキーワードを指定した場合、更新されたトポロジ・ファイルを Disaster Recovery 本番環境のインスタンスすべてに伝播するように OracleAS Guard を指定します。検証、インスタンス化、同期化、およびスイッチオーバーなど、スタンバイ・サイトに影響を与える OracleAS Guard 操作を実行すると、スタンバイ環境全体に本番トポロジ・ファイルが自動的に伝播されます。

### 使用方法

このコマンドは、OracleAS Guard クライアントが接続されている OracleAS Guard サーバー上のインスタンスの管理に役立ちます。たとえば、このローカル・ホスト・システム上のインスタンスに問題があるために、[remove instance](#) コマンドを使用して、ローカル・トポロジ・ファイルから、またはトポロジ内部のすべてのトポロジ・ファイルから、そのインスタンスを削除する場合があります。そこで、ローカル・トポロジ・ファイルまたはトポロジ内のトポロジ・ファイルすべてに、適切なインスタンスを追加することにします。この場合、[asgctl](#) コマンドを起動してトポロジ全体に対して実行したとき、ポリシー・ファイルではこのインスタンスについて成功要件属性が Ignore に設定された可能性があるため、そのポリシー・ファイルは不適切なインスタンスの管理に使用するには好ましくないことがあります。

このコマンドは、OID が利用できない Disaster Recovery ファーム、つまり OracleAS 10g リリース 3 (10.1.3) のみのトポロジの管理に特に役立ちます。このファーム内の各インスタンスにトポロジ・ファイルを最初に作成するには、[discover topology within farm](#) コマンドを使用する必要があります。その後、[add instance](#) および [remove instance](#) コマンドを使用して、ローカル・トポロジ・ファイルで個別のインスタンスを追加または削除することで、インスタンスを管理できるようになります。to topology または from topology キーワードを指定した場合、更新済のローカル・トポロジ・ファイルの変更が Disaster Recovery 環境におけるインスタンスすべてに伝播します。トポロジ・ファイルの詳細は、[第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#)を参照してください。

このコマンドは、OracleAS 10.1.3 J2EE インスタンスを OID ベースの 10.1.2.0.2 トポロジに追加して、バージョンが混在した Disaster Recovery 環境をサポートする場合に役立ちます。たとえば、[add instance](#) コマンドを使用して、OracleAS 10.1.3 J2EE インスタンスを OID ベースの 10.1.2.0.2 トポロジに追加できます。ユースケースについては、[第 6.9 項「既存の Oracle Identity Management 10.1.2.0.2 トポロジと統合されている冗長な単一の OracleAS 10.1.3 ホーム J2EE トポロジでの OracleAS 10.1.3 インスタンスの追加または削除」](#)を参照してください。トポロジ・ファイルの詳細は、[第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#)を参照してください。

またこのコマンドは、リリース 9.0.4 からリリース 10.1.3 までの任意の OracleAS リリースに対して、バージョンが混在したトポロジをサポートするためにも役立ちます。唯一の要件は、リリース 10.1.3 の OracleAS Guard スタンドアロン・キットが、リリース 9.0.4 の Disaster Recovery 環境におけるすべてのシステムにインストールされていることです。詳細は、[第 9 章「OracleAS Disaster Recovery サイトのアップグレード手順」](#)を参照してください。

**例**

次の例では、ローカル・トポロジ・ファイルから `oc4j1` というインスタンスを削除します。

```
ASGCTL> remove instance oc4j1
```

---

## run

OracleAS Guard がインストールされている任意のホームにあるスクリプトまたはプログラムをリモートで実行します。run コマンドは、トポロジ内または指定したインスタンスで実行できます。

### 書式

```
run [at topology [using policy <file>]] <command>
```

```
run [at instance <instance_name>] <command>
```

### パラメータ

#### at topology

キーワード。コマンドラインにこのキーワードを指定すると、run 操作がトポロジ全体で実行されます。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

#### command

実行するスクリプトまたはバイナリ・プログラムのコマンド文字列としての名前。

#### at instance

キーワード。コマンドラインにこのキーワードを指定すると、run 操作が指定したインスタンスで実行されます。

#### instance\_name

run コマンドを実行するインスタンスの名前。

### 使用方法

このコマンドは、OracleAS Guard がインストールされている任意の Oracle ホームにあるスクリプトまたはプログラムをリモートで実行する場合に役立ちます。このスクリプトおよびプログラムは、トポロジ内、または指定した実行場所であるインスタンスの各 Oracle ホームに物理的に存在している必要があります。asgctl ユーザーは、asgctl の run コマンドを呼び出す前に、まず OracleAS Guard サーバーに接続して Application Server の JAZN 資格証明 (ias\_admin または oc4jadmin) を指定する必要があります。ユーザーが JAZN 資格証明を知っている場合、そのユーザーはホームでスクリプトまたはプログラムを実行できると判断されます。run コマンドの呼び出しを受信すると、OracleAS Guard では、スクリプトまたはプログラムを実行する前に、OracleAS Guard サーバーが稼働する Oracle ホームにコマンド文字列で指定したファイルが存在するかどうか確認されます。スクリプトの出力は asgctl コンソールにエコーされます。

### 例

Oracle RAC の障害時リカバリの配置では、スイッチオーバー操作を実行する前にすべてのインスタンスを停止します。これを行うには、スクリプトを作成し、run コマンドを使用してそのスクリプトを実行します。詳細は、第 6.12.1.1 項「スケジューリングした停止」の手順 5 を参照してください。次のコマンドの例は、asdb というインスタンス用にスクリプト shutdown\_asdb\_instance.sh を記述し、それをリモートで実行することを想定しています。このスクリプトでは、分散された ASG のスクリプティング機能を利用しています。これにより、システム管理者は asgctl ユーティリティからスイッチオーバー操作を実行できます。

```
ASGCTL> run at instance asdb shutdown_asdb_instance.sh
```

## set asg credentials

OracleAS Guard サーバーへの OracleAS Guard の接続を認証する資格証明を設定します。

### 書式

```
set asg credentials <host>[:<port>] <ias_administrative_account>/<password> [for farm] [for topology]
```

### パラメータ

#### host

資格証明を適用するホスト・システムの名前。OracleAS Guard がこのホストに接続する場合に、資格証明が使用されます。

#### port

Oracle ホーム内の OracleAS Guard サーバーのポート番号。

#### <ias\_administrative\_account>/password

OracleAS 10.1.3 インストールの場合、ユーザー名は oc4jadmin を使用し、この oc4jadmin アカウントのパスワードは Oracle Application Server 10.1.3 のインストール時に作成されたものを使用する必要があります。OracleAS 10.1.2.0.2 以前のインストールの場合、ユーザー名は ias\_admin を使用し、この ias\_admin アカウントのパスワードは Oracle Application Server のインストール時に作成されたものを使用する必要があります。このアカウント名は、Oracle Application Server ホームの最低 1 つに属するアカウント名と同じにする必要があります。

#### for farm (廃止済)

キーワード。コマンドラインにこのキーワードを指定した場合、ローカル・ホスト・システムと同じファームに属するすべてのホスト・システムに資格証明が設定されます。

#### for topology

キーワード。コマンドラインにこのキーワードを指定した場合、ローカル・ホスト・システムと同じトポロジに属するすべてのホスト・システムに資格証明が設定されます。

### 使用方法

デフォルトでは、asgctl connect コマンドで使用した資格証明は、OracleAS Guard サーバーが他の OracleAS Guard サーバーに接続する際に必ず使用されます。しかし、特定のサーバーに異なる資格証明を使用することが必要な場合もあります。このコマンドを使用すると、トポロジ内のすべてのノードに同じ資格証明を使用できます。たとえば、プライマリ・トポロジで使用されている資格証明とは異なる、資格証明のセットをスタンバイ・トポロジで共有することができます。

あるトポロジに資格証明を設定した場合、その資格証明はトポロジ全体に継承されます。トポロジ上の個別のホストに資格証明を設定した場合、(そのホストの) 資格証明が、トポロジに設定されたデフォルトの資格証明よりも優先されます。

Oracle Internet Directory と OracleAS Metadata Repository が同じ場所にあって、Portal の OracleAS Metadata Repository が別の場所にあるような、複数の Infrastructure が存在するトポロジでは、OracleAS Guard でインスタンス化、同期化、スイッチオーバー、フェイルオーバーなどの重要な操作を実行する前に、Infrastructure が存在する各システムに対して OracleAS Guard で資格証明を設定する必要があります。これは実際は 2 段階のプロセスで、最初に各 Infrastructure に set primary database コマンドを使用してトポロジのすべての OracleAS Infrastructure データベースを特定し、次にその Infrastructure が存在する OracleAS Guard サーバーに OracleAS Guard を接続するときの認証に使用する資格証明を設定する必要があります。次の例は、この概要を示します。本番トポロジとスタンバイ・トポロジは、Infrastructure と中間層ソフトウェア・アプリケーションがインストールされている次のようなシステムで構成されているものとします。

本番トポロジ:

host01 (Identity Management+OracleAS Metadata Repository)、host04 (OracleAS Metadata Repository のみ)、host06 (J2EE)、host06 (Portal & Wireless)

スタンバイ・トポロジ:

host02 (Identity Management+OracleAS Metadata Repository)、host05 (OracleAS Metadata Repository のみ)、host07 (J2EE)、host07 (Portal & Wireless)

次の OracleAS Guard の set primary database と set asg credentials コマンドは、インスタンス化、同期化、スイッチオーバー、フェイルオーバー操作を実行する前に、Infrastructure を正しく特定し、OracleAS Guard の OracleAS Guard サーバーへの接続を認証する必要があります。Oracle Identity Management+OracleAS Metadata Repository Infrastructure が orcl というサービス名を持ち、Portal OracleAS Metadata Repository が asdb というサービス名を持つものとします。

```
ASGCTL> set primary database sys/<password>@orcl.us.oracle.com
ASGCTL> set primary database sys/<password>@asdb.us.oracle.com
ASGCTL> set asg credentials host01.us.oracle.com ias_admin/<password>
ASGCTL> set asg credentials host04.us.oracle.com ias_admin/<password>
```

フェイルオーバー操作の場合、これらの手順はスタンバイ・トポロジで実行し、ホスト・システム名は次のように変更します。

```
ASGCTL> set primary database sys/<password>@orcl.us.oracle.com
ASGCTL> set primary database sys/<password>@asdb.us.oracle.com
ASGCTL> set asg credentials host02.us.oracle.com ias_admin/<password>
ASGCTL> set asg credentials host05.us.oracle.com ias_admin/<password>
```

このコマンドを使用する前に、OracleAS Guard クライアントが OracleAS Guard サーバーに接続されている必要があります。

ホスト名には、IP アドレスを使用することができます。

詳細は、第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」および第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。

## 例

次の例では、ホスト・システム standbyinfra に OracleAS Guard の資格証明を設定し、それを同じトポロジ内のすべてのホスト・システムに適用しています。

```
ASGCTL> set asg credentials standbyinfra ias_admin/<password> for topology
```

---

## set echo

asgctl スクリプトのコマンドエコーのオンおよびオフを設定します。

### 書式

```
set echo on | off
```

### パラメータ

#### on | off

on を指定すると、asgctl スクリプトのコマンドエコーが有効になります。off を指定すると、asgctl スクリプトのコマンドエコーが無効になります。

### 使用方法

このコマンドは、大きな asgctl スクリプトを実行する場合に役立ちます。たとえば、asgctl スクリプトにエラー・テスト・ケースがあり、各テスト・ケースの前または各 asgctl コマンドの前にコメントを入力した場合には、エコーを on に設定しておくことで、各テスト・ケースまたは各 asgctl コマンドが実行される前にコメントが表示され、そのテスト・ケースの目的や asgctl コマンドを実行した場合の影響などを確認できます。

このコマンドはネストされたスクリプトにも使用できます。

### 例

次の例は、コマンドエコーの有効化、テスト・ケースの実行、OracleAS Guard サーバーへの接続、トポロジに関する詳細情報の表示、エコーの無効化、OracleAS Guard サーバーの切断、および OracleAS Guard クライアントの終了を行う asgctl スクリプトを示しています。

```
> ASGCTL @myasgctltestscript.txt

# myasgctltestscript.txt
# turn on echo
set echo on

# make sure you are not connected
disconnect

# not connected, should get an error message
dump topology

# connect to an ASG server
connect asg prodinfra ias_admin/adminpwd

#display detailed info about the topology
dump topology

#disconnect
disconnect

# turn off echo
echo off
exit
```

## set new primary database

フェイルオーバー操作の前に、スタンバイ・トポロジの OracleAS Infrastructure データベースを新しいプライマリ・データベースとして指定します。このコマンドは、フェイルオーバー操作の際にのみ使用します。

### 書式

```
set new primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]
```

### パラメータ

#### username/password

sysdba の権限を持つデータベース・アカウントのユーザー名とパスワード。

#### servicename

OracleAS Infrastructure データベースの TNS サービス名。TNS サービス名は OracleAS Infrastructure ホスト・システムで定義されている必要がありますが、OracleAS Guard クライアント・ホスト・システムで定義されている必要はありません。

#### pfile filename

プライマリ (OracleAS Infrastructure) データベースの初期化ファイルのファイル名。プライマリ・データベースが起動されるときに使用されます。

#### spfile filename

サーバー (OracleAS Infrastructure) の初期化ファイルのファイル名。データベースが起動されるときに使用されます。

### 使用方法

フェイルオーバー操作を実行する前に、スタンバイ・トポロジの Infrastructure ノードに接続し、新しいプライマリ・データベースを定義する必要があります。スタンバイ・サイトの Oracle Infrastructure データベースを新しいプライマリ・データベースとして指定したら、フェイルオーバー操作を開始できます。

### 例

次の例では、フェイルオーバー操作の前に、スタンバイ・トポロジの OracleAS Infrastructure データベースの情報を新しいプライマリ / 本番トポロジとして設定しています。

```
ASGCTL> connect asg standbyinfra ias_admin/adminpwd
Successfully connected to standbyinfra:7890
ASGCTL> set new primary database sys/testpwd@asdb
ASGCTL> failover
.
.
.
ASGCTL>
```

---

## set noprompt

asgctl スクリプトでコマンドを実行するときのユーザーとの対話をプロンプトのない状態に設定します。

### 書式

```
set noprompt
```

### パラメータ

なし。

### 使用方法

設定されると、すべての対話プロンプトにデフォルト値が適用されます。ユーザー名とパスワードのプロンプトは、プロンプトなしの状態ではエラー・メッセージが返されます。

### 例

次の例は、asgctl の set noprompt コマンドを含む asgctl スクリプトです。このコマンド以降のすべての対話プロンプトは無視されてスクリプトが実行されます。

```
> ASGCTL @myasgctltestscript.txt

# myasgctltestscript.txt

# connect to an ASG server
connect asg prodinfra ias_admin/adminpwd

# set the primary database
set primary database sys/testpwd@asdb

# discover the production topology
discover topology oidpassword=oidpwd

# set the noprompt state
set noprompt

#display detailed info about the topology
dump topology

#disconnect
disconnect

exit
```

## set primary database

プライマリ・トポロジの OracleAS Infrastructure データベースを指定します。

### 書式

```
set primary database <username>/<password>@<servicename> [pfile <filename> | spfile <filename>]
```

### パラメータ

#### username/password

sysdba の権限を持つデータベース・アカウントのユーザー名とパスワード。

#### servicename

OracleAS Infrastructure データベースの TNS サービス名。TNS サービス名は OracleAS Infrastructure ホスト・システムで定義されている必要がありますが、OracleAS Guard クライアント・ホスト・システムで定義されている必要はありません。

#### pfile filename

プライマリ (OracleAS Infrastructure) データベースの初期化ファイルのファイル名。プライマリ・データベースが起動されるときに使用されます。

#### spfile filename

サーバー (OracleAS Infrastructure) の初期化ファイルのファイル名。データベースが起動されるときに使用されます。

### 使用方法

プライマリ・データベースは、インスタンス化、同期化またはスイッチオーバー操作を実行する前に指定する必要があります。

プライマリ・データベースを設定すると、OracleAS Guard サーバーはそのデータベースにログインして接続を検証します。

本番サイトまたはスタンバイ・サイトに複数の OracleAS Metadata Repository のインスタンスがインストールされているときに、インスタンス化、同期化、スイッチオーバー、またはフェイルオーバー操作を実行する場合、実行前に **set primary database** コマンドを実行して、すべての OracleAS Metadata Repository インスタンスを特定する必要があります。また、Oracle Internet Directory と OracleAS Metadata Repository が同じ場所にあつて、Portal の OracleAS Metadata Repository が別の場所にあるような、複数の Infrastructure が存在するトポロジでは、OracleAS Guard でインスタンス化、同期化、スイッチオーバー、フェイルオーバーなどの重要な操作を実行する前に、Infrastructure が存在する各システムに対して OracleAS Guard で資格証明を設定する必要があります。「[set asg credentials](#)」に、例を示します。

OracleAS Guard は、データベースにパスワード・ファイル認証がある必要があります。データベースにパスワード・ファイルがない場合、`orapwd` ユーティリティを使用してパスワード・ファイルを作成する必要があります。また、`REMOTE_LOGIN_PASSWORDFILE` 初期化パラメータを `EXCLUSIVE` に設定します。

詳細は、[第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」](#) および [第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

## 例

次の例では、OracleAS Infrastructure データベースの情報をプライマリ / 本番トポロジに設定しています。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL>
```

次の例では、スイッチオーバー操作の前にプライマリ / 本番トポロジにインストールする各 OracleAS Metadata Repository の OracleAS Infrastructure データベースの情報を設定しています。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@portal_1
Checking connection to database portal_1
ASGCTL> set primary database sys/testpwd@portal_2
Checking connection to database portal_2
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> discover topology oidpassword=oidpwd
ASGCTL> switchover topology to standbyinfra
.
.
.
```

## set trace

トレース出力を OracleAS Guard ログ・ファイルに記録するための、トレース・フラグのオンまたはオフを設定します。

### 書式

```
set trace on | off <traceflags>
```

### パラメータ

#### on | off

on を指定するとトレースが有効になります。off を指定するとトレースが無効になります。

#### traceflags

有効にするトレースフラグ。2つ以上のトレースフラグを指定する場合は、カンマ (,) で区切ります。トレースフラグには次のものがあります。

- DB: Oracle Database 環境の処理に関するトレース情報
- HOME: Oracle ホームに関するトレース情報
- IAS: Oracle Application Server の処理に関するトレース情報
- OPMN: OracleAS OPMN コールへのアクセスに関するトレース情報
- IP: ネットワーク・アクセスとアドレス変換に関するトレース情報
- CLIPBOARD: クリップボード処理に関するトレース情報
- COPY: ファイルのコピー処理に関するトレース情報
- FLOW: ワーク・フロー処理に関するトレース情報
- NET: ネットワーク処理に関するトレース情報
- RUNCMD: 外部コマンドの実行に関するトレース情報
- SESSION: セッション管理に関するトレース情報
- TOPOLOGY: トポロジ情報の処理に関するトレース情報

### 使用方法

このコマンドは、接続中に asgctl コマンドにかかわったすべてのホストに適用されます。

このコマンドを使用する前に、OracleAS Guard クライアントが OracleAS Guard サーバーに接続されている必要があります。

### 例

次の例では、データベース操作のトレースを有効にしています。

```
ASGCTL> set trace on db
```

---

## show env

OracleAS Guard クライアントが接続している OracleAS Guard サーバーの現在の環境を示します。

### 書式

```
show env
```

### パラメータ

なし。

### 使用方法

なし。

### 例

次の例は、OracleAS Guard クライアントが接続している OracleAS Guard サーバーの環境を示します。最初の例ではプライマリ・データベースと新しいプライマリ・データベースがホスト `prodinfra` に設定されておらず、2 番目の例ではプライマリ・データベースがホスト `standbyinfra` に設定されています。

例 1

```
ASGCTL> show env
```

```
ASG Server Connection:
  Host: prodinfra
  Port: 7890

Primary database: <not set>
New primary database: <not set>
```

例 2

```
ASGCTL> ASGCTL> show env
```

```
ASG Server Connection:
  Host: standbyinfra
  Port: 7890

Gathering information from the database orcl

Primary database: :
  User: sys
  Service: orcl
  Role: The database role is
        PHYSICAL STANDBY

New primary database: <not set>
```

## show operation

現在のセッションで OracleAS Guard クライアントが接続されたトポロジ内の全ノードのすべての操作を表示します。

### 書式

```
show op[eration] [full] [[his]tory]
```

### パラメータ

#### full

すべての操作に対して、操作番号、ジョブ名、ジョブ所有者のユーザー名、ジョブ ID、操作の開始時刻、操作の終了時刻、操作の経過時間およびジョブに含まれるすべてのタスクが表示されます。

#### history

現在実行されていない操作のみに対する、操作番号とジョブ名が表示されます。

### 使用方法

なし。

### 例

次の例では、現在の操作の状態を表示しています。

```
ASGCTL> show operation
*****
OPERATION: 19
  Status: running
  Elapsed Time: 0 days, 0 hours, 0 minutes, 28 secs
  TASK: syncFarm
    TASK: backupFarm
      TASK: fileCopyRemote
      TASK: fileCopyRemote
    TASK: restoreFarm
      TASK: fileCopyLocal
```

次の例では、すべての操作の履歴を表示しています。

```
ASGCTL> show op his
*****
OPERATION: 7
  Status: success
  Elapsed Time: 0 days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
*****
OPERATION: 16
  Status: success
  Elapsed Time: 0
  days, 0 hours, 0 minutes, 0 secs
  TASK: getTopology
    TASK: getInstance
*****
OPERATION: 19
  Status: success
  Elapsed Time: 0 days, 0 hours, 1 minutes, 55 secs
  TASK: syncFarm
    TASK: backupFarm
```

```
TASK: fileCopyRemote  
TASK: fileCopyRemote  
TASK: restoreFarm  
TASK: fileCopyLocal
```

## shutdown

OracleAS Guard クライアントが接続している稼働中の OracleAS Guard サーバーを停止します。このコマンドは、OPMN が実行されていないホスト・システムで、インスタンスまたはトポロジのクローニング手順を実行する場合にのみ使用します。

### 書式

```
shutdown [local]
```

### パラメータ

**local**

これを指定すると、asgctl のローカル Oracle ホームの OracleAS Guard サーバーを停止します。

### 使用方法

OracleAS Guard サーバーは、OPMN opmnctl コマンドの startproc ではなく、asgctl の startup コマンドを使用して起動する必要があります。

### 例

次の例では、OPMN が実行されていないホスト・システムで、OracleAS Guard サーバーを停止しています。

```
> asgctl.sh shutdown
```

---

## shutdown topology

トポロジ全体の OracleAS コンポーネント・サービスを停止します。OracleAS Guard サーバーと OPMN は稼働し続けます。

### 書式

```
shutdown topology
```

### パラメータ

なし。

### 使用方法

これは、トポロジ全体を停止するときに便利なコマンドです。トポロジを再起動する場合は、`startup topology` コマンドを使用します。

このコマンドで、OID、OC4J、WebCache などの OracleAS サービスを停止します。

### 例

次の例では、`prodinfra` 本番トポロジを停止しています。

```
ASGCTL> shutdown topology
Generating default policy for this operation

prodinfra:7890
  Shutting down each instance in the topology

asmid2:7890 (home /private1/OraHome2/asmid2)
  Shutting down component HTTP_Server
  Shutting down component OC4J
  Shutting down component dcm-daemon
  Shutting down component LogLoader

asmid1:7890 (home /private1/OraHome/asmid1)
  Shutting down component HTTP_Server
  Shutting down component OC4J
  Shutting down component dcm-daemon
  Shutting down component LogLoader

prodinfra:7890 (home /private1/OraHome2/asr1012)
  Shutting down component OID
  Shutting down component HTTP_Server
  Shutting down component OC4J
  Shutting down component dcm-daemon
  Shutting down component LogLoader
ASGCTL>
```

## startup

asgctl プロンプトで OracleAS Guard サーバーを起動します。このコマンドは、OPMN が実行されていないホスト・システムで、インスタンスまたはトポロジのクローニング手順を実行する場合にのみ使用します。

### 書式

startup [asg]

### パラメータ

#### **asg**

オプションのキーワードで、application server guard の頭文字。このパラメータには、connect asg や set asg credentials コマンドと類似の形式を示す以外に意味はありません。

### 使用方法

なし。

### 例

次の例では、OPMN が実行されていないホスト・システムで、OracleAS Guard サーバーを停止しています。

```
> asgctl.sh startup
```

---

## startup topology

トポロジ全体の OracleAS コンポーネント・サービスを起動して、停止したトポロジを起動します。

### 書式

startup topology

### パラメータ

なし。

### 使用方法

これは、shutdown topology コマンドを使用して停止した後にトポロジ全体を起動するときに便利なコマンドです。

このコマンドで、OID、OC4J、WebCache などの OracleAS サービスを起動します。startup topology コマンドの実行は、トポロジの各インスタンスすべてに opmnctl startup コマンドを実行することと同じことです。

### 例

次の例では、本番トポロジを起動しています。

```
ASGCTL> startup topology
Generating default policy for this operation

profinfra:7890
  Starting each instance in the topology

prodfinfra:7890 (home /private1/OraHome2/asr1012)
  Executing opmnctl startall command

asmid1:7890 (home /private1/OraHome/asmid1)
  Executing opmnctl startall command

asmid2:7890 (home /private1/OraHome2/asmid2)
  Executing opmnctl startall command
ASGCTL>
```

## stop operation

サーバーで実行中の特定の操作を停止します。

### 書式

```
stop op[eration] <op #>
```

### パラメータ

**op #**

操作の番号。

### 使用方法

サーバーで実行中の操作の番号は、`show operation` コマンドで確認できます。

### 例

次の例では、サーバーで実行中の操作（15）を最初に表示し、`stop operation` コマンドでその操作を停止しています。

```
ASGCTL> show operation
*****
OPERATION: 15
  Status: running
  Elapsed Time: 0 days, 0 hours, 1 minutes, 35 secs
  TASK: instantiateFarm
        TASK: verifyFarm

ASGCTL> stop operation 15
```

## switchover topology

本番サイトのスケジューリングした停止の間に、本番サイトからスタンバイ・サイトへのスイッチオーバー操作を実行します。

### 書式

```
switchover topology to <standby_topology_host>[:<port>] [using policy <file>]
```

### パラメータ

#### standby\_topology\_host

スタンバイ・ホスト・システムの名前。このパラメータは、OracleAS Guard 調整サーバー・インスタンスを指定して、スタンバイ・サイトを構成するインスタンスを検出するために必要です。ホスト・システムは、スタンバイ・トポロジのメンバーである必要があります。

#### port

Oracle ホーム内にある OracleAS Guard サーバーのスタンバイ・ホスト・システムのポート番号。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

### 使用方法

プライマリ Infrastructure システムで、**emagent** プロセスが停止していることを確認します。停止していない場合は、**emagent** プロセスがデータベースと接続されているため、スイッチオーバー操作の実行時に次のエラーが発生することがあります。

```
prodinfra: -->ASG_DGA-13051: Error performing a physical standby switchover.
prodinfra: -->ASG_DGA-13052: The primary database is not in the proper state to perform
a switchover. State is "SESSIONS ACTIVE"
```

UNIX システムの場合、**emagent** プロセスを停止するために、**Application Server Control** を停止します。このコンポーネントは次のように **iasconsole** でコールします。

```
> <ORACLE_HOME>/bin/emctl stop iasconsole
```

UNIX システムの場合、**emagent** プロセスが実行されていないかを確認するために、次の操作を実行します。

```
> ps -ef | grep emagent
```

UNIX システムの場合、**stop iasconsole** の操作の後で **emagent** プロセスが依然として動作しているときに、次のコマンドで前の **ps** コマンドで決定されたプロセス ID (PID) を取得し、停止します。

```
> kill -9 <emagent-pid>
```

Windows システムの場合、「サービス」コントロールパネルを開きます。OracleAS10gASControl サービスを見つけて、このサービスを停止します。

スイッチオーバー操作を実行する前に、OracleAS Infrastructure データベースがプライマリ・トポロジで実行されていることを確認する必要があります。また、**asgctl** の **set primary database** コマンドによって、OracleAS Infrastructure データベースの情報が設定されている必要があります。

グローバル DNS 名を使用してスイッチオーバーを実行することもできます。この場合、OracleAS Disaster Recovery 環境で利用されている高可用性のネーミングと異なることになります。検出メカニズムが、ローカルの名前解決に従って自動的にトポロジを対応するピアにマップします。

OracleAS Guard のスイッチオーバー操作の際に、implicit sync topology 操作が実行され、2つのトポロジが同一であるか確認されます。また、OPMN が新しいスタンバイ Infrastructure ノードで OracleAS Guard サーバーを起動し、このサーバーが永続的に実行されます。次に、このサーバーが新しいスタンバイ・トポロジ内の他のノードの OracleAS Guard サーバーを起動し、起動されたサーバーが一時サーバーになります。

スイッチオーバー・ポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンス（中間層および OracleAS Metadata Repository）でオプションに設定され、Oracle Internet Directory ホームでは必須に設定されています。

スイッチオーバー操作中に、opmn.xml ファイルがプライマリ・サイトからスタンバイ・サイトにコピーされます。このため、TMP 変数の値はプライマリ・サイトとスタンバイ・サイトの両方の opmn.xml ファイルで同じ値に定義する必要があります。そうしないと、ディレクトリが見つからないというメッセージが表示され、このスイッチオーバー操作は失敗します。したがって、TMP 変数が同じ値に定義され、スイッチオーバー操作を試行する前に両方のサイトの同じディレクトリ構造に解決されることを確認してください。

2つの Oracle Identity Management インスタンスが稼動しているプライマリ・サイト (im.machineA.us.oracle.com および im.machineB.us.oracle.com) から、Oracle Identity Management インスタンスが1つだけ稼動している (im.machineA.us.oracle.com) 非対称トポロジのスタンバイ・サイトへのスイッチオーバー操作を実行する場合、つまり、スイッチオーバー・サイトでもう1つのノード (im.machineB.us.oracle.com) を無視する場合、このスイッチオーバー操作を成功させるには、システム管理者が switchover\_policy.xml ポリシー・ファイルを編集してノードを Ignore に設定するだけでなく、このノード (im.machineB.us.oracle.com) で実行中のすべてのプロセスを停止する必要があります。

2つの中間層を持つプライマリ・サイト (Oracle Internet Directory に core1 と core2 インスタンスが登録されている場合など) から中間層 core1 のみを持つ非対称トポロジのスタンバイ・サイトへのスイッチオーバー操作を実行する場合、スタンバイ・サイトの Oracle Internet Directory では、core1 と core2 中間層が両方とも登録されることとなります。switchover\_policy.xml ポリシー・ファイルを編集して、core2 中間層を無視するようにします。core2 中間層はスイッチオーバー操作の間はスタンバイ・サイトに存在しません。しかし、Oracle データベースが格納されている Oracle Internet Directory は本番サイト・トポロジとスタンバイ・サイト・トポロジで同一なので、core2 中間層もスタンバイ・サイト・トポロジの Oracle Internet Directory に登録されていることとなります。そのため、再度対称トポロジにするために、そのスタンバイ・サイト・トポロジに同じ core2 中間層をインストールできません。これは非対称スタンバイ・トポロジを使用するスイッチオーバー操作の厳しい制限です。

スイッチオーバー操作の後に discover topology コマンドが実行される際に、非対称スタンバイ・サイト・トポロジに存在する中間層 (例: instA、instB) が元の本番サイト・トポロジにある中間層 (例: instA、instB、instC) より少ない場合、存在していない中間層のインスタンスごとに警告エラー・メッセージが表示されます (この場合、instC に対して表示される)。このメッセージの表示は、予期されている動作であるため無視できます。スイッチオーバー操作の後で discover topology コマンドが実行されると、OracleAS Server Guard は Oracle Internet Directory の情報を読み取ります。これは、新しいプライマリ・サイト (前のスタンバイ・サイト) にある元のプライマリ・サイトの Oracle Internet Directory の情報と同じです。この Oracle Internet Directory 情報は元のプライマリ・サイトの Oracle Internet Directory 情報と同じであるため、OracleAS Server Guard はこれらの中間層の各インスタンスのホスト / ホームでその存在を検証すると、一部が存在しないことが判明し警告を發します。

詳細は、第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」および第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。

**例**

次の例では、DNS 名が standbyinfra のスタンバイ・サイトにスイッチオーバー操作を実行しています。

```

ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
ASGCTL> switchover topology to standbyinfra
Generating default policy for this operation
prodinfra:7890
    Switchover each instance in the topology to standby topology
prodinfra:7890 (home /private1/OraHome2/asr1012)
    Connecting to the primary database asdb.us.oracle.com
    Gathering information from the primary database asdb.us.oracle.com
    Shutting down each instance in the topology
.
.
.
prodinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
prodinfra:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
ASGCTL>

# Command to use if you are using a policy file
# switchover topology to standbyinfra using policy <file>

```

## sync topology

スタンバイ・サイトとプライマリ・サイトを同期化し、2つのサイト間で一貫性を保ちます。トポロジの同期化操作では、外部の構成ファイルをトポロジ全体に同期化する際に、OracleAS Infrastructure データベースの REDO ログをスタンバイ・サイトに適用します。

### 書式

```
sync topology to <standby_topology_host>[:<port>] [full | incr[emental]] [using policy <file>]
```

### パラメータ

#### standby\_topology\_host

スタンバイ・サイト・ホスト・システムの名前。このパラメータは、OracleAS Guard 調整サーバー・インスタンスを指定して、スタンバイ・サイトを構成するインスタンスを検出するために必要です。ホスト・システムは、スタンバイ・トポロジのメンバーである必要があります。

#### port

Oracle ホーム内にある OracleAS Guard サーバーのスタンバイ・ホスト・システムのポート番号。

#### full | incremental

プライマリ・サイトとスタンバイ・サイトの間で一貫性を保つために同期をとる際には、full または incremental を指定できます。デフォルトは incremental です。全体バックアップが実行されていない場合、デフォルトでは、増分バックアップ操作が実行されません。かわりに、全体バックアップ操作が実行されます。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

### 使用方法

デフォルトでは incremental で同期化が実行され、スタンバイ・サイトとプライマリ・サイトで一貫性が維持され、最も高いパフォーマンスになります。ただし、次の3つの条件に該当する場合には、同期化を full に指定してください。

- スタンバイ・サイトのある時点（現在）のプライマリ・サイトと完全に一致させる場合など、完全な同期化を強制する必要がある場合。
- セカンダリ・サイトと同期するプライマリ・サイトで、短時間に多数のトランザクションの変更が発生することがわかっている場合。
- セカンダリ・サイトと同期するプライマリ・サイトで、長時間にわたって多数のトランザクションの変更が蓄積していることがわかっている場合。

同期化操作により、暗黙の検証操作が実行されます。

同期化のポリシー・ファイルでは、デフォルトで成功要件属性はすべてのインスタンスで必須に設定されています。

詳細は、第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」および第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」を参照してください。

## 例

次の例では、指定したスタンバイ・サイトを OracleAS Guard 調整サーバー（プライマリ・サイト）と同期化しています。デフォルトでは、sync mode に incremental が指定されています。

```
ASGCTL> connect asg prodinfra ias_admin/adminpwd
Successfully connected to prodinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> sync topology to standbyinfra
Generating default policy for this operation
prodinfra:7890
    Synchronizing each instance in the topology to standby topology
prodinfra:7890 (home /private1/OraHome2/asr1012)
    Starting backup of topology ""
        Backing up and copying data to the standby topology
        Backing up each instance in the topology
        Starting backup of instance "asr1012.infra.us.oracle.com"
        Configuring the backup script
asmid1:7890 (home /private1/OraHome/asmid1)
    Starting backup of instance "asmid1.asmid1.us.oracle.com"
asmid2:7891 (home /private1/OraHome/asmid2)
    Starting backup of instance "asmid2.asmid2.us.oracle.com"
.
.
.
asmid2:7890 (home /private1/OraHome2/asr1012)
    Starting backup/synchronization of database "asdb.us.oracle.com"
    Starting restore/synchronization of database "asdb.us.oracle.com"
    Synchronizing topology completed successfully
ASGCTL>

# Command to use if you are using a policy file
# sync topology to standbyinfra using policy <file>
```

## verify topology

プライマリ・トポロジが実行中であること、およびその構成が有効であることを検証します。スタンバイ・トポロジを指定した場合は、ローカル・ホスト・システムがメンバーであるプライマリ・トポロジと、スタンバイ・トポロジを比較し、両方のトポロジが一致していること、および OracleAS Disaster Recovery の要件を順守していることを確認します。

### 書式

```
verify topology [with <host>[:<port>]] [using policy <file>]
```

### パラメータ

#### host

スタンバイ・ホスト・システムの名前。ホスト・システムは、スタンバイ・トポロジのメンバーである必要があります。

#### port

Oracle ホームの OracleAS Guard サーバーのホスト・システムのポート番号。

#### using policy <file>

XML ポリシー・ファイルのフル・パスとファイル名による指定。

### 使用方法

ホスト・システム名を指定しない場合は、ローカル・ホスト・システムが属すトポロジが OracleAS Disaster Recovery のローカル・ルールを順守しているかどうかを確認されます。

スタンバイ・ホスト・システム名を指定した場合は、そのスタンバイ・サイトのトポロジが、本番トポロジとともに、ローカル・ルールと分散配置の OracleAS Disaster Recovery のルールを順守しているかどうかを確認されます。また、プライマリ・サイトとスタンバイ・サイト間の対称性もチェックされます。

検証のポリシー・ファイルでは、デフォルトで成功要件属性はすべての OracleAS ホーム（中間層および OracleAS Metadata Repository）でオプションに設定され、Oracle Internet Directory ホームでは必須に設定されています。

詳細は、[第 7.1 項「OracleAS Guard asgctl コマンドに共通な情報」](#) および [第 7.2 項「OracleAS Guard コマンドの一部に特有の情報」](#) を参照してください。

### 例

次の例では、プライマリ・トポロジが実行中であること、およびその構成が有効であることを検証しています。

```
ASGCTL> connect asg ias_admin/iastest2
Successfully connected to prodinfra:7890
ASGCTL> verify topology
Generating default policy for this operation
prodinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
ASGCTL>
```

次の例では、ローカル・ホスト・システムをメンバーに持つトポロジが、ホスト・システム standbyinfra をメンバーに持つスタンバイ・トポロジと一致していることを検証しています。

```
ASGCTL> connect asg prodfinfra ias_admin/adminpwd
Successfully connected to prodfinfra:7890
ASGCTL> set primary database sys/testpwd@asdb
Checking connection to database asdb
ASGCTL> verify topology with standbyinfra
Generating default policy for this operation
prodfinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
standbyinfra:7890
    HA directory exists for instance asr1012.infra.us.oracle.com
asmid2:7890
    HA directory exists for instance asmid2.asmid2.us.oracle.com
asmid1:7890
    HA directory exists for instance asmid1.asmid1.us.oracle.com
prodfinfra:7890
    Verifying that the topology is symmetrical in both primary and standby configuration
ASGCTL>

# Command to use if you are using a policy file
# verify topology using policy <file>
```

## dump farm (廃止済)

asgctl に、ファームに関する詳細な情報を、指定したファイルに書き込むように指示します。

---

---

**注意：** dump farm コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[dump topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

---

### 書式

dump farm [to <file>]

### パラメータ

**to <file>**

詳細情報の出力の書き込み先である、OracleAS Guard クライアント・ノード内のファイルの名前。

### 使用方法

なし。

### 例

例は「[dump topology](#)」を参照してください。

## instantiate farm (廃止済)

スタンバイ・サイトのファームをインスタンス化します。本番サイトとスタンバイ・サイトの現在のファームの定義を検出し、それらの定義が、これらのシステムに配置されている現在の OracleAS ソフトウェアの OracleAS Disaster Recovery のルールおよび制限を順守していることをインスタンス化の前に検証します。また、スタンバイ・サイトとプライマリ・サイトを同期化し、プライマリ・サイトとスタンバイ・サイトの間で一貫性を保ちます。

---

---

**注意：** `instantiate farm` コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[instantiate topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

---

### 書式

```
instantiate farm to <standby_farm_host>[:<port>]
```

### パラメータ

#### standby\_farm\_host

スタンバイ・ホスト・システムの名前。このパラメータは、OracleAS Guard 調整サーバー・インスタンスを指定して、スタンバイ・サイトを構成するインスタンスを検出するために必要です。ホスト・システムは、スタンバイ・ファームのメンバーである必要があります。

#### port

Oracle ホーム内の OracleAS Guard サーバーのポート番号。

### 使用方法

本番ローカル・システムは、そのサイトの Oracle Notification Server (ONS) ファームに属している必要があります。

スタンバイ・ホストは、スタンバイ・サイトの ONS ファームに属している必要があり、本番ファームのファームと対称性を持っている必要があります。

ファームのインスタンス化操作を実行する前に、OracleAS Infrastructure データベースがプライマリ・ファームで実行されていることを確認する必要があります。また、`asgctl` の `set primary database` コマンドによって、OracleAS Infrastructure データベースの情報が設定されている必要があります。

グローバル DNS 名を使用してインスタンス化を実行することもできます。この場合、OracleAS Disaster Recovery 環境で利用されている高可用性のネーミングと異なることになります。検出メカニズムが、ローカルの名前解決に従って自動的にファームを対応するピアにマップします。

### 例

例は「[instantiate topology](#)」を参照してください。

## shutdown farm (廃止済)

稼働中のファームを停止します。

---

---

**注意：** shutdown farm コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[shutdown topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

---

### 書式

shutdown farm

### パラメータ

なし。

### 使用方法

これは、全ファームを停止するときに便利なコマンドです。ファームを再起動する場合は、[startup farm](#) コマンドを使用します。

### 例

例は「[shutdown topology](#)」を参照してください。

---

## startup farm (廃止済)

停止したファームを起動します。

---

---

**注意：** startup farm コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[startup topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

---

### 書式

startup farm

### パラメータ

なし。

### 使用方法

これは、shutdown farm コマンドを使用して停止した後にファーム全体を起動するときに便利なコマンドです。

### 例

例は「[startup topology](#)」を参照してください。

---

## switchover farm (廃止済)

本番サイトのスケジューリングした停止の間に、本番サイトからスタンバイ・サイトへのスイッチオーバー操作を実行します。

---

**注意：** switchover farm コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[switchover topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

### 書式

```
switchover farm to <standby_farm_host>[:<port>]
```

### パラメータ

#### standby\_farm\_host

ファーム・ホスト・システムの名前。このパラメータは、OracleAS Guard 調整サーバー・インスタンスを指定して、スタンバイ・サイトを構成するインスタンスを検出するために必要です。ホスト・システムは、スタンバイ・ファームのメンバーである必要があります。

#### port

Oracle ホーム内にある OracleAS Guard サーバーのスタンバイ・ホスト・システムのポート番号。

### 使用方法

プライマリ Infrastructure システムで、emagent プロセスが停止していることを確認します。停止していない場合は、emagent プロセスがデータベースと接続されているため、スイッチオーバー操作の実行時に次のエラーが発生することがあります。

```
prodinfra: -->ASG_DGA-13051: Error performing a physical standby switchover.  
prodinfra: -->ASG_DGA-13052: The primary database is not in the proper state to perform  
a switchover. State is "SESSIONS ACTIVE"
```

UNIX システムの場合、emagent プロセスを停止するために、Application Server Control を停止します。このコンポーネントは次のように iasconsole でコールします。

```
> <ORACLE_HOME>/bin/emctl stop iasconsole
```

UNIX システムの場合、emagent プロセスが実行されていないかをチェックするために、次の操作を実行します。

```
> ps -ef | grep emagent
```

UNIX システムの場合、stop iasconsole の操作の後で emagent プロセスが依然として動作しているときに、次のコマンドで前の ps コマンドで決定されたプロセス ID (PID) を取得し、停止します。

```
> kill -9 <emagent-pid>
```

Windows システムの場合、「サービス」コントロールパネルを開きます。OracleAS10gASControl サービスを見つけて、このサービスを停止します。

本番ローカル・システムは、そのサイトの Oracle Notification Server (ONS) ファームに属している必要があります。

スタンバイ・ホストは、スタンバイ・サイトの ONS ファームに属している必要があり、本番ファームのファームと対称性を持っている必要があります。

スイッチオーバー操作を実行する前に、OracleAS Infrastructure データベースがプライマリ・ファームで実行されていることを確認する必要があります。また、`asgctl` の `set primary database` コマンドによって、OracleAS Infrastructure データベースの情報が設定されている必要があります。

グローバル DNS 名を使用してスイッチオーバーを実行することもできます。この場合、OracleAS Disaster Recovery 環境で利用されている高可用性のネーミングと異なることとなります。検出メカニズムが、ローカルの名前解決に従って自動的にファームを対応するピアにマップします。

OracleAS Guard のスイッチオーバー操作の際に、`implicit sync farm` 操作が実行され、2つのファームが同一であるか確認されます。さらに、OPMN は新しいスタンバイ Infrastructure ノードの OracleAS Guard サーバーを自動的に起動し、このサーバーは稼働し続けます。次に、新しいスタンバイ・ファームの他のノードの OracleAS Guard サーバーを起動します。それぞれは一時サーバーになります。

## 例

例は「[switchover topology](#)」を参照してください。

## sync farm (廃止済)

スタンバイ・サイトとプライマリ・サイトを同期化し、2つのサイト間で一貫性を保ちます。トポロジの同期化操作では、外部の構成ファイルをトポロジ全体に同期化する際に、OracleAS Infrastructure データベースの REDO ログをスタンバイ・サイトに適用します。

---

---

**注意：** sync farm コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[sync topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

---

### 書式

```
sync farm to <standby_farm_host>[:<port>] [full | incr[emental]]
```

### パラメータ

#### standby\_farm\_host

スタンバイ・サイト・ホスト・システムの名前。このパラメータは、OracleAS Guard 調整サーバー・インスタンスを指定して、スタンバイ・サイトを構成するインスタンスを検出するために必要です。ホスト・システムは、スタンバイ・ファームのメンバーである必要があります。

#### port

Oracle ホーム内にある OracleAS Guard サーバーのスタンバイ・ホスト・システムのポート番号。

#### full | incremental

プライマリ・サイトとスタンバイ・サイトの間で一貫性を保つために同期をとる際には、full または incremental を指定できます。デフォルトは incremental です。全体バックアップが実行されていない場合、デフォルトでは、増分バックアップ操作が実行されません。かわりに、全体バックアップ操作が実行されます。

### 使用方法

デフォルトでは、sync\_mode が incremental に設定され、最も高いパフォーマンスになります。ただし、次の3つの条件に該当する場合には、sync\_mode を full に指定してください。

- スタンバイ・サイトのある時点（現在）のプライマリ・サイトと完全に一致させる場合など、完全な同期化を強制する必要がある場合。
- セカンダリ・サイトと同期するプライマリ・サイトで、短時間に多数のトランザクションの変更が発生することがわかっている場合。
- セカンダリ・サイトと同期するプライマリ・サイトで、長時間にわたって多数のトランザクションの変更が蓄積していることがわかっている場合。

### 例

例は「[sync topology](#)」を参照してください。

## verify farm (廃止済)

プライマリ・ファームが実行中であること、およびその構成が有効であることを検証します。スタンバイ・ファームを指定した場合は、ローカル・ホスト・システムがメンバーであるプライマリ・ファームと、スタンバイ・ファームを比較し、両方のファームが一致していること、および OracleAS Disaster Recovery の要件を順守していることを確認します。

---

---

**注意：** verify farm コマンドは、OracleAS 10g リリース 2 (10.1.2.0.2) から廃止されています。そのため、[verify topology](#) コマンドを使用します。これは現在および今後の OracleAS リリースの OracleAS Disaster Recovery トポロジの概念をサポートしています。

---

---

### 書式

```
verify farm [with <host>[:<port>]]
```

### パラメータ

#### host

スタンバイ・ホスト・システムの名前。ホスト・システムは、スタンバイ・ファームのメンバーである必要があります。

#### port

Oracle ホーム内の OracleAS Guard サーバーのポート番号。

### 使用方法

ホスト・システム名を指定しない場合は、ローカル・ホスト・システムが属すファームが OracleAS Disaster Recovery のローカル・ルールを順守しているかどうかを確認されます。

スタンバイ・ホスト・システム名を指定した場合は、そのスタンバイ・サイトのファームが、本番ファームとともに、ローカル・ルールと分散配置の OracleAS Disaster Recovery のルールを順守しているかどうかを確認されます。また、プライマリ・サイトとスタンバイ・サイト間の対称性もチェックされます。

### 例

例は「[verify topology](#)」を参照してください。



---

## 手動同期化操作

なんらかの理由でセカンダリ（スタンバイ）サイトがプライマリ・サイトと同期化されていない場合や、[第 8.1.1 項「本番サイトの手動バックアップ」](#)の説明に従って、プライマリ・サイトの中間層および OracleAS Infrastructure の構成ファイルの定期的なバックアップ操作を実行する場合は、次の手動同期化操作を実行する必要があります。次に、[第 8.1.2 項「スタンバイ・サイトの手動リストア」](#)の説明に従って、バックアップの構成ファイルをリストアする必要があります。スタンバイ・サイトで構成ファイル（OracleAS Infrastructure および中間層）をリストアしたら、[6-45 ページの「サイト・フェイルオーバー操作」](#)で説明されている手順 2 に進みます。

## 8.1 OracleAS Guard の asgctl コマンドライン・ユーティリティを使用しない、ベースライン・インストールとスタンバイ・サイトの手動による同期化

---

**注意：** この項、第 8.1.1 項「本番サイトの手動バックアップ」および第 8.1.2 項「スタンバイ・サイトの手動リストア」では、スタンバイ・サイトがプライマリ・サイトと同期化されていない特殊なケースを想定しています。この場合、スタンバイ・サイトでは第 8.1.2 項「スタンバイ・サイトの手動リストア」の説明に従って、前回バックアップした構成ファイルをリストアする必要があります。

asgctl を使用してセカンダリ（スタンバイ）サイトとプライマリ・サイトを継続的に同期化している場合、両サイトはすでに同期化されているはずなので、手動によるリストア操作を実行する必要はありません。計画外停止をリカバリする 6-45 ページの「サイト・フェイルオーバー操作」の手順 1 から始めることができます。

---

本番サイトとスタンバイ・サイト間で Oracle Data Guard を設定した後は、両サイトを同期化する手順を実行できます。本番サイトを使用する前に、インストール後の本番サイトのベースライン・スナップショットをスタンバイ・サイトで取得するために、最初の同期化を実行する必要があります。今後、スタンバイ・サイトで本番サイト構成のリカバリが必要になった場合は、このベースラインを使用できます。

本番サイトから一貫性のあるポイント・イン・タイム・スナップショットを取得するには、OracleAS Infrastructure データベースに格納されている情報と、Oracle Application Server 関連の構成ファイルに格納されている情報（中間層ホストと OracleAS Infrastructure ホスト内にある）を同時に同期化する必要があります。構成ファイルの同期化は、Oracle Application Server Recovery Manager を使用して構成ファイルをバックアップし、それらをスタンバイ・ホストにリストアすることで実行できます。OracleAS Infrastructure データベースの同期化は、Oracle Data Guard を使用してアーカイブ・ログをスタンバイ OracleAS Infrastructure に転送し、それらを構成ファイルのリストアと連携しながら適用することで実行できます。

ベースラインを同期化する手順は次のとおりです（この手順は、今後の同期化の実行でも有効です）。

- OracleAS Infrastructure データベースのアーカイブ・ログの転送
- 構成ファイルのバックアップ（OracleAS Infrastructure および中間層）
- 構成ファイルのリストア（OracleAS Infrastructure および中間層）
- OracleAS Infrastructure データベースのリストア：ログ・ファイルの適用

この後の 2 つの主要な項で、これらの手順について説明します。

### 8.1.1 本番サイトの手動バックアップ

本番サイトとスタンバイ・サイト間で構成情報を同期化するための実行計画とアプローチは、OracleAS Infrastructure および中間層構成ファイルのバックアップとスタンバイ OracleAS Infrastructure データベースへのログ情報の適用を同時に行うことに重点を置いています。

Oracle Application Server では、すべての構成情報が OracleAS Infrastructure データベースにあるわけではありません。データベース・ファイルのバックアップは、中間層および OracleAS Infrastructure 構成ファイルのバックアップと同時性が維持される必要があります。このため、ログ適用サービスはスタンバイ・データベースで有効にできません。ログ・ファイルは本番 OracleAS Infrastructure からスタンバイ OracleAS Infrastructure に転送されますが、適用されません。

本番サイトのバックアップ・プロセスには、中間層および OracleAS Infrastructure ノードの構成ファイルのバックアップが含まれます。さらに、OracleAS Infrastructure データベースのアーカイブ・ログがスタンバイ・サイトに転送されます。

次の項で、バックアップおよびログ転送を実行する手順について説明します。

- [OracleAS Infrastructure データベースのアーカイブ・ログの転送](#)
- [構成ファイルのバックアップ \(OracleAS Infrastructure および中間層\)](#)

---

---

**重要：** これらの項の手順の実行中は、Oracle Application Server システム (構成ファイルおよび OracleAS Infrastructure データベースの基礎になっている) の構成をいっさい変更しないでください。

---

---

---

---

**注意：** 本番サイトで管理上の変更 (中間層および OracleAS Infrastructure ノードでの OracleAS Infrastructure データベースおよび構成ファイルへの変更を含む) が実行される際には、どのような変更であっても必ず、これらの項と「[スタンバイ・サイトの手動リストア](#)」の項で説明するバックアップとリストア手順を実行してください。それに加えて、スケジューリングした定期的なバックアップとリストアも実行する必要があります (たとえば、毎日や週 2 回のペースで)。バックアップとリストア手順の詳細は、『Oracle Application Server 管理者ガイド』を参照してください。

---

---

### 8.1.1.1 OracleAS Infrastructure データベースのアーカイブ・ログの転送

OracleAS Disaster Recovery ソリューションのインストール後は、Oracle Data Guard が本番データベースとスタンバイ・データベースの両方にインストールされているはずですが、本番 OracleAS Infrastructure データベースからスタンバイ OracleAS Infrastructure データベースにアーカイブ・ログを転送する手順には、Oracle Data Guard の構成、および本番データベースとスタンバイ・データベースの両方でいくつかのコマンドを実行することが含まれます。OracleAS Infrastructure データベースのログを転送する手順は次のとおりです。

1. ログ適用サービスが無効になっていない場合は、スタンバイ・ホストで次の SQLPLUS 文を実行して、ログ適用サービスを無効にします。

```
SQL> alter database recover managed standby database cancel;
```

2. 次のコマンドを実行して、本番 OracleAS Infrastructure データベースでログ・スイッチを実行します。これにより、最新のログ・ファイルがスタンバイ OracleAS Infrastructure データベースに転送されます。

```
SQL> alter system switch logfile;
```

3. 本番サイトの通常の運用では、本番データベースのログ・ファイルはスタンバイ・データベースへと頻繁に転送されますが、適用されません。スタンバイ・サイトでは、本番サイトの構成ファイルがバックアップされた時点までの一貫したログを適用する必要があります。次の SQL 文を使用すると、すべての OracleAS Infrastructure データベースの変更を最新のログにカプセル化して、Oracle Data Guard 転送サービスにより、このログをスタンバイ・サイトの OracleAS Infrastructure に転送できます。

```
SQL> select first_change# from v$log where status='CURRENT';
```

SCN つまり順序番号が返されます。この番号は実質的には、転送されたログのタイムスタンプを表します。

4. SCN 番号を書き留めます。この番号は、本番データベースの変更をスタンバイ・サイトにリストアする際に必要となります。

次の項に進み、中間層ホストと OracleAS Infrastructure ホストの構成ファイルをバックアップします。

### 8.1.1.2 構成ファイルのバックアップ (OracleAS Infrastructure および中間層)

この項の手順に従って、構成ファイルをバックアップします。この手順には、OracleAS Recovery Manager を使用する必要があります。OracleAS Recovery Manager は OracleAS の各インストール (中間層および OracleAS Infrastructure) でカスタマイズが必要なため、すでにインストールされ構成済であることを前提にします。OracleAS Recovery Manager の詳細は (インストールと構成手順を含む)、『Oracle Application Server 管理者ガイド』を参照してください。

中間層と OracleAS Infrastructure の各インストールで、次の手順を実行します (中間層および OracleAS Infrastructure の構成ファイルのインストールでも同様です)。

1. 『Oracle Application Server 管理者ガイド』に説明されているインストールおよび構成手順が実行されている場合は、Oracle Application Server Recovery Manager の構成ファイル config.inp で、変数 oracle\_home、log\_path および config\_backup\_path に適切な値が設定されているはずですが、また、OracleAS Recovery Manager の次のコマンドを実行して構成を反映している必要があります。

```
perl bkp_restore.pl -m configure_nodb
```

Windows の場合は、Perl 実行可能ファイルが <ORACLE\_HOME>%perl%<perl\_version>%bin%MSWin32-x86 に用意されています。

これらのタスクが完了していない場合は、この後の手順に進む前に完了してください。

2. 次のコマンドを実行して、現行インストールから構成ファイルをバックアップします。

```
perl bkp_restore.pl -v -m backup_config
```

このコマンドにより、config.inp ファイルの config\_backup\_path 変数で指定されている場所にディレクトリが作成されます。ディレクトリ名には、バックアップの時間が挿入されます。たとえば、config\_bkp\_2003-09-10\_13-21 です。

3. また、バックアップのログが、config.inp ファイルの log\_path 変数で指定されている場所に生成されます。バックアップ・プロセス中にエラーが発生していないかどうか、ログ・ファイルをチェックします。
4. OracleAS Recovery Manager のディレクトリ構造と内容を、現行ノードからスタンバイ・サイトの同等のノードにコピーします。スタンバイ・ノードのパス構造が現行ノードと同じであることを確認します。
5. バックアップ・ディレクトリ (config\_backup\_path で定義されている) を、現行ノードからスタンバイ・サイトの同等のノードにコピーします。スタンバイ・ノードのパス構造が現行ノードと同じであることを確認します。
6. 本番サイト (中間層および OracleAS Infrastructure) における Oracle Application Server の各インストールで上述の手順を繰り返します。

---

**注意:** 本番サイトとスタンバイ・サイト間で一貫性を維持する必要があります。重要な項目が 2 つあります。ディレクトリ名は両サイトで同じである必要があります。また、特定のバックアップ・ディレクトリと SCN との対応関係を、両サイトの管理手順として書き留めておく必要があります。

---

## 8.1.2 スタンバイ・サイトの手動リストア

中間層 Oracle Application Server インスタンスおよび OracleAS Infrastructure からの構成ファイルを OracleAS Infrastructure データベースとともにバックアップした後は、この項の手順に従って、スタンバイ・サイトにファイルとデータベースをリストアします。この項は、次の項目で構成されています。

- 構成ファイルのリストア (OracleAS Infrastructure および中間層)
- OracleAS Infrastructure データベースのリストア: ログ・ファイルの適用

### 8.1.2.1 構成ファイルのリストア（OracleAS Infrastructure および中間層）

本番サイトからのバックアップ・ファイルのリストアには、バックアップに使用した OracleAS Recovery Manager が必要です。この項の手順では、スタンバイ・サイトの中間層ノードおよび OracleAS Infrastructure ノードの各 OracleAS インストールで、OracleAS Recovery Manager がインストールされ構成済であることを前提にします。OracleAS Recovery Manager のインストール手順は、『Oracle Application Server 管理者ガイド』を参照してください。

スタンバイ・サイトの中間層と OracleAS Infrastructure の各インストールで、次の手順を実行します（中間層および OracleAS Infrastructure の構成ファイルのインストールでも同様です）。

1. OracleAS Recovery Manager のディレクトリ構造と本番サイトの同等のインストールから取得したバックアップ・ディレクトリが現行ノードにあることをチェックします。
2. リストア・プロセス中に構成ファイルが変更されないよう Oracle Application Server インスタンスとそのプロセスを停止します。次の OPMN コマンドを使用します。

UNIX の場合：

```
<ORACLE_HOME>/opmn/bin/opmnctl stopall
```

Windows の場合：

```
<ORACLE_HOME>%opmn%bin%opmnctl stopall
```

関連プロセスがいったい実行されていないことをチェックします。UNIX では、次のコマンドを使用します。

```
ps -ef | grep <ORACLE_HOME>
```

Windows の場合は、[Ctrl] キーと [Alt] キーを押しながら [Del] キーを押してタスク マネージャを起動し、プロセスが停止されていることを確認します。

3. バックアップ・ユーティリティの Oracle ホームを構成します。

これは、OracleAS Recovery Manager の Oracle ホームを構成するか、または本番サイトの同等のノードからバックアップ構成ファイル config.inp をコピーすることで実行可能です。次に、OracleAS Recovery Manager の構成オプションを実行する例を示します。

```
perl bkp_restore.pl -v -m configure_nodb
```

Windows の場合は、Perl 実行可能ファイルが <ORACLE\_HOME>%perl%<perl\_version>%bin%MSWin32-x86 に用意されています。

4. 次のコマンドを実行して、有効な構成バックアップ場所を一覧表示します。

```
perl bkp_restore.pl -v -m restore_config
```

5. 次のコマンドを使用して、構成ファイルをリストアします。

```
perl bkp_restore.pl -v -m restore_config -t <backup_directory>
```

<backup\_directory> は、本番サイトからコピーしたバックアップ・ファイルのあるディレクトリの名前です。たとえば、config\_bkp\_2003-09-10\_13-21 です。

6. config.inp で指定されているログ・ファイルで、リストア・プロセス中にエラーが発生していないかどうかをチェックします。
7. 本番サイト（中間層および OracleAS Infrastructure）における Oracle Application Server の各インストールで上述の手順を繰り返します。

### 8.1.2.2 OracleAS Infrastructure データベースのリストア: ログ・ファイルの適用

バックアップ・フェーズでは、本番サイトからスタンバイ・サイトにデータベース・ログ・ファイルを転送するための手順を実行しました。バックアップ用に転送されたのは、書き留めておくよう指示された SCN 番号までのログ・ファイルでした。スタンバイ・データベースをその SCN 番号までリストアするには、次の SQLPLUS 文を使用してスタンバイ OracleAS Infrastructure データベースにログ・ファイルを適用します。

```
SQL> alter database recover automatic from '/private/oracle/oracleas/standby/' standby
database until change <SCN>;
```

(Windows の場合は、この例のパスを適切なものに置き換えます。)

このコマンドを実行し、中間層と OracleAS Infrastructure の各インストールで構成ファイルのリストア手順を完了した後は、スタンバイ・サイトと本番サイトが同期化されています。しかしログ・ファイルの適用時には、不適切なパスの指定によるエラー、およびスタンバイ・サイトに転送されたログ・ファイルの相違という 2 つの障害がよく発生します。

次に、これらの問題を解決する方法を示します。

#### 1. 適切なログ・パスを調べます。

スタンバイ OracleAS Infrastructure データベースで、次の SQLPLUS 文を使用して、受信したアーカイブ・ログの格納場所と数を確認します。

```
SQL> show parameter standby_archive_dest
```

| NAME                 | TYPE   | VALUE                             |
|----------------------|--------|-----------------------------------|
| standby_archive_dest | string | /private/oracle/oracleas/standby/ |

(この例で示したのは UNIX パスです。Windows では、Windows システムでの同等のパスが示されます。)

#### 2. 前の手順で取得したログ・パスを使用して、すべてのログ・ファイルが転送されていることを確認します。

スタンバイ OracleAS Infrastructure データベースで、次を実行します。

```
standby> cd /private/oracle/oracleas/standby
standby> ls
1_13.dbf 1_14.dbf 1_15.dbf 1_16.dbf 1_17.dbf 1_18.dbf 1_19.dbf
```

(Windows では、適切なディレクトリへの移動には cd コマンド、ディレクトリの内容の一覧表示には dir コマンドを使用します。)

本番 OracleAS Infrastructure データベースで、次の SQLPLUS 文を実行します。

```
SQL> show parameter log_archive_dest_1
```

| NAME                | TYPE   | VALUE                                     |
|---------------------|--------|-------------------------------------------|
| log_archive_dest1   | string | LOCATION=/private/oracle/oracleas/oradata |
| log_archive_dest_10 | string |                                           |

(この例で示したのは UNIX パスです。Windows では、Windows システムでの同等のパスが示されます。)

- 手順 1 で指定したパスを使用して、ログ・ファイルの数と順序に注目します。次に、例を示します。

```
production> cd /private/oracle/oracleas/oradata
production> ls
1_10.dbf 1_12.dbf 1_14.dbf 1_16.dbf 1_18.dbf asdb
1_11.dbf 1_13.dbf 1_15.dbf 1_17.dbf 1_19.dbf
```

(Windows では、適切なディレクトリへの移動には cd コマンド、ディレクトリの内容の一覧表示には dir コマンドを使用します。)

この例では、スタンバイ OracleAS Infrastructure に 1\_10.dbf から 1\_12.dbf までのファイルがないことがわかります。ログ・ファイルのこの相違は過去におけるものであるため、ログ転送に使用したネットワークなどの以前の設定に問題があった可能性があります。この問題は明らかに修正されており、以降のログは転送されています。この問題を解決するには、ログ・ファイルをスタンバイ OracleAS Infrastructure データベース・ホストの適切なディレクトリにコピー (FTP) して、この項の最初に示した SQLPLUS リカバリ文を再度実行します。



---

## OracleAS Disaster Recovery サイトの アップグレード手順

この章では、Oracle Application Server Disaster Recovery (OracleAS Disaster Recovery) のサイト全体を OracleAS 10g (9.0.4) から OracleAS 10g リリース 3 (10.1.3.1.0) にアップグレードする方法について説明します。この手順は、サイトを定義しているトポロジ内の Oracle ホーム・タイプすべてに対してリリース 9.0.4 からリリース 10.1.3 にアップグレードすることが可能であることを前提としており、これらの手順を OracleAS Disaster Recovery (DR) ソリューションにおいて拡張します。

このサイトは、OracleAS 10g (9.0.4) 用『Oracle Application Server 高可用性ガイド』の「Oracle Application Server Disaster Recovery」に説明されているように、既存のサポート済 DR 実装をアップグレードします。このプロセスでは、サポートされていない DR 環境はアップグレード済 DR 環境にはアップグレードされません。また、この手順では既存のリリース 9.0.4 Oracle ホーム内でスタンドアロンの OracleAS Guard インストールが利用され、OracleAS Guard の操作手順を使用して指示が表示されます。この環境が可能でない場合は、同等の手動手順を実行できます。つまり、OracleAS Guard の `sync topology` コマンドを使用して、OracleAS 10g (9.0.4) 用『Oracle Application Server 高可用性ガイド』の「Oracle Application Server Disaster Recovery」に説明されているサイトの同期手順を実行して同等の結果を得ることができます。

## 9.1 前提条件

OracleAS 10g (9.0.4) から OracleAS 10g リリース 3 (10.1.3.1.0) への DR サイトのフル・アップグレードを実行するための前提条件は次のとおりです。

- 第 6 章「OracleAS Disaster Recovery」のガイドラインに従って構成されている DR サイトが必要です。
- OracleAS Recovery Manager (旧 OracleAS バックアップおよびリストア・ユーティリティ) が本番サイトとスタンバイ・サイトの両方の Oracle ホームすべてにインストールされている必要があります。使用するバックアップおよびリストア・ユーティリティのバージョンは、そのリリースをサポートしているバージョンになります。たとえば、Oracle 10g (9.0.4) から OracleAS 10g リリース 3 (10.1.3.1.0) への Disaster Recovery サイトのフル・アップグレードを実行する場合、バックアップおよびリストア・ユーティリティのバージョンは OracleAS 10g (9.0.4) である必要があります。
- CD-ROM Disk 2 に収録されている OracleAS Guard の OracleAS 10g リリース 3 (10.1.3.1.0) スタンドアロン・インストールは、すべての OracleAS 10g (9.0.4) Oracle ホームにインストールされます。詳細は、Oracle Application Server のインストール・ガイドの OracleAS Disaster Recovery のインストール情報を参照してください。

## 9.2 Disaster Recovery のトポロジ

この DR 環境に関するシステムは、サイト A とサイト B の 2 つのサイトに含まれています。各サイトの初期のロールは次のとおりです。

- サイト A は本番サイトです。
- サイト B はスタンバイ・サイトです。

これらのサイトは地理的に離れているため、各サイトの現在のロールはこの手順の終了時におけるこれらのサイトの最終的なロールになるものとみなされます。ただし、手順の実行中にこれらのロールは変更されます。したがって、すべての参照はサイト A および B に対して行われます。特定の時点におけるサイトのロールによっては、使用される用語によって混乱を招く場合があります。

## 9.3 高レベル OracleAS Disaster Recovery のアップグレード手順

次の手順では、OracleAS 10g (9.0.4) から OracleAS 10g リリース 3 (10.1.3.1.0) Disaster Recovery へのアップグレード・シナリオについて説明します。これらの手順では、それぞれサイト A とサイト B の Infrastructure システム infra1 および infra2 を参照しています。

1. OracleAS Guard の OracleAS 10g リリース 3 (10.1.3.1.0) スタンドアロン・インストールを本番サイトとスタンバイ・サイトの各 Oracle ホームにインストールします。

1 つのシステムに複数の Oracle ホームが存在する場合は、この構成ファイルで各 OracleAS Guard サーバーにそれぞれ異なるポートが構成されていることを確認してください。デフォルトのポート番号は 7890 です。

```
<ORACLE_HOME>/dsa/dsa.conf
```

2. スタンバイ・サイト [サイト B] で、OracleAS Guard サーバーを起動します。

```
<ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=DSA
```

3. 本番サイト [サイト A] で、OracleAS Guard の Infrastructure システム infra1 に接続し、同期操作を実行します。

この操作によって、トポロジ全体の Oracle ホームが論理的に同期されます。

- a. asgctl クライアントを起動します。

UNIX システムの場合

```
<ORACLE_HOME>/dsa/bin/asgctl.sh
```

Windows システムの場合

```
<ORACLE_HOME>%dsabin%asgctl
```

- b. サイト A の Infrastructure システム `infra1` への接続操作を実行します。

```
ASGCTL> connect asg infra1 ias_admin/<password>
```

- c. プライマリ・データベースをサイト A の OracleAS Metadata Repository に設定します。

```
ASGCTL> set primary database sys/<password>@<site A's servicename>
```

- d. トポロジを検出します。

```
ASGCTL> discover topology oidpassword=<oidpwd>
```

- e. トポロジをダンプします。

```
ASGCTL> dump topology to c:%policy_file_no_904_instances.txt
```

- f. トポロジ・ファイル（この例では `policy_file_no_904_instances.txt`）を編集し、たとえば次のように、すべての 9.0.4 インスタンスを `Ignore` に設定します。

```
<policy>
.
.
.
<instanceList successRequirement="Ignore">
  <instance>904Portal_3</instance>
</instanceList>
.
.
.
</policy>
```

このポリシー・ファイルを Disaster Recovery に関連するすべての操作に使用します。

- g. 手順 3f の編集済ポリシー・ファイルを使用して、スタンバイ・サイト B の Infrastructure システム `infra2` を本番サイトと同期化します。

```
ASGCTL> sync topology to infra2 using policy c:%policy_file_no_904_instances.txt
```

- h. アップグレード手順中に環境が変更されないようにします。これには顧客データの変更は含まれません。顧客データは ID 管理 (IM) および Metadata Repository (MR) データとは別のデータベースに配置されるためです。ただしこのモデルでは、アップグレード手順中に IM および MR データを再び同期することはできません。

4. スタンバイ [サイト B] の Infrastructure に接続し、フェイルオーバーします。

この手順の目的は、DR 環境を 2 つの独立したサイトに分割することです。これによって、サイト B を最初にアップグレードできるようになります。サイト B がアップグレードされたら、アプリケーション・レベルのテストを実行し、アップグレードが完了しており、このサイトが稼動していることを確認できます。この方法を使用する場合、本番サイトであるサイト A は、アップグレード中、完全に DR トレラントにはなりません。理論的には、サイト B がアップグレードされたため、この時点で別のスタンバイ・サイトを確立できます。

OracleAS Guard のフェイルオーバー操作を実行する手順は次のとおりです。

- a. サイト B の Infrastructure システム `infra2` への接続操作を実行します。

```
ASGCTL> connect asg infra2 ias_admin/<password>
```

- b. 新しいプライマリ・データベースをサイト B の OracleAS Metadata Repository に設定します。

```
ASGCTL> set new primary database sys/<password>@<site B's servicename>
```

- c. スタンバイ・サイトであるこのサイト B に対してフェイルオーバー操作を実行します。フェイルオーバー操作によって、`opmnctl startall` コマンドと同等のすべての OPMN 管理サービスがトポロジ全体で開始されます。

```
ASGCTL> failover using policy c:¥policy_file_no_904_instances.txt
```

5. サイト B のサイトに対して他のサービスを開始します。  
アプリケーション、データベース・ジョブ、Enterprise Manager などの、テストに必要な追加のサービスは手動で実行する必要があります。
6. サイト B のシステムに対して、OracleAS アップグレードを実行します（詳細は、『Oracle Application Server アップグレードおよび互換性ガイド』を参照）。
7. 本番サイトでアプリケーションをテストするか、解決が必要な問題を確認します。アップグレードが正常に完了するまで、テストを実行し続けます。
8. 必要に応じてサイトのアクセスをサイト B にリダイレクトします。
- a. 次の操作では、サイト A がアップグレードされます。サイト B はこのアップグレード手順中になんらかのレベルのサービスを提供できます。理論的には、この時点ですべてのアクセスを付与できます。サイト B がアップグレードされると、このサイトでリクエストが処理され、これが DR 環境の本番ロールとなります。サイト A がアップグレードされると、両方のサイトのソフトウェア・バージョンが同じになり、DR のインスタンス化および同期操作が可能になります（手順 12 で実行）。この方法を使用する場合、元の本番サイト [サイト A] で行われた更新は失われます。
- b. 手順 8a が実装され、サイト B が本番サイトになった場合は、サイト A がその後アップグレードされるため、手順 3h の制限は無視してください。
9. サイト A のシステムに対して、OracleAS のアップグレードを実行します（詳細は、『Oracle Application Server アップグレードおよび互換性ガイド』を参照）。
10. アプリケーションをテストするか、解決が必要な問題を確認します。アップグレードが正常に完了するまで、テストを実行し続けます。  
この手順が完了すると、2つのサイトのアップグレードは機能的に同じになり、OracleAS Disaster Recovery 10.1.3.1.0 にアップグレードされた状態になります。サイト B では完全なサイトの機能が有効になり、本番とスタンバイの関係を確立できるようになります。
11. 古い OracleAS 9.0.4 の Oracle ホームで OracleAS Guard サーバーを停止し、UNIX システムの `<ORACLE_HOME>/dsa` ディレクトリまたは Windows システムの `<ORACLE_HOME>¥dsa` ディレクトリにある `dsa.conf` ファイルを削除して、新しい OracleAS 10.1.3 の Oracle ホームのすべてのシステムで DSA プロセスと OPMN サーバーを再起動します。

UNIX システムの場合：

```
<ORACLE_HOME>/opmn/bin/opmnctl stopall
<ORACLE_HOME>/opmn/bin/opmnctl startall
<ORACLE_HOME>/opmn/bin/opmnctl startproc ias-component=DSA
```

Windows システムの場合：

```
<ORACLE_HOME>¥opmn¥bin¥opmnctl stopall
<ORACLE_HOME>¥opmn¥bin¥opmnctl startall
<ORACLE_HOME>¥opmn¥bin¥opmnctl startproc ias-component=DSA
```

12. 手順 8a を利用する場合は、OracleAS Guard を使用して、サイト B からサイト A へのサイトのインスタンス化を実行します。

この手順によって、サイト B とサイト A の OracleAS Disaster Recovery 環境が再確立されます。この手順では、サイト B が本番サイトで、サイト A はサイト B をミラー化するように更新されます。

次の asgctl 手順を実行してこの操作を完了します。

- a. asgctl クライアントを起動します。

UNIX システムの場合  
`<ORACLE_HOME>/dsa/bin/asgctl.sh`

Windows システムの場合  
`<ORACLE_HOME>%dsa%bin%asgctl`

- b. サイト B の Infrastructure システム infra2 への接続操作を実行します。

`ASGCTL> connect asg infra2 ias_admin/<password>`

- c. プライマリ・データベースをサイト B の OracleAS Metadata Repository に設定します。

`ASGCTL> set primary database sys/<password>@<site B's servicename>`

- d. トポロジを検出します。

`ASGCTL> discover topology oidpassword=<oidpwd>`

- e. トポロジをダンプします。

`ASGCTL> dump topology to d:%policy_file_no_904_instances.txt`

- f. トポロジ・ファイル（この例では `policy_file_no_904_instances.txt`）を編集し、たとえば次のように、すべての 9.0.4 インスタンスを `Ignore` に設定します。

```
<policy>
.
.
.
<instanceList successRequirement="Ignore">
  <instance>904Portal_3</instance>
</instanceList>
.
.
.
</policy>
```

このポリシー・ファイルを Disaster Recovery に関連するすべての操作に使用します。

- g. 手順 12f の編集済ポリシー・ファイルを使用して、サイト A のスタンバイ Infrastructure システム infra1 にトポロジをインスタンス化します。

`ASGCTL> instantiate topology to infra1 using policy d:%policy_file_no_904_instances.txt`

13. ドメイン名システム (DNS) のスイッチオーバー操作を実行します。

通常はこの手順をここで実行し、スイッチオーバー操作中の DNS タイムアウトを吸収するようにします。DNS、サイトのサービスおよびアプリケーションがすべてスイッチオーバーされ、実行されるまで、エンド・ユーザーのアクセス・エラー（サービス利用不能）が発生します。

14. OracleAS Guard を使用して、サイト B からサイト A へのスイッチオーバーを実行します。

最終的な目標は、アップグレードの終了時点でこのプロセスの開始時点と同じアクセスを得ることです。したがって、ロールをサイト間で交換する必要があります。サイト B の **Infrastructure** に接続し、プライマリ・データベースを設定し、トポロジの検出を実行し、手順 12f の編集済ポリシー・ファイルを使用してサイト A の **Infrastructure** システム `infra1` へのスイッチオーバーを実行します。

```
ASGCTL> connect asg infra2 ias_admin/<password>
ASGCTL> set primary database sys/<password>@<site B's servicename>
ASGCTL> switchover topology to infra1 using policy d:%policy_file_no_904_
instances.txt
```

15. 手順 9 ~ 14 のかわりに次の手順を実行することもできます。

- a. 本番サイト [サイト A] を停止します。
- b. サイト A のシステムに対して、OracleAS のアップグレードを実行します（詳細は、『Oracle Application Server アップグレードおよび互換性ガイド』を参照）。
- c. OracleAS Guard を使用して、サイト A からサイト B へのインスタンス化を実行します。

16. アプリケーション用に本番サイト A のサービスを開始します。

これで、OracleAS 10g (9.0.4) から OracleAS 10g リリース 3 (10.1.3.1.0) への OracleAS Disaster Recovery サイトのアップグレード手順に必要な手順が完了しました。

## 9.4 既存の OracleAS Disaster Recovery 環境へのパッチ適用

OracleAS Disaster Recovery 環境にパッチを適用して（リリース 10.1.3.1.0 を使用した OracleAS Guard 10.1.2.n.n（n.n は 0.0 および 0.2 を表す）へのパッチ適用）この最新リリースの OracleAS Guard の機能を活用する方法の詳細は、プラットフォーム固有の Oracle Application Server のインストレーション・ガイド、特に「高可用性環境へのインストール: OracleAS Disaster Recovery」を参照してください。この章には、このパッチ適用プロセスについて説明した項「OracleAS Guard リリース 10.1.2.n.n へのリリース 10.1.3.0.0 のパッチの適用」が含まれています。

## DNS サーバーの設定

この章では、UNIX で DNS サーバーを設定する手順について説明します。ここで説明する手順は、[図 6-9 「DNS 解決トポロジの概要」](#) の例のホスト名解決に使用する、サイト固有の DNS ゾーンの設定に適用されます。

---

**注意：** この章で説明する DNS 設定情報の例は、OracleAS Disaster Recovery 運用の理解に役立ちます。これはあくまで DNS の一般的な説明です。DNS のより詳細な情報は、他の適切な DNS ドキュメントを参照してください。

---

この章の説明では、設定した DNS サーバーにより、一意のドメイン名 `oracleas` の新しい DNS ゾーンを作成しサービスを提供します。DNS サーバーはこのゾーン内で、`oracleas` ドメインへのすべてのリクエストを解決し、他のリクエストは企業全体のワイド・エリア DNS サーバーに転送します。

DNS ゾーン・サーバーとして動作する UNIX ホストで、次の手順を実行します。

1. ネーム・サーバー構成ファイル `/var/named.conf` を作成します。企業全体のワイド・エリア DNS サーバーの IP アドレスが `123.1.15.245` である場合、このファイルの内容は次のようになります。

```
options {
    directory "/var/named";
    forwarders {
        123.1.15.245;
    };
};

zone "." in {
    type hint;
    file "named.ca";
};

zone "oracleas" {
    type master;
    file "oracleas.zone";
};

zone "0.0.127.IN-ADDR.ARPA" {
    type master;
    file "127.zone";
};
```

2. 次の内容のルート・ヒント・ファイル `/var/named/named.ca` を作成します（このゾーンの DNS サーバーの IP アドレスは `123.1.2.117` になります）。

```
.          999999  IN      NS      ourroot.private.
ourroot.private.  IN      A       123.1.2.117
```

3. 次の内容のルーブバック・アドレス・ファイル `/var/named/127.zone` を作成します (このゾーンの DNS サーバーのホスト名が `aszone1` であると想定します)。

```
$ORIGIN 0.0.127.IN-ADDR.ARPA.
0.0.127.IN-ADDR.ARPA. IN SOA aszone1.oracleas. root.aszone1.oracleas.
(
    25          ; serial number
    900         ; refresh
    600         ; retry
    86400       ; expire
    3600        ) ; minimum TTL

0.0.127.IN-ADDR.ARPA. IN NS  aszone1.oracleas.
1                    IN PTR localhost.oracleas.
```

4. 次の内容のゾーン・データ・ファイル `/var/named/oracleas.dns` を作成します (ここでの値は図 6-9 の本番サイトの例に基づいています)。

```
;
; Database file oracleas.dns for oracleas zone.
; Zone version: 25
;
$ORIGIN oracleas.
oracleas. IN SOA  aszone1.oracleas. root.aszone1.oracleas (
    25          ; serial number
    900         ; refresh
    600         ; retry
    86400       ; expire
    3600        ) ; minimum TTL

;
; Zone NS records
;
oracleas. IN NS   aszone1.oracleas.

;
; Zone records
;
localhost IN A   127.0.0.1

asmid1    IN A   123.1.2.333
asmid2    IN A   123.1.2.334
infra     IN A   123.1.2.111
remoteinfra IN A 213.2.2.210
```

5. 次のコマンドを実行して、ネーム・サーバーを起動します。

```
/sbin/in.named
```

6. この DNS サーバーのサービスを受けるドメイン内のすべてのホストで、ファイル `/etc/resolv.conf` の `domain` および `nameserver` の設定を次のように編集します (以前の `nameserver` 設定はすべて削除する必要があり、`123.1.2.117` はゾーンの DNS サーバーの IP アドレスを指します)。

```
domain oracleas
nameserver 123.1.2.117
```

# 11

---

---

## Secure Shell (SSH) ポート・フォワーディング

この章では、Oracle Data Guard での Secure Shell (SSH) ポート・フォワーディングの使用方法について説明します。

## 11.1 SSH ポート・フォワーディング

OracleAS Guard は、Oracle Data Guard の使用を自動化します。Oracle Data Guard では、Oracle Net を使用して、REDO データがネットワーク経由でスタンバイ・システムに送信されます。Oracle Data Guard では、本番システムから転送される前の REDO データの暗号化と圧縮およびスタンバイ・システムでの受信時の復号化と圧縮解除を一貫的に実行する手段として、SSH トンネリングを使用することができます。

### 関連項目：

- Data Guard での SSH ポート・フォワーディングの実装：  
<http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=25633.1>
- Data Guard のネットワーク問題のトラブルシューティング：  
<http://metalink.oracle.com/metalink/plsql/showdoc?db=NOT&id=241925.1>

# 第 IV 部

---

## 付録

この部には、このマニュアルのこれまでの章の補足情報が含まれており、次の付録で構成されています。

- [付録 A 「高可用性のトラブルシューティング」](#)
- [付録 B 「OracleAS Guard エラー・メッセージ」](#)



---

---

## 高可用性のトラブルシューティング

この付録では、Oracle Application Server を高可用性構成で配置および管理するときに発生する問題と、それらの解決方法を説明します。この付録の項は次のとおりです。

- 第 A.1 項「OracleAS Disaster Recovery トポロジのトラブルシューティング」
- 第 A.2 項「中間層コンポーネントのトラブルシューティング」
- 第 A.3 項「その他の問題の場合」

## A.1 OracleAS Disaster Recovery トポロジのトラブルシューティング

この項では、OracleAS Disaster Recovery の構成の一般的な障害と解決策について説明します。この項の項目は次のとおりです。

- 第 A.1.1 項「スタンバイ・サイトが同期化されない」
- 第 A.1.2 項「フェイルオーバーまたはスイッチオーバー後にスタンバイ・インスタンスの起動に失敗する」
- 第 A.1.3 項「`dcmctl resyncInstance -force -script` の手順でスイッチオーバー操作に失敗する」
- 第 A.1.4 項「スタンバイ・サイトでスタンドアロンの OracleAS Web Cache インストールを起動できない」
- 第 A.1.5 項「スタンバイ・サイトの中間層インストールで間違ったホスト名が使用されている」
- 第 A.1.6 項「スタンバイ・ファームのファーム検証操作に失敗する」
- 第 A.1.7 項「ファームの同期操作でエラー・メッセージが返される」
- 第 A.1.8 項「Windows システムで PATH 環境変数が 1024 文字を超過している場合に `asgctl startup` コマンドの実行に失敗する可能性がある」

### A.1.1 スタンバイ・サイトが同期化されない

OracleAS Disaster Recovery スタンバイ・サイトの OracleAS Metadata Repository が、プライマリ・サイトの OracleAS Metadata Repository と同期化されない問題について説明します。

#### 障害

OracleAS Disaster Recovery ソリューションでは、手動による構成と、プライマリ・サイトからスタンバイ・サイトへのデータ・ファイルの転送が必要です。また、データ・ファイル（アーカイブ・データベース・ログ・ファイル）は、スタンバイ・サイトで自動的に適用されません。つまり、OracleAS Disaster Recovery は Oracle Data Guard の管理リカバリを使用しません。

#### 解決策

アーカイブ・ログ・ファイルは手動で適用する必要があります。これらの作業の実行に必要な手順については、第 6 章「OracleAS Disaster Recovery」を参照してください。

### A.1.2 フェイルオーバーまたはスイッチオーバー後にスタンバイ・インスタンスの起動に失敗する

スタンバイ・インスタンスが、フェイルオーバーまたはスイッチオーバー操作の後に起動しない問題について説明します。

#### 障害

IP アドレスはインスタンスの構成で使用されます。OracleAS Disaster Recovery の設定では、本番サイトとスタンバイ・サイトのピア・インスタンスに、同一の IP アドレスは必要ありません。OracleAS Disaster Recovery の同期では、本番サイトとスタンバイ・サイト間の IP アドレスの不一致が調整されません。したがって、構成で明示的な IP アドレス xxx.xx.xxx.xx を使用する場合、同期化後のスタンバイの構成は機能しません。

#### 解決策

明示的な IP アドレスの使用を避けます。たとえば、OracleAS Web Cache および Oracle HTTP Server の構成では、リスニング・アドレスとして IP アドレスのかわりに ANY またはホスト名を使用します。

---

## A.1.3 dcmctl resyncInstance -force -script の手順でスイッチオーバー操作に失敗する

OracleAS Disaster Recovery の asgctl switchover 操作では、プライマリ・サイトとスタンバイ・サイトの両方の opmn.xml ファイルで TMP 変数を同じ値に定義する必要があります。

### 障害

OracleAS Disaster Recovery のスイッチオーバーが dcmctl resyncInstance -force -script の手順で失敗し、ディレクトリが見つからないことを示すメッセージが表示されます。

### 解決策

スイッチオーバー操作中に、opmn.xml ファイルがプライマリ・サイトからスタンバイ・サイトにコピーされます。このため、TMP 変数はプライマリ・サイトとスタンバイ・サイトの両方の opmn.xml ファイルで同じ値に定義する必要があります。そうしないと、スイッチオーバー操作に失敗します。asgctl switchover 操作を実行する前に、両方のサイトの opmn.xml ファイルで TMP 変数が同じ値に定義されており、同じディレクトリ構造に解決されることを確認してください。

たとえば、次のコードは、Windows および UNIX 環境での TMP 変数のサンプル定義を示しています。

Example in Windows Environment:

```
-----  
.  
.  
.  
<ias-instance id="infraprod.iasha28.us.oracle.com">  
  <environment>  
    <variable id="TMP" value="C:¥DOCUME~1¥ntregres¥LOCALS~1¥Temp"/>  
  </environment>  
.  
.  
.
```

Example in Unix Environment:

```
-----  
.  
.  
.  
<ias-instance id="infraprod.iasha28.us.oracle.com">  
  <environment>  
    <variable id="TMP" value="/tmp"/>  
  </environment>  
.  
.  
.
```

この問題の回避策は、プライマリ・サイトの opmn.xml ファイルの TMP 変数値を変更し、dcmctl update config 操作を実行し、asgctl switchover 操作を実行することです。この方法によれば、中間層を再インストールしなくても、変更した TMP 変数を利用できるようになります。

## A.1.4 スタンバイ・サイトでスタンドアロンの OracleAS Web Cache インストールを起動できない

スタンバイ・サイトで OracleAS Web Cache を起動できないのは、フェイルオーバーまたはスイッチオーバー後のスタンドアロン OracleAS Web Cache の構成に誤りがあることが考えられます。

### 障害

OracleAS Disaster Recovery の同期では、スタンドアロンの OracleAS Web Cache インストールが同期化されません。

### 解決策

標準の Oracle Application Server の完全な CD イメージを使用して、OracleAS Web Cache コンポーネントをインストールします。

## A.1.5 スタンバイ・サイトの中間層インストールで間違ったホスト名が使用されている

マシンの物理的なホスト名が変更された後も、スタンバイ・サイトの中間層インストールで間違ったホスト名が使用されている問題について説明します。

### 障害

物理的なホスト名の変更に加えて、`/etc/hosts` ファイルにも、物理的なホスト名を最初のエントリとして入力する必要があります。このファイルの変更に失敗すると、インストーラーが間違ったホスト名を使用してしまいます。

### 解決策

`/etc/hosts` ファイルに物理的なホスト名を最初のエントリとして入力します。詳細は、[6-18 ページの第 6.2.2 項「ホスト名解決の構成」](#)を参照してください。

## A.1.6 スタンバイ・ファームのファーム検証操作に失敗する

スタンバイ・ファームのファーム検証操作を実行すると、中間層マシンのインスタンスが見つからず、スタンバイ・ファームが本番のファームと対称性がないことを示すエラー・メッセージが表示され、失敗します。

### 障害

スタンバイ・ファームのファーム検証操作では、本番およびスタンバイ・ファームが互いに対称で、一貫性があり、障害時リカバリの要件に一致していることが検証されます。

検証操作は、中間層インスタンスを `mid_tier.<physical_hostname>` ではなく `mid_tier.<hostname>` とみなすため、失敗します。インストール時に設定した環境変数 `_CLUSTER_NETWORK_NAME_` に問題があると考えられることもできます。ただし、この状況では違います。`_CLUSTER_NETWORK_NAME_` 環境変数設定のチェックで、このエントリが正しいことがわかっています。一方、`/etc/hosts` ファイルのコンテンツのチェックで、問題になっている中間層のエントリが正しくないことがわかります。つまり、すべての中間層インストールは、`/etc/hosts` ファイルの 2 番目の列からホスト名を取得します。

たとえば、次の使用例を想定します。

- `examp1` および `examp2` の 2 つの環境を使用しています。
- 最初に、OracleAS Infrastructure (Oracle Identity Management および OracleAS Metadata Repository) が `examp1` および `examp2` にホスト `infra` としてインストールしています。
- 次に、OracleAS 中間層 (OracleAS Portal および OracleAS Wireless) が `examp1` および `examp2` にホスト `node1` としてインストールしています。
- 要するに、1 つのノード上に 2 つのインストール (OracleAS Infrastructure および OracleAS 中間層) があります。

- 4つの Oracle ホームすべてが最新の duf.jar および backup\_restore ファイルで更新されています。
- 4つの Oracle ホーム(2つのノードにある OracleAS Infrastructure および OracleAS 中間層)のすべてで、OracleAS Guard (asgctl) を起動しています。
- connect asg、set primary、dump farm の asgctl 操作を実行しています。
- standby farm に、asgctl verify farm の操作を実行しました。しかし、インスタンスを mid\_tier.node1.us.oracle.com ではなく mid-tier.examp1 とみなすため失敗します。

/etc/hosts ファイルのチェックにより、次のエントリが示されます。

```
123.45.67.890 examp1 node1.us.oracle.com node1 infra
```

ias.properties およびファームが次のように表示され、検証操作が失敗します。

```
IASname=midtier_inst.examp1
```

/etc/hosts ファイルは、実際は次のようにする必要があります。

```
123.45.67.890 node1.us.oracle.com node1 infra
```

ias.properties およびファームが次のようになり、検証操作が成功します。

```
IASname=midtier_inst.node1.us.oracle.com
```

### 解決策

/etc/hosts ファイルの 2 番目の列のエントリをチェックし、前述で説明した、問題になっている中間層ノードのホスト名と一致するように変更します。

## A.1.7 ファームの同期操作でエラー・メッセージが返される

sync farm to 操作を実行すると、「ASDB に接続できません」というエラー・メッセージが返される問題について説明します。

### 障害

事前に asdb データベース接続を確立する必要がある操作の実行で、ときどき管理者が asgctl コマンドライン・ユーティリティを使用したプライマリ・データベースの設定を忘れることがあります。このような sync farm to 操作の使用例を次に示します。

```
ASGCTL> connect asg hsunnab13 ias_admin/iactest2
```

```
Successfully connected to hsunnab13:7890
```

```
ASGCTL>
```

```
.
```

```
.
```

```
.
```

```
<Other asgctl operations may follow, such as verify farm, dump farm,
<and show operation history, and so forth that do not require the connection
<to the asdb database to be established or a time span may elapse of no activity
<and the administrator may miss performing this vital command.
```

```
.
```

```
.
```

```
.
```

```
ASGCTL> sync farm to usunnaa11
```

```
prodinfra(asr1012): Synchronizing each instance in the farm to standby farm
```

```
prodinfra: -->ASG_ORACLE-300: ORA-01031: insufficient privileges
```

```
prodinfra: -->ASG_DUF-3700: Failed in SQL*Plus executing SQL statement: connect
null/*****@asdb.us.oracle.com as sysdba;.
```

```
prodinfra: -->ASG_DUF-3502: Failed to connect to database asdb.us.oracle.com.
```

```
prodinfra: -->ASG_DUF-3504: Failed to start database asdb.us.oracle.com.
```

```
prodinfra: -->ASG_DUF-3027: Error while executing Synchronizing each instance in the
farm to standby farm at step - init step.
```

**解決策**

asgctl set primary database コマンドを実行します。このコマンドは、sync farm to 操作を実行するために、asdb データベースを開くのに必要な接続パラメータを設定します。現在の接続セッションでプライマリ・データベースが指定されていない場合、set primary database コマンドを、instantiate farm to コマンドおよび switchover farm to コマンドの前に実行する必要があります。

## A.1.8 Windows システムで PATH 環境変数が 1024 文字を超過している場合に asgctl startup コマンドの実行に失敗する場合がある

Windows システムでは、多くの OracleAS インスタンスまたは多くのサード・パーティ・ソフトウェア、あるいはこれら両方がシステムにインストールされているために、システム PATH 環境変数が 1024 文字の制限を超過していると、OracleAS Guard サーバーは OPMN の外部で起動され、システムがディレクトリ・パスを解決できないため、asgctl startup コマンドの実行に失敗する場合があります。

**障害**

多くの OracleAS インスタンスまたは多くのサード・パーティ・ソフトウェア、あるいはこれら両方がインストールされている Windows システムでは、OPMN の外部で実行される asgctl startup コマンドによって、特定のファイルのダイナミック・リンク・ライブラリ orawsec9.dll が見つからないことを示すポップアップ・エラーに続いて、DufException が返されることがあります。次に、例を示します。

```
C:\product\10.1.3\OC4J_1\dsa\bin> asgctl startup
<<Popup Error:>>
The dynamic link library *orawsec9.dll* could not be found.
<<The exception:>>
oracle.duf.DufException
    at oracle.duf.DufOsBase.constructInstance(DufOsBase.java:1331)
    at oracle.duf.DufOsBase.getDufOs(DufOsBase.java:122)
    at
oracle.duf.DufHomeMgr.getCurrentHomePath(DufHomeMgr.java:582)
    at oracle.duf.dufclient.DufClient.main(DufClient.java:132)
stado42: -->ASG_SYSTEM-100: oracle.duf.DufException
-----
```

ただし、この dll は ORACLE\_HOME\bin ディレクトリに存在しません。

このエラーは、OracleAS Guard スタンドアロン・キットでは発生しません。ファイル orawsec9.dll は ORACLE\_HOME\dsa\bin フォルダに存在するためです。

**解決策**

回避策は、必要なパス情報を使用してシステム PATH 変数を手動で編集するか、関連する %PATH% 変数を指定してコマンド・プロンプトの PATH を手動で上書きすることです。次に、例を示します。

```
C:\set PATH=C:\product\10.1.3\OracleAS_OC4J_2\bin;
C:\product\10.1.3\OracleAS_OHS1\jre\1.4.2\bin\client;
C:\product\10.1.3\OracleAS_OHS1\jre\1.4.2\bin;
C:\product\10.1.3\OracleAS_OHS1\bin;C:\product\10.1.3\OC4J_1\bin

C:\product\10.1.3\OC4J_1\dsa\bin> asgctl startup
```

## A.2 中間層コンポーネントのトラブルシューティング

この項では、高可用性構成の中間層コンポーネントの一般的な障害と解決策について説明します。この項の項目は次のとおりです。

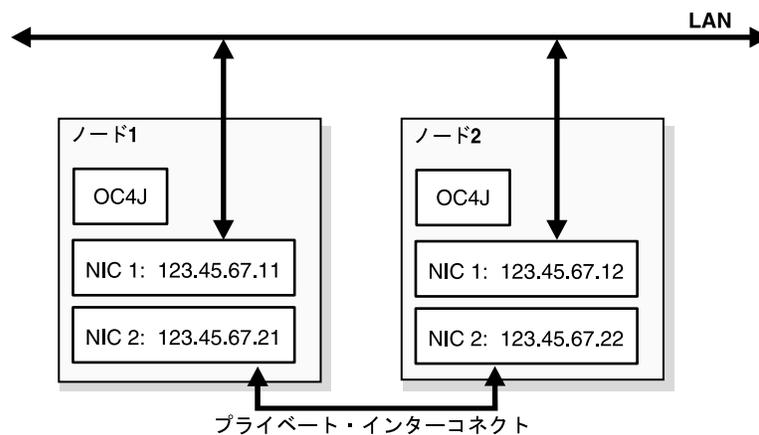
- 第 A.2.1 項「OracleAS Cluster (OC4J-EJB) での複数の NIC の使用」
- 第 A.2.2 項「「opmn:」 URL 接頭辞を使用するとパフォーマンスが遅くなる」

### A.2.1 OracleAS Cluster (OC4J-EJB) での複数の NIC の使用

#### 障害

NIC (ネットワーク・インタフェース・カード) が 2 つ搭載されているコンピュータで OracleAS Cluster (OC4J-EJB) を実行しており、一方の NIC をネットワーク接続に、もう一方の NIC をクラスタ内の他のノードへの接続に使用している場合に、マルチキャスト・メッセージが正常に送受信されないことがあります。これは、セッション情報がクラスタ内のノード間でレプリケートされないことを意味します。

図 A-1 NIC を 2 つ搭載したコンピュータで動作している OracleAS Cluster (OC4J-EJB)



#### 解決策

`oc4j.multicast.bindInterface` パラメータを、ノード上のもう一方の NIC の名前または IP アドレスに設定して、OC4J インスタンスを起動する必要があります。

たとえば、図 A-1 に示す値を使用して、これらのパラメータで OC4J インスタンスを起動します。

ノード 1 では、次のパラメータで起動するように OC4J インスタンスを構成します。

```
-Doc4j.multicast.bindInterface=123.45.67.21
```

ノード 2 では、次のパラメータで起動するように OC4J インスタンスを構成します。

```
-Doc4j.multicast.bindInterface=123.45.67.22
```

このパラメータと値は、Application Server Control コンソールの「サーバー・プロパティ」ページ、「コマンドライン・オプション」セクションの「Java オプション」フィールドで指定します (図 A-2)。

図 A-2 Application Server Control コンソールの「サーバー・プロパティ」ページ

**Ports**

 **TIP** Be sure that the port ranges specified below are large enough to accommodate the the Clusters(OC4J) table.

RMI Ports	12401-12500
JMS Ports	12601-12700
AJP Ports	12501-12600

**RMI-IIOP Ports**

IIOP Ports	
IIOP SSL (Server only)	
IIOP SSL (Server and Client)	

**Command Line Options**

Java Executable	
OC4J Options	
Java Options	-Xrs -server -Djava.security.policy=\${ORACLE_HOME}/j2ee/home/col

## A.2.2 「opmn:」 URL 接頭辞を使用するとパフォーマンスが遅くなる

**障害**

Context.PROVIDER\_URL プロパティで opmn: の接頭辞を使用するアプリケーションがある場合に、InitialContext メソッドのパフォーマンスが遅くなる場合があります。

次のサンプル・コードでは、PROVIDER\_URL が、opmn: 接頭辞を含む URL に設定されます。

```
Hashtable env = new Hashtable();
env.put (Context.PROVIDER_URL, "opmn:ormi://hostname:port/compapp");
// ... set other properties ...
Context context = new InitialContext (env);
```

PROVIDER\_URL で指定されているホストが停止している場合、アプリケーションは OPMN に対してネットワーク接続を行って別のホストを検索する必要があります。ネットワーク経由での OPMN への接続には時間がかかります。

**解決策**

別のホストを見つけるために OPMN に対して別のネットワーク接続を行うことを避けるには、OPMN から最初に返された値をキャッシュし、キャッシュ内の値をアプリケーションで使用できるように、oracle.j2ee.naming.cache.timeout プロパティを設定します。

次のサンプル・コードを実行すると、oracle.j2ee.naming.cache.timeout プロパティが設定されます。

```
Hashtable env = new Hashtable();
env.put (Context.PROVIDER_URL, "opmn:ormi://hostname:port/compapp");

// set the cache value
env.put ("oracle.j2ee.naming.cache.timeout", "30");

// ... set other properties ...

Context context = new InitialContext (env);
```

表 A-1 は、有効な `oracle.j2ee.naming.cache.timeout` プロパティ値を示しています。

**表 A-1 oracle.j2ee.naming.cache.timeout プロパティの値**

値	意味
-1	キャッシュなし。
0	リフレッシュなしで1回だけキャッシュ。
1以上	ここで指定した秒数を経過すると、キャッシュをリフレッシュできます。ただしこれは自動ではありません。リフレッシュは、「 <code>new InitialContext()</code> 」を再度起動した場合にのみ行われます。 このプロパティを設定しない場合のデフォルト値は 60 です。

このプロパティを設定しても、最初の `new InitialContext()` コールではある程度の遅延が見られますが、その後のコールでは、OPMN にネットワーク接続せずにキャッシュからデータを取得するため、速度が増します。

パフォーマンスを最適にするために、さらに `Dedicated.Connection` を YES または `DEFAULT` に、`Dedicated.RMIcontext` を FALSE に設定してください。

## A.3 その他の問題の場合

前述の項で説明した情報では十分でない場合は、[Oracle MetaLink](http://metalink.oracle.com) (<http://metalink.oracle.com>) を参照してください。問題の解決策が見つからない場合は、サービス・リクエストをログに記録してください。

### 関連項目：

- Oracle Application Server のリリース・ノート (Oracle Technology Network の Web サイト：<http://www.oracle.com/technology/documentation/index.html> で入手できます)



---

---

## OracleAS Guard エラー・メッセージ

この付録では、OracleAS Guard エラー・メッセージについて説明します。OracleAS Guard エラー・メッセージには先頭に ASG の接頭辞が付けられますが、ここでは省略しています。エラー・メッセージは、次のグループやサブグループに分類されます。

- DGA エラー・メッセージ
  - LRO エラー・メッセージ
  - Undo エラー・メッセージ
  - テンプレートの作成中のエラー・メッセージ
  - フィジカル・スタンバイのスイッチオーバーのエラー・メッセージ
- Duf エラー・メッセージ
  - データベース・エラー・メッセージ
  - 接続とネットワークのエラー・メッセージ
  - SQL\*Plus エラー・メッセージ
  - JDBC エラー・メッセージ
  - OPMN エラー・メッセージ
  - Net Services エラー・メッセージ
  - システム・エラー・メッセージ
  - 警告エラー・メッセージ
  - OracleAS データベース・エラー・メッセージ
  - OracleAS トポロジ・エラー・メッセージ
  - OracleAS バックアップおよびリストアのエラー・メッセージ
  - OracleAS Guard 同期化エラー・メッセージ
  - OracleAS Guard インスタンス化エラー・メッセージ

## B.1 DGA エラー・メッセージ

DGA エラー・メッセージは、次のとおりです。

---

**注意：** 記号の {0}、{1} および {2} は変数で、オブジェクトの名前に置き換えられます。

---

### 12001、DGA テンプレートの作成中に発生したエラー

**原因：** テンプレート・ファイルの作成中にエラーが発生しました。

**処置：** 2 次エラーを参照してください。

### 12500、スタンバイ・データベース・インスタンス {0} は、すでにホスト {1} に存在しています。

**原因：** 指定されたデータベース・インスタンスは、すでに対象のホストに存在します。

**処置：** 新しいインスタンスを選択するか、現行のインスタンスを削除します。

## B.1.1 LRO エラー・メッセージ

LRO エラー・メッセージは、次のとおりです。

### 13000、フィジカル・スタンバイの作成中に発生したエラー：準備 - 初期化

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13001、フィジカル・スタンバイの作成中に発生したエラー：準備 - スタンバイの確認

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13002、フィジカル・スタンバイの作成中に発生したエラー：準備 - プライマリの処理中

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13003、フィジカル・スタンバイの作成中に発生したエラー：準備 - スタンバイの処理中

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13004、フィジカル・スタンバイの作成中に発生したエラー：準備 - SQL\*Net 構成

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13005、フィジカル・スタンバイの作成中に発生したエラー：コピー - 初期化

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13006、フィジカル・スタンバイの作成中に発生したエラー：コピー - スタンバイの検証

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

### 13007、フィジカル・スタンバイの作成中に発生したエラー：コピー - ファイル・コピー

**原因：** 指定の手順で発生したエラーです。

**処置：** 2 次エラーを参照してください。

**13008、フィジカル・スタンバイの作成中に発生したエラー：終了-初期化**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

**13009、フィジカル・スタンバイの作成中に発生したエラー：終了-プライマリの準備**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

**13010、フィジカル・スタンバイの作成中に発生したエラー：終了-プライマリの構成**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

**13011、フィジカル・スタンバイの作成中に発生したエラー：終了-スタンバイの構成**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

## B.1.2 Undo エラー・メッセージ

Undo エラー・メッセージは、次のとおりです。

**13015、フィジカル・スタンバイ作成の取消し時に発生したエラー：準備**

原因：準備作業の取消し中にエラーが発生しました。

処置：2次エラーを参照してください。

**13016、フィジカル・スタンバイ作成の取消し時に発生したエラー：コピー**

原因：コピー作業の取消し中にエラーが発生しました。

処置：2次エラーを参照してください。

**13017、フィジカル・スタンバイ作成の取消し時に発生したエラー：終了**

原因：終了作業の取消し中にエラーが発生しました。

処置：2次エラーを参照してください。

## B.1.3 テンプレートの作成中のエラー・メッセージ

テンプレートの作成中のエラー・メッセージは、次のとおりです。

**13020、テンプレートの作成中に発生したエラー：初期化**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

**13021、テンプレートの作成中に発生したエラー：プライマリの処理中**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

**13022、テンプレートの作成中に発生したエラー：スタンバイの処理中**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

**13023、テンプレートの作成中に発生したエラー：終了**

原因：指定の手順で発生したエラーです。

処置：2次エラーを参照してください。

## B.1.4 フィジカル・スタンバイのスイッチオーバーのエラー・メッセージ

フィジカル・スタンバイのスイッチオーバーのエラー・メッセージは、次のとおりです。

### 13051、フィジカル・スタンバイのスイッチオーバー実行中に発生したエラー

**原因:** スイッチオーバーの実行中にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

### 13052、プライマリ・データベースは、スイッチオーバーを実行する適切な状態ではありません

**原因:** プライマリ・データベースのスイッチオーバー・ステータスは、TO STANDBY または SESSIONS ACTIVE になっている必要があります。

**処置:** V\$DATABASE 表の SWITCHOVER\_STATUS が、TO STANDBY または SESSIONS ACTIVE になっていることを確認してください。

### 13053、スタンバイ・データベースは、スイッチオーバーを実行する適切な状態ではありません

**原因:** スタンバイ・データベースのスイッチオーバー・ステータスは、TO PRIMARY または SWITCHOVER PENDING になっている必要があります。

**処置:** V\$DATABASE 表の SWITCHOVER\_STATUS が、TO PRIMARY または SWITCHOVER PENDING になっていることを確認してください。

### 13504、データベース・ロールのプライマリからスタンバイへの切り替え中にエラーが発生しました。

**原因:** データベース・ロールのプライマリからスタンバイへの切り替えに失敗しました。

**処置:** 2 次エラーを参照してください。

### 13505、データベース・ロールのスタンバイからプライマリへの切り替え中にエラーが発生しました。

**原因:** データベース・ロールのスタンバイからプライマリへの切り替えに失敗しました。

**処置:** 2 次エラーを参照してください。

### 13061、フィジカル・スタンバイ・データベースをフェイルオーバー中に発生したエラー。

**原因:** スタンバイ・データベースのフェイルオーバーの実行中にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

## B.2 Duf エラー・メッセージ

Duf エラー・メッセージは、次のとおりです。

### 3000、サーバーのエラー {0} です。

**原因:** 無効な引数が指定されています。

**処置:** 有効な引数を指定してください。

### 3001、無効な引数 {0} です。

**原因:** 無効な引数が指定されています。

**処置:** 有効な引数を指定してください。

### 3002、無効なログのパス {0} です。

**原因:** 無効なログ・パスの指定です。

**処置:** 有効なログ・パスを指定してください。

### 3003、無効なコマンドライン値 {0} が指定されました。

**原因:** 無効なコマンドラインの指定です。

**処置:** コマンドライン・オプションを修正して、再度試行してください。

- 3004、無効なコマンド操作 {0} が指定されました。**  
原因：無効なコマンド操作の指定です。  
処置：コマンドライン操作を修正して、再度試行してください。
- 3005、無効なコマンド引数 {0} が指定されました。コマンドはハイフンで開始する必要があります。**  
原因：コマンド引数の先頭にハイフンが使用されていません。  
処置：正しいコマンドライン引数値を入力してください。
- 3006、コマンドライン引数 {0} に必要な値が欠落しています。**  
原因：コマンド引数に必要な値が欠落しています。  
処置：正しいコマンドライン引数値を入力してください。
- 3007、コマンドライン引数 {0} に無効な値 {1} が指定されています。**  
原因：コマンド引数値が正しくありません。  
処置：正しいコマンドライン引数値を入力してください。
- 3008、コマンドライン引数 {0} が必要ですが欠落しています。**  
原因：コマンド引数値が欠落しています。  
処置：正しいコマンドライン引数値を入力してください。
- 3009、無効なセッション ID です。**  
原因：クライアントが無効なセッション ID を渡しました。  
処置：正しいコマンドライン引数値を入力してください。
- 3010、重複しているセッション ID です。**  
原因：このセッション ID はすでに使用されています。  
処置：正しいコマンドライン引数値を入力してください。
- 3011、ライブラリ DufNatives の {0} に不十分なリンクのエラーがあります。**  
原因：ライブラリ DufNatives を使用したコールに失敗しました。  
処置：ライブラリ DufNatives が正しくインストールされていることを確認してください。
- 3012、パスワードのチェックサム・エラーです。**  
原因：ログイン・パスワードにチェックサム・エラーがあります。  
処置：再接続してみてください。
- 3013、操作が失敗しました。**  
原因：指定された操作が失敗しました。  
処置：2 次エラーを参照してください。
- 3014、無効なコマンドラインが指定されました。**  
原因：無効なコマンドラインの指定です。  
処置：コマンドライン・オプションを修正して、再度試行してください。
- 3015、ローカル・ホスト名を取得中に発生したエラー**  
原因：ローカル・ホスト名の取得中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3016、暗号鍵なし**  
原因：暗号鍵がありません。暗号化には暗号鍵が必要です。  
処置：これは、内部プログラミング・エラーです。

**3017、データを暗号化中に発生したエラー**

**原因:** 指定されたデータの暗号化に失敗しました。

**処置:** 2 次エラーを参照してください。

**3018、指定したリクエスト {0} に計画が不足しているエラーです。処理できません**

**原因:** 指定したリクエストの計画が見つかりませんでした。

**処置:** 有効なリクエストか、有効な計画を指定してください。

**3019、サーバーがアプリケーション ID を認識しません**

**原因:** サーバーがサポートしないアプリケーション ID が、クライアントで指定されています。

**処置:** Oracle サポート・サービスに連絡してください。

**3020、ユーザー {0} の認証に失敗しました。正しいユーザー名とパスワードを入力してください。**

**原因:** クライアントで、不適切な OS ユーザー名またはパスワード、あるいはその両方を指定しました。

**処置:** 正しい OS ユーザー名とパスワードを使用していることを確認してください。

**3021、認証にユーザー名またはパスワード (あるいはその両方) が指定されていません。**

**原因:** クライアントで、connect duf コマンドを使用してユーザー名またはパスワード、あるいはその両方を指定しませんでした。

**処置:** ユーザーが、他のコマンドよりも先に connect duf コマンドを発行していることを確認してください。

**3022、ユーザー {0} の認可に失敗しました。ユーザーはサーバー・システムの管理者権限が必要です。**

**原因:** クライアントで指定したユーザー名で DUF サーバーに接続するには、サーバー・システムに対する管理者権限が必要です。このエラーは、Windows システムにのみ適用されます。

**処置:** ユーザー・アカウントが、サーバー・システムの管理者グループに属していることを確認してください。

**3023、エラー: DUF サーバーに接続していません。**

**原因:** 他のコマンドを発行する前に、DUF サーバーに接続する必要があります。

**処置:** DUF サーバーに接続してください。

**3024、ユーザー {0} の認可に失敗しました。Oracle ホームの所有者アカウントを使用する必要があります。**

**原因:** DUF サーバーに接続するためにクライアントで指定するユーザー名は、Oracle ホームのインストール・ユーザーと同じである必要があります。このエラーは、UNIX システムにのみ適用されます。

**処置:** ユーザー・アカウントが、Oracle ホームのアカウントと同じであることを確認してください。

**3025、操作は取り消されました。**

**原因:** 操作が、ユーザーまたは DUF の内部ソフトウェアのいずれかによって取り消されました。

**処置:** なし。

**3026、{1} タスクを実行する前に {0} タスクが正常に完了している必要があります。**

**原因:** 前の必須タスクが正常に完了する前に、指定されたタスクが実行されました。

**処置:** 前の必須タスクを再実行してください。

- 3027、手順 - {1} で {0} を実行中にエラーが発生しました。**  
原因：指定された操作の指定の手順を実行する際に発生するエラーです。  
処置：2 次エラーを参照してください。
- 3028、ホスト {0} で DUF サーバーの起動に失敗しました。**  
原因：指定された操作の指定の手順を実行する際に発生するエラーです。  
処置：詳細は、DUF ログ・ファイルを確認してください。
- 3029、例外により {0} サーバーの起動に失敗しました。**  
原因：サーバーの起動中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3030、エラーです。ホスト名 {0} を解決できません。**  
原因：指定されたホスト名の解決中にエラーが発生しました。  
処置：ホスト名が正しく指定されていることを確認してください。
- 3031、エラーです。ユーザー名 {0} が無効です。ASG サーバーに接続できるのは、{1} アカウントのみです。**  
原因：ASG サーバーに接続できるのは、ias\_admin のみです。  
処置：ias\_admin を使用して、ASG サーバーに接続してください。
- 3032、ホスト {1} で {0} サーバーを起動できませんでした。指定されたホストでサーバーを起動し、再接続してください。**  
原因：接続中に、指定されたホストでサーバーを起動しようとしたためエラーが発生しました。  
処置：サーバーを手動で起動して、再度接続してください。
- 3033、エラーです。サーバーは停止します。**  
原因：サーバーとの通信中にエラーが発生しました。  
処置：再度試行してください。
- 3034、無効なコマンドラインが指定されました：- {0}。**  
原因：無効なコマンドラインの指定です。  
処置：コマンドライン・オプションを修正して、再度試行してください。
- 3035、エラー {1} が発生している ASG サーバー・プロセス {0} の停止に失敗しました。**  
原因：OracleAS Guard クライアントは、OracleAS Guard (ASG) サーバー・プロセスを停止できません。  
処置：コマンドライン・プロンプトから、kill -9 <pid> コマンドを使用してプロセスを停止します。
- 3100、ファイル {0} の読取り中にエラーが発生しました。**  
原因：ファイルからの読取り中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3101、ファイル {0} の書込み中にエラーが発生しました。**  
原因：ファイルの書込み中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3102、ファイル {0} の作成中にエラーが発生しました。**  
原因：指定されたファイルの作成中にエラーが発生しました。  
処置：2 次エラーを参照してください。

- 3103、ファイル {0} の削除中にエラーが発生しました。**  
原因：ファイルの削除中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3104、ファイル {0} のオープン中にエラーが発生しました。**  
原因：ファイルのオープン中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3105、ファイル {0} が見つかりません**  
原因：ファイルのオープン中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3106、ファイル {0} の読取り権限がありません。**  
原因：ファイルのオープン中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3107、ファイル {0} の書込み権限がありません。**  
原因：ファイルのオープン中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3108、ファイル仕様 {0} は完全である必要があります**  
原因：ファイルのオープン中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3109、ファイル {0} のクローズ中にエラーが発生しました**  
原因：ファイルのクローズ中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3110、ディレクトリ {0} の作成中にエラーが発生しました。**  
原因：指定されたディレクトリの作成中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3111、ディレクトリ {0} の削除中にエラーが発生しました。**  
原因：指定されたディレクトリの削除中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3112、ファイルのワイルド・カード仕様 {0} の拡張中にエラーが発生しました。**  
原因：ファイルのワイルド・カード仕様の処理中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3120、構成ファイル {0} のオープン中にエラーが発生しました。**  
原因：構成ファイルのオープン中にエラーが発生しました。  
処置：構成ファイルが存在すること、または指定が正しいことを確認してください。
- 3121、ZIP ファイル {0} の作成中にエラーが発生しました。**  
原因：ZIP ファイルの作成中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 3122、圧縮するファイルがありません。**  
原因：ZIP 圧縮するディレクトリ内にファイルがありません。  
処置：ZIP 圧縮するディレクトリ内にファイルがあることを確認してください。

**3123、ディレクトリ {0} のファイルを ZIP ファイルに追加中にエラーが発生しました。**

原因：指定されたディレクトリ内のファイルを ZIP ファイルに追加する際にエラーが発生しました。

処置：2 次エラーを参照してください。

**3124、ZIP ファイルが指定されていません**

原因：ZIP ファイルが指定されていません。

処置：内部エラーです。

**3125、ZIP ファイル {0} からファイルを解凍中にエラーが発生しました。**

原因：ZIP ファイルからのファイルの解凍中にエラーが発生しました。

処置：2 次エラーを参照してください。

**3400、XML 文書の処理中にエラーが発生しました。**

原因：XML 文書の処理中にエラーが発生しました。

処置：2 次エラーを参照してください。

**3401、XML ノードの処理中にエラーが発生しました。**

原因：XML ノードの処理中にエラーが発生しました。

処置：2 次エラーを参照してください。

**3402、XML リクエスト・メッセージの解析中に発生したエラー**

原因：XML リクエスト・メッセージの解析中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3403、XML レスポンス・メッセージの解析中に発生したエラー**

原因：XML レスポンス・メッセージの解析中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3404、XML ボディ文字列の解析中に発生したエラー**

原因：XML ボディ文字列の解析中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3405、XML DOM へのボディの書き込み中に発生したエラー**

原因：XML ボディ文字列の書き込み中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3406、XML DOM からのボディの読取り中に発生したエラー**

原因：XML ボディ文字列の読取り中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3407、XML DOM への作業項目の書き込み中に発生したエラー**

原因：XML ボディ文字列の書き込み中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3408、XML DOM からの作業項目の読取り中に発生したエラー**

原因：XML ボディ文字列の読取り中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3409、XML 文字列の解析中のエラー**

原因：XML 文字列の解析中にエラーが発生しました。

処置：Oracle サポート・サービスに連絡してください。

**3410、XML DOM の文字列への変換中に発生したエラー**

原因: DOM ツリーから XML 文字列への変換中にエラーが発生しました。

処置: Oracle サポート・サービスに連絡してください。

**3411、XML DOM ツリーの読取り中に発生したエラー**

原因: XML DOM ツリーの読取り中にエラーが発生しました。

処置: Oracle サポート・サービスに連絡してください。

## B.2.1 データベース・エラー・メッセージ

データベース・エラー・メッセージは、次のとおりです。

**3501、DufDb クラスの初期化に失敗しました**

原因: DufDb クラスの作成中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3502、データベース {0} への接続に失敗しました**

原因: データベースへの接続中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3503、データベース {0} の検証に失敗しました**

原因: データベースの検証中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3504、データベース {0} の起動に失敗しました**

原因: データベースの起動中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3505、spfile を含む pfile の作成に失敗しました**

原因: 指定された pfile の作成中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3506、データベースの ARCHIVELOG モードの開始に失敗しました**

原因: ARCHIVELOG モードの開始中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3507、スタンバイ・データベースの制御ファイルの作成に失敗しました**

原因: スタンバイ・データベースの制御ファイルの作成中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3508、pfile の作成に失敗しました**

原因: データベースの初期化パラメータ・ファイルの作成中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3509、spfile の作成に失敗しました**

原因: データベースの spfile の作成中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**3510、{0} の出力リーダー・スレッドが終了しました**

原因: 出力リーダー・スレッドが終了しました。

処置: Oracle サポート・サービスに連絡してください。

- 3511、ノード {0} でのローカル・ワーカの作成中に発生したエラー**  
原因: これは、内部エラーです。  
処置: Oracle サポート・サービスに連絡してください。
- 3512、ノード {0} でのリモート・ワーカの作成中に発生したエラー**  
原因: リモート・サーバーとの通信中に問題が発生しました。  
処置: リモート・サーバーにアクセス可能かどうかを確認してください。
- 3513、データベース {0} は起動していません**  
原因: 指定されたデータベースが起動していません。  
処置: 指定されたデータベースを起動してください。
- 3514、データベース {0} の停止に失敗しました**  
原因: データベースの停止中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3515、現在のアーカイブ・ログ・モードを判断するためのデータベースの問合せに失敗しました**  
原因: 現在のアーカイブ・モードを判断するために、データベースへ問い合わせる際にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3516、データベースの REDO ログ情報の問合せに失敗しました**  
原因: データベースの REDO ログ情報の問合せ中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3517、データベースのスタンバイ REDO ログの削除に失敗しました**  
原因: スタンバイ REDO ログの削除中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3518、スタンバイ・データベースの管理リカバリの起動に失敗しました**  
原因: スタンバイ・データベースの管理リカバリの起動中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3519、スタンバイ・データベースの管理リカバリの取消しに失敗しました**  
原因: スタンバイ・データベースの管理リカバリの取消し中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3520、データベース・インスタンスの有無の判断に失敗しました**  
原因: 指定されたデータベース・インスタンスの有無の判断中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3521、無効なデータベース・インスタンス {0} がテンプレート・ファイルで指定されました ; DUF がインスタンス {1} を検出しました**  
原因: テンプレート・ファイルに指定されているスタンバイ・データベース・インスタンスは、DUF がシステムで検出したものと異なります。  
処置: 新しいスタンバイ・インスタンスで準備とコピーの各フェーズを再実行するか、システムで検出された正しいスタンバイ・データベース・インスタンスを指定してください。
- 3522、spfile の生成に必要な pfile{0} が欠落しています**  
原因: スタンバイ・データベースで使用される spfile の作成に必要な pfile がありません。  
処置: 準備とコピーの各フェーズを再実行して pfile を生成するか、正しい値の pfile を手動で作成してください。

- 3523、スタンバイ・データベースはプライマリ・データベースと同一のサービス名を持つことはできません  
原因: スタンバイ・サービス名は、プライマリと同じです。  
処置: スタンバイ・サービス名を変更してください。
- 3524、エラー: プライマリ・データベースが設定されていません  
原因: プライマリ・データベースが定義されていません。  
処置: 最初にプライマリ・データベースを設定してください。
- 3525、エラー: スタンバイ・データベースが設定されていません  
原因: スタンバイ・データベースが定義されていません。  
処置: 最初にスタンバイ・データベースを設定してください。
- 3526、スタンバイ・データベースを設定する前にプライマリ・データベースを設定してください  
原因: スタンバイ・サービス名は、同一ホスト上のプライマリと同じです。  
処置: スタンバイ・サービス名を変更してください。
- 3527、データベースの表領域マップが NULL です  
原因: これは、内部エラーです。  
処置: Oracle サポート・サービスに連絡してください。
- 3528、初期化パラメータ・ファイル {0} を初期化中に発生したエラー  
原因: パラメータ・ファイルの初期化中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3529、初期化パラメータ・ファイル {0} を書き込み中に発生したエラー  
原因: パラメータ・ファイルの書き込み中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3530、データベース {0} の保護モードを設定中に発生したエラー  
原因: 保護モードの設定中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3531、データベース {0} を読取り専用モードでオープン中に発生したエラー  
原因: データベースを読取り専用モードでオープン中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3532、{0} からの初期化パラメータ値の取得に失敗しました。  
原因: 初期化パラメータ・ファイルからのパラメータ値の取得中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 3533、データベース {0} にユーザー名またはパスワード (あるいはその両方) が指定されていません。  
原因: 初期化パラメータ・ファイルからのパラメータ値の取得中にエラーが発生しました。  
処置: ユーザーは、set primary database または set standby database コマンドを使用してデータベースに接続するために、ユーザー名とパスワードを指定する必要があります。
- 3534、スタンバイ・データベースはプライマリ・データベースと同一のホストを持つことはできません  
原因: スタンバイ・ホストは、プライマリと同じです。  
処置: スタンバイまたはプライマリのデータベース・ホスト名を変更してください。

**3535、スタンバイ REDO ログの作成に失敗しました。**

**原因:** スタンバイ REDO ログの作成中にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

**3536、ログ・アーカイブの宛先からスタンバイ・データベースのリストの取得に失敗しました。**

**原因:** ログ・アーカイブの保存先パラメータから、スタンバイ・データベースのリストを取得している際にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

**3537、ログ・アーカイブの宛先としてスタンバイ・データベースの追加に失敗しました。**

**原因:** ログ・アーカイブの保存先であるスタンバイ・データベースを追加する際にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

**3538、ログ・アーカイブの宛先としてスタンバイ・データベースの削除に失敗しました。**

**原因:** ログ・アーカイブの保存先であるスタンバイ・データベースを削除する際にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

**3539、エラー: 新しいプライマリ・データベースが設定されていません。**

**原因:** 新しいプライマリ・データベースが定義されていません。

**処置:** 最初に新しいプライマリ・データベースを設定してください。

**3540、テンプレート・ファイル {0} の処理中にエラーが発生しました。**

**原因:** テンプレート・ファイルの処理中にエラーが発生しました。

**処置:** 保護を修正して、再度試行してください。

**3541、無効なデータベース保護がテンプレート・ファイル {0} で指定されました。**

**原因:** テンプレート・ファイルの保護値の処理中にエラーが発生しました。

**処置:** 保護を修正して、再度試行してください。

**3542、データベース・ロールの問合せに失敗しました。**

**原因:** データベース・ロールの問合せ中にエラーが発生しました。

**処置:** 2 次エラーを参照してください。

**3543、コマンドの処理中のエラーです。プライマリ・トポロジの OracleAS Guard サーバーに接続する必要があります。**

**原因:** ユーザーが、プライマリ・トポロジではないトポロジ上のサーバーに接続していません。

**処置:** プライマリ・トポロジのノードに接続します。

**3544、コマンドの処理中のエラーです。スタンバイ・トポロジの OracleAS Guard サーバーに接続する必要があります。**

**原因:** ユーザーが、スタンバイ・トポロジではないトポロジ上のサーバーに接続していません。

**処置:** プライマリ・トポロジのノードに接続します。

**3545、新規データベースの作成時、旧パスワード・ファイル %1 の削除中にエラーが発生しました。**

**原因:** データベースの削除操作の際に旧パスワード・ファイルを削除できませんでした。これは、新規データベースを作成するときに問題になります。

**処置:** 古いパスワード・ファイルを削除します。

- 3546、エラーです。データベース SID には値が必要でしたが空です。**  
原因：データベース SID には値が必要でしたが空でした。  
処置：これは、内部エラーです。
- 3547、サーバーのクリップボードにデータベース資格証明を格納中にエラーが発生しました。**  
原因：指定されたサーバーのクリップボード上へのデータベース資格証明の格納に失敗しました。  
処置：内部エラーです。
- 3548、サーバーのクリップボードにデータベース情報を格納中にエラーが発生しました。**  
原因：指定されたサーバーのクリップボード上へのデータベース情報の格納に失敗しました。  
処置：内部エラーです。
- 3549、スタンバイ・ホストでデータベースをクリーンアップ中にエラーが発生しました。**  
原因：スタンバイ・ホスト上のデータベースのクリーンアップに失敗しました。  
処置：2 次エラーを参照してください。
- 3550、有効な Oracle ホームの検出に失敗しました**  
原因：この操作で、有効な Oracle ホームが検出されませんでした。  
処置：有効な Oracle ホームを作成してください。
- 3551、Oracle Data Guard ホームにはデータベース・サーバー・ホームと同じ所有者が必要です**  
原因：Oracle Data Guard ホームが、データベース・サーバー・ホームと異なるユーザーに所有されています。  
処置：Oracle データベース・サーバーの所有者として、Oracle Data Guard ユーザーを再インストールしてください。
- 3552、指定された Oracle ホーム {0} が見つかりませんでした**  
原因：指定された Oracle ホームが見つかりませんでした。  
処置：有効な Oracle ホームを指定してください。
- 3553、システム上の Oracle ホームのリストを取得中にエラーが発生しました。**  
原因：Oracle ホームのリストを読み取ることができませんでした。  
処置：Oracle インベントリが有効であることを確認してください。
- 3554、SID {0} を含む Oracle ホームが見つかりません。**  
原因：特定の SID を含む Oracle ホームが見つかりません。  
処置：Oracle ホーム・インベントリが有効であることを確認してください。
- 3555、ホーム・インベントリにアクセス中にエラーが発生しました。インベントリ・ファイルが存在することを確認してください。**  
原因：Oracle ホーム・インベントリにアクセスできません。  
処置：Oracle ホーム・インベントリが存在することを確認してください。
- 3556、エラー：パス {0} 内のホームを検出できません**  
原因：指定されたパスに Oracle ホームが見つかりません。  
処置：Oracle ホーム・インベントリが存在することを確認してください。

## B.2.2 接続とネットワークのエラー・メッセージ

接続とネットワークのエラー・メッセージは、次のとおりです。

### 3600、サーバーへの接続中に発生したエラー: 不明なノード {0}

原因: クライアントがサーバー・ホストを認識していません。

処置: Oracle サポート・サービスに連絡してください。

### 3601、サーバー・ノード {0} への接続中に発生したエラー

原因: クライアントがサーバーに接続できません。

処置: Oracle サポート・サービスに連絡してください。

### 3602、ファイル・コピーのプロトコル・エラー

原因: ファイルのコピー中に内部プロトコル・エラーが発生しました。

処置: Oracle サポート・サービスに連絡してください。

### 3603、ネットワークを介してデータを送信中に発生したエラー

原因: ネットワーク・エラーが発生しました。

処置: 再度試行してください。

### 3604、ネットワークを介してデータを受信中に発生したエラー

原因: ネットワーク・エラーが発生しました。

処置: 再度試行してください。

### 3605、ファイル・コピー操作が終了しました

原因: エラーが発生したため、コピー操作が強制終了されました。

処置: 再度試行してください。

### 3606、ファイル・コピー・サーバー {0}(ポート {0} 上) への接続中に発生したエラー

原因: コピー・サーバーは稼動していません。

処置: Oracle サポート・サービスに連絡してください。

### 3607、{0}(ポート {0}) でファイル・コピーのサーバー・ソケットをオープン中に発生したエラー

原因: エラーが発生したため、コピー操作が強制終了されました。

処置: 再度試行してください。

### 3608、クリップボードへの接続中に発生したエラー

原因: クリップボード・サーバーに接続できません。

処置: 再度試行してください。

### 3609、{0} から {1} にコピー中にエラーが発生しました。

原因: ファイルのコピー中にエラーが発生しました。

処置: 2次エラーを参照してください。

### 3610、オンライン・バックアップを開始中に発生したエラー

原因: 表領域をオンライン・バックアップ・モードにする際にエラーが発生しました。

処置: 2次エラーを参照してください。

### 3611、オンライン・バックアップを終了中に発生したエラー

原因: 表領域をオンライン・バックアップ・モードからリストアする際にエラーが発生しました。

処置: 2次エラーを参照してください。

**3612、サーバー・ポート {0} でリスニング中に発生したエラー**

原因: ポートをリスニング中にエラーが発生しました。

処置: サーバーがすでに稼動しているかどうかを確認してください。

**3613、ネットワーク・バッファのオーバーフローが検出されました。**

原因: ネットワーク・プロトコルが、バグまたは攻撃によるバッファ・オーバーフローを検出しました。

処置: Oracle サポート・サービスに連絡してください。

## B.2.3 SQL\*Plus エラー・メッセージ

SQL\*Plus エラー・メッセージは、次のとおりです。

**3700、SQL\*Plus で SQL 文の実行に失敗しました: {0}**

原因: 指定された SQL 文の実行に失敗しました。

処置: 2 次エラーを参照してください。

**3701、SQL\*Plus の起動に失敗しました: {0}**

原因: 指定された SQL 文の実行に失敗しました。

処置: 2 次エラーを参照してください。

## B.2.4 JDBC エラー・メッセージ

JDBC エラー・メッセージは、次のとおりです。

**3751、Oracle JDBC ドライバ: oracle.jdbc.OracleDriver の登録に失敗しました**

原因: Oracle JDBC ドライバの登録に失敗しました。

処置: Oracle JDBC ドライバがローカル・システムにインストールされていることを確認してください。

**3752、データベースへの JDBC 接続が存在しません**

原因: データベース・サーバーに接続できません。

処置: 最初にデータベース・サーバーに接続してから、再度試行してください。

**3753、データベースへの接続に失敗しました。**

原因: データベース・サーバーに接続できません。

処置: 2 次エラーを参照してください。

**3754、データベースからの切断に失敗しました**

原因: データベース・サーバーを切断できません。

処置: 2 次エラーを参照してください。

**3755、SQL 文の実行に失敗しました**

原因: SQL 文の実行に失敗しました。

処置: 2 次エラーを参照してください。

**3756、SQL 問合せの実行に失敗しました**

原因: SQL 問合せ文の実行に失敗しました。

処置: 2 次エラーを参照してください。

**3757、Oracle 結果セットまたは文オブジェクトのクローズに失敗しました**

原因: Oracle 結果セットまたは文オブジェクトのクローズに失敗しました。

処置: 2 次エラーを参照してください。

- 3758、この方法は、**フィジカル・スタンバイ・データベースの検証に使用することはできません**  
原因: これは、プログラミング・エラーです。  
処置: Oracle サポート・サービスに連絡してください。
- 3759、**データベースの問合せでデータが戻されないことを確認**  
原因: データベースの問合せでデータが戻されないことを確認します。  
処置: 2 次エラーを参照してください。
- 3760、**アーカイブ・ログの宛先情報の問合せに失敗しました**  
原因: アーカイブ・ログの宛先情報の問合せに失敗しました。  
処置: 2 次エラーを参照してください。
- 3761、**REDO ログ情報の問合せに失敗しました**  
原因: REDO ログ情報の問合せに失敗しました。  
処置: 2 次エラーを参照してください。
- 3762、**SQL 文の結果の処理に失敗しました**  
原因: SQL 文の結果の処理に失敗しました。  
処置: 2 次エラーを参照してください。
- 3763、**データベースのデータファイルの問合せに失敗しました**  
原因: データベースのデータファイルの問合せに失敗しました。  
処置: 2 次エラーを参照してください。
- 3764、**データベースで使用されるログ・ファイルの問合せに失敗しました**  
原因: データベースで使用されるログ・ファイルの問合せに失敗しました。  
処置: 2 次エラーを参照してください。
- 3765、**表領域情報の問合せに失敗しました**  
原因: データベースの表領域情報の問合せに失敗しました。  
処置: 2 次エラーを参照してください。

## B.2.5 OPMN エラー・メッセージ

OPMN エラー・メッセージは、次のとおりです。

- 3800、**OPMN Manager への接続の試行に失敗**  
原因: OPMN Manager への接続の試行中にエラーが発生しました。  
処置: OPMN Manager が起動されていることを確認してください。
- 3801、**{0} 上の OPMN Manager からトポロジ情報を取得しようとして失敗しました。**  
原因: OPMN Manager からのトポロジ情報の取得中にエラーが発生しました。  
処置: OPMN Manager が起動され、正しく機能していることを確認してください。
- 3802、**OPMN コンポーネント "{0}" の停止に失敗しました。**  
原因: 指定された OPMN コンポーネントの停止に失敗しました。  
処置: 2 次エラーを参照してください。
- 3803、**OPMN コンポーネント "{0}" の起動に失敗しました。**  
原因: 指定された OPMN コンポーネントの起動に失敗しました。  
処置: 2 次エラーを参照してください。

- 3804、"{0}" からトポロジ・エントリを削除しようとして失敗しました。  
原因: opmn.xml からのトポロジ・エントリの削除に失敗しました。  
処置: 2 次エラーを参照してください。
- 3900、サービスにはすでに削除のマークが付けられていたため、Oracle データベース・サービスの作成中にエラーが発生しました。ノード「{0}」で Windows Service Control Manager を終了してください。再試行しますか。  
原因: ユーザーが SCM を開いているために、サービスの操作が失敗しました。  
処置: SCM GUI を終了する必要があります。

## B.2.6 Net Services エラー・メッセージ

Net Services エラー・メッセージは、次のとおりです。

- 4000、{0} のデフォルトのネット・サービス・ドメインの取得の試行に失敗しました  
原因: デフォルトのネット・サービス・ドメインの取得の試行に失敗しました。  
処置: 2 次エラーを参照してください。
- 4001、{0} のネット・サービス名エントリの追加を試行中に発生したエラー  
原因: 指定されたサービス名の追加に失敗しました。  
処置: 2 次エラーを参照してください。
- 4002、{0} のネット・サービス名エントリの取得を試行中に発生したエラー  
原因: 指定されたサービス名の取得に失敗しました。  
処置: 2 次エラーを参照してください。
- 4003、{0} のネット・サービス・エントリからホスト名の取得を試行中に発生したエラー  
原因: ネット・サービス・エントリからのホスト名の取得に失敗しました。  
処置: 2 次エラーを参照してください。
- 4004、ネット・サービスの説明からのホスト名の取得を試行中に発生したエラー  
原因: ネット・サービスの説明からのホスト名の取得に失敗しました。  
処置: 2 次エラーを参照してください。
- 4005、ネット・サービス・リスナー情報の取得を試行中にエラーが発生しました。  
原因: ネット・サービス・リスナー情報の取得に失敗しました。  
処置: 2 次エラーを参照してください。
- 4006、デフォルトのネット・サービス・リスナーの作成を試行中に発生したエラー  
原因: デフォルト・リスナーの作成に失敗しました。  
処置: 2 次エラーを参照してください。
- 4007、ネット・サービス・リスナー {1} への SID エントリ {0} の追加を試行中にエラーが発生しました。  
原因: リスナーへの SID エントリの追加に失敗しました。  
処置: 2 次エラーを参照してください。
- 4008、ネット・サービス・リスナー・コマンド用のコマンド・スクリプトを生成中にエラーが発生しました:{0}。  
原因: リスナーのコマンド・スクリプトの生成に失敗しました。  
処置: 2 次エラーを参照してください。

- 4009、ネット・サービス・リスナー・コマンド用のコマンド・スクリプトを実行中にエラーが発生しました:{0}。  
原因: リスナーのコマンド・スクリプトの実行に失敗しました。  
処置: 2 次エラーを参照してください。
- 4010、{0} のネット・サービス TNS エントリを追加中にエラーが発生しました。  
原因: TNS エントリの追加に失敗しました。  
処置: 2 次エラーを参照してください。
- 4011、ネット・サービス・リスナー {1} に対する SID エントリ {0} の削除を試行中にエラーが発生しました。  
原因: リスナーからの SID エントリの削除に失敗しました。  
処置: 2 次エラーを参照してください。
- 4012、リスナー構成の保存を試行中にエラーが発生しました。  
原因: リスナー情報が変更され、情報の保存に失敗しました。  
処置: 2 次エラーを参照してください。
- 4013、{0} のネット・サービス TNS エントリを削除中にエラーが発生しました。  
原因: TNS エントリの削除に失敗しました。  
処置: 2 次エラーを参照してください。
- 4014、lsnrctl コマンドを使用して TNS リスナーを開始中にエラーが発生しました。  
原因: TNS リスナーの開始に失敗しました。  
処置: 2 次エラーを参照してください。
- 4030、コマンド {0} はタイムアウトのため失敗しました  
原因: コマンドのタイムアウトが発生しました。  
処置: 構成ファイルのタイムアウト値を増やしてください。
- 4031、enve コマンドを使用した環境変数の取得でエラーが発生しました  
原因: env コマンドが機能しません。  
処置: /bin ディレクトリまたは /usr/bin ディレクトリに env 実行可能ファイルが含まれていることを確認してください。
- 4040、外部プログラムまたはスクリプトの実行中にエラーが発生しました。  
原因: 指定されたコマンドの実行に失敗しました。  
処置: 2 次エラーを参照してください。
- 4041、TNS 名記述子 {1} からの {0} の値の取得に失敗しました。  
原因: TNS 名記述子からの指定されたパラメータ値の取得に失敗しました。  
処置: 2 次エラーを参照してください。
- 4042、TNS 名記述子 {1} 用の {0} の値の更新に失敗しました。  
原因: TNS 名記述子の指定されたパラメータ値の更新に失敗しました。  
処置: 2 次エラーを参照してください。
- 4043、TNS 記述子エントリ {0} とエントリ {1} の比較に失敗しました。  
原因: 2 つの TNS エントリの比較に失敗しました。  
処置: 2 次エラーを参照してください。

**4044、サービス名用のリモート TNS 名記述子の生成に失敗しました**

原因: 指定されたローカル・データベースのリモート TNS 名記述子の生成に失敗しました。

処置: 2 次エラーを参照してください。

**4045、サービス名用のリモート TNS サービス名の取得に失敗しました**

原因: 指定されたローカル・データベースのリモート TNS サービス名の取得に失敗しました。

処置: 2 次エラーを参照してください。

## B.2.7 LDAP または OID のエラー・メッセージ

LDAP または OID のエラー・メッセージは、次のとおりです。

**4101、ホスト {0}、ポート {1} の OID サーバーへの接続に失敗しました。**

原因: このホストおよびポートから OID サーバーに接続できませんでした。

処置: 2 次エラーを参照してください。

**4102、ホスト {0}、ポート {1} で、SSL を介した OID サーバーへの接続に失敗しました。**

原因: このホストおよびポートから、SSL を介して OID サーバーに接続できませんでした。

処置: 2 次エラーを参照してください。

**4103、ユーザーは OID サーバーのホスト、ポート、ユーザー名およびパスワードを指定する必要があります。**

原因: ユーザーがこれらのパラメータを一部指定していません。

処置: OID サーバーにアクセスするには、これらのパラメータをすべて指定する必要があります。

**4104、OID サーバーからの属性 {0} の値の取得に失敗しました。**

原因: この属性の値の取得に失敗しました。

処置: 2 次エラーを参照してください。

**4105、OID サーバーからの DN{0} の属性の取得に失敗しました。**

原因: この DN の属性の取得に失敗しました。

処置: 2 次エラーを参照してください。

**4106、OID サーバーからの Oracle アプリケーション・サーバー・インスタンスの取得に失敗しました**

原因: OID サーバーから Oracle アプリケーション・サーバー・インスタンスを取得できませんでした。

処置: 2 次エラーを参照してください。

**4107、OID サーバーからのインフラストラクチャ・データベースの取得に失敗しました。**

原因: OID サーバーからインフラストラクチャ・データベースを取得できませんでした。

処置: 2 次エラーを参照してください。

**4110、ファイル {0} は存在しないため、そのファイルへ現在のトポロジを設定できません。**

原因: トポロジ・ファイルが存在しません。

処置: 存在するファイルのファイル名を指定します。

**4111、現在のトポロジ・ファイル {0} は存在しません。set topology コマンドを使用して、有効なトポロジ・ファイルを指定します**

原因: トポロジ・ファイルが存在しません。

処置: dsa.conf に存在するファイルのファイル名を指定します。

## B.2.8 システム・エラー・メッセージ

システム・エラー・メッセージは、次のとおりです。

### 4900、サーバーで例外が発生しました

原因：サーバーの例外が発生しました。

処置：2 次エラーを参照してください。

### 4901、サーバーで NULL ポインタ例外が発生しました

原因：ソフトウェア・エラーです。

処置：2 次エラーを参照してください。

### 4902、キー {0} ではクリップボードにオブジェクトが見つかりません

原因：ソフトウェア・エラーです。

処置：2 次エラーを参照してください。

### 4903、{0} の成功最小値は、グループ {1} のワーカーに適合しませんでした

原因：同じグループに所属するワーカーの成功数が指定した値以上になる必要があります。この最小成功値が満たされませんでした。

処置：2 次エラーを参照してください。

### 4950、IP {1} とポート {2} を持つホスト {0} にエラーが発生しました

原因：サーバーでエラーが発生しました。

処置：2 次エラーを参照してください。

## B.2.9 警告エラー・メッセージ

警告エラー・メッセージは、次のとおりです。

### 15305、警告：バックアップのサマリー情報を収集中に問題が発生しました。

原因：トポロジのバックアップ操作の gatherInfo 手順中にエラーが発生しました。

処置：2 次エラーを参照してください。

### 15306、取消し処理を実行中に警告が発行されました。

原因：取消し処理を実行中にエラーが発生しました。

処置：2 次エラーを参照してください。

## B.2.10 OracleAS データベース・エラー・メッセージ

OracleAS データベース・エラー・メッセージは、次のとおりです。

### 15604、フィジカル・スタンバイ・データベースの作成を完了中に発生したエラー

原因：スタンバイ・データベースの作成の完了に失敗しました。

処置：2 次エラーを参照してください。

### 15605、フィジカル・スタンバイ・データベースを作成中に発生したエラー

原因：スタンバイ・データベースの作成に失敗しました。

処置：2 次エラーを参照してください。

### 15606、プライマリ・トポロジで同期データベース操作の実行に失敗しました

原因：プライマリ・トポロジでのデータベースの同期化操作の実行に失敗しました。

処置：2 次エラーを参照してください。

### 15607、スタンバイ・トポロジで同期データベース操作の実行に失敗しました

原因：スタンバイ・トポロジでのデータベースの同期化操作の実行に失敗しました。

処置：詳細は、2 次エラーとログ・ファイルを参照してください。

- 15608、無効なバックアップ・モードがテンプレート・ファイル {0} に指定されました。**  
原因: テンプレート・ファイルのバックアップ・モード値の処理中にエラーが発生しました。  
処置: バックアップ・モードを修正して、再度試行してください。
- 15609、データベース・バックアップ・ファイルの取得に失敗しました。**  
原因: データベース・バックアップ・ファイルの取得中にエラーが発生しました。  
処置: 2 次エラーを参照してください。

## B.2.11 OracleAS トポロジ・エラー・メッセージ

OracleAS トポロジ・エラー・メッセージは、次のとおりです。

- 15620、無効なトポロジが指定されました**  
原因: トポロジ・オブジェクトの処理中にエラーが発生しました。  
処置: 有効なトポロジ・オブジェクトを取得してください。
- 15621、トポロジ {0} の検証を試行中に発生したエラー**  
原因: 指定されたトポロジで検証操作中にエラーが発生しました。  
処置: 詳細は、2 次エラーを参照してください。
- 15622、インスタンス {0} の検証を試行中に発生したエラー**  
原因: 指定されたインスタンスで検証操作中にエラーが発生しました。  
処置: 詳細は、2 次エラーを参照してください。
- 15623、トポロジ {0} はトポロジ {1} と対称的ではありません**  
原因: 指定されたトポロジは対称的ではありません。  
処置: 詳細は、2 次エラーを参照してください。
- 15624、無効なトポロジが指定されました。トポロジ {0} に有効なインスタンスが含まれていません**  
原因: トポロジ・オブジェクトの処理中にエラーが発生しました。トポロジ・オブジェクトには、有効なインスタンスが含まれていませんでした。  
処置: 最低でも 1 つのインスタンスを持つ有効なトポロジ・オブジェクトを取得してください。
- 15625、ファーム {1} 内で一致するインスタンス {0} を検出できませんでした**  
原因: 一致するインスタンスを取得できませんでした。両トポロジは対称的ではないようです。  
処置: 両トポロジを対称的にしてください。
- 15626、トポロジ名 {0} がトポロジ名 {1} と同じではないため、トポロジは対称的ではありません。**  
原因: トポロジ名が同じでないため、両トポロジは対称的ではありません。  
処置: 両トポロジを対称的にしてください。
- 15627、異なる Oracle ホーム名 {1}、{2} のため、インスタンス {0} は対称的ではありません。**  
原因: 指定された両トポロジのインスタンス・ホーム名は対称的ではありません。  
処置: 両トポロジを対称的にしてください。
- 15628、異なる Oracle ホーム・パス {1}、{2} のため、インスタンス {0} は対称的ではありません。**  
原因: 指定された両トポロジのインスタンス・ホーム・パスは対称的ではありません。  
処置: 両トポロジを対称的にしてください。

**15629、異なるホスト名 {1}、{2} のため、インスタンス {0} は対称的ではありません。**

**原因:** 指定された両トポロジのインスタンス・ホスト名は対称的ではありません。

**処置:** 両トポロジを対称的にしてください。

**15630、指定したインスタンス {0} を検出できませんでした。**

**原因:** このノードでは、指定されたインスタンス情報を検出できませんでした。

**処置:** 間違ったインスタンス名またはホスト名が、サーバーへのリクエストで指定されています。

**15631、プライマリおよびスタンバイ・トポロジは、両方ともホスト {1} 上にインスタンス {0} を持つため、同じである可能性があります**

**原因:** 1つのインスタンスは、1つのトポロジのメンバーにしかなれないため、プライマリおよびスタンバイ・トポロジは同一のようです。

**処置:** プライマリ・トポロジとは別のスタンバイ・トポロジを指定してください。

**15632、インスタンス {0} を含むホームは見つかりませんでした。**

**原因:** このノードのどのホームにも、指定されたインスタンスが見つかりませんでした。

**処置:** システムの Oracle ホーム情報が正しくありません。

**15633、無効なトポロジが指定されました。トポロジには {0} という重複したインスタンスが含まれています**

**原因:** OPMN から取得したトポロジ情報に、重複するインスタンスが含まれています。

**処置:** トポロジ情報が正しいか、OPMN を確認してください。

## B.2.12 OracleAS バックアップおよびリストアのエラー・メッセージ

OracleAS バックアップおよびリストアのエラー・メッセージは、次のとおりです。

**15681、バックアップ・ディレクトリを指定してください。**

**原因:** 操作を完了するには、バックアップ・ディレクトリを指定する必要があります。

**処置:** 2次エラーを参照してください。

**15682、構成ファイルの初期化に失敗しました: {0}。**

**原因:** バックアップ・スクリプトの構成ファイルの初期化に失敗しました。

**処置:** 2次エラーを参照してください。

**15683、Oracle ホーム {0} に HA ディレクトリが存在しません。**

**原因:** OracleAS の Oracle ホームに HA ディレクトリが存在しません。

**処置:** バックアップおよびリストア・スクリプトを含む HA ディレクトリが、OracleAS の Oracle ホームにコピーされていることを確認してください。

**15684、バックアップ・スクリプトとリストア・スクリプトの構成ファイルの生成に失敗しました。**

**原因:** バックアップ・スクリプトとリストア・スクリプトの構成ファイルの生成に失敗しました。

**処置:** 2次エラーを参照してください。

**15685、インスタンス {0} の構成データのバックアップに失敗しました。**

**原因:** 指定されたインスタンスの構成データのバックアップに失敗しました。

**処置:** 2次エラーを参照してください。

**15686、インスタンス {0} の構成データのリストアに失敗しました。**

**原因:** 指定されたインスタンスの構成データのリストアに失敗しました。

**処置:** 2次エラーを参照してください。

- 15687、データベース・バックアップ・ファイルの取得に失敗しました。**  
原因: ログからのデータベース・バックアップ・ファイル名の取得に失敗しました。  
処置: 2 次エラーを参照してください。
- 15688、構成スクリプトを実行中にエラーが発生しました。**  
原因: 構成スクリプトの実行に失敗しました。  
処置: 構成スクリプトによって生成されたログ・ファイルを確認してください。
- 15689、バックアップ・スクリプトを実行中にエラーが発生しました。**  
原因: バックアップ・スクリプトの実行に失敗しました。  
処置: バックアップ・スクリプトによって生成されたログ・ファイルを確認してください。
- 15690、リストア・スクリプトを実行中にエラーが発生しました。**  
原因: リストア・スクリプトの実行に失敗しました。  
処置: リストア・スクリプトによって生成されたログ・ファイルを確認してください。
- 15691、ZIP ファイルは検出されませんでした**  
原因: ZIP ファイルは検出されませんでした。  
処置: バックアップが正常に実行されていることを確認してください。
- 15692、構成ファイル {0} は空です。**  
原因: 指定された構成ファイルが空です。  
処置: 元の構成ファイルを、バックアップ・リストア・スクリプトがある ha ディレクトリからコピーしてください。
- 15693、ZIP ファイルが指定されていません。**  
原因: ユーザーが、ZIP の解凍操作に ZIP ファイルを指定していません。  
処置: 内部エラーです。
- 15694、実行中の手順のエラー - バックアップ・トポロジの {0}**  
原因: トポロジのバックアップが指定された手順で失敗しました。  
処置: 2 次エラーを参照してください。
- 15695、実行中の手順のエラー - リストア・トポロジの {0}**  
原因: トポロジのリストアが指定された手順で失敗しました。  
処置: 2 次エラーを参照してください。
- 15696、トポロジのバックアップ操作初期設定中にエラーが発生しました**  
原因: トポロジのバックアップ操作を初期設定中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 15697、トポロジのバックアップ操作実行中のエラー - バックアップ手順**  
原因: トポロジのバックアップのバックアップ手順の処理中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 15698、トポロジのバックアップ操作実行中のエラー - コピー手順**  
原因: トポロジのバックアップのコピー手順の処理中にエラーが発生しました。  
処置: 2 次エラーを参照してください。
- 15699、トポロジのリストア操作を初期設定中にエラーが発生しました**  
原因: トポロジのリストア操作を初期設定中にエラーが発生しました。  
処置: 2 次エラーを参照してください。

**15700、バックアップ・ファイルが見つかりませんでした**

原因: バックアップ・ファイルが見つかりませんでした。

処置: バックアップが正常に実行されていることを確認してください。

**15701、インスタンス "{0}" の DCM-resyncforce オプションでの構成のリストアに失敗しました**

原因: DCM の -resyncforce オプションによる構成のリストアに失敗しました。

処置: 2 次エラーを参照してください。

**15702、クローン・インスタンスの操作を初期化中にエラーが発生しました**

原因: クローン・インスタンスの操作を初期化中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**15703、クローン・トポロジの操作を初期設定中にエラーが発生しました**

原因: クローン・トポロジの操作を初期化中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**15704、エラー。クローン化されるインスタンスのホーム {0} はすでに存在しています**

原因: インスタンスのクローニングでエラーが発生しました。Oracle ホームはすでに存在します。

処置: Oracle ホームをクリーンアップして再試行してください。

**15705、インスタンス {0} のクローニングでエラーが発生しました。クローニングでは OPMN を停止する必要があるため、asgctl を使用して OracleAS Guard サーバーを起動する必要があります**

原因: クローニングするには OPMN を停止する必要があります。これが原因で OracleAS Guard サーバー (ASG サーバー・プロセス) が停止します。これによってクローニングが失敗します。

処置: opmnctl を使用して OracleAS Guard サーバー (ASG サーバー・プロセス) を停止します。次に asgctl startup topology コマンドを使用して、OracleAS Guard サーバーをこのインスタンス用に再起動します。

**15706、ユーザー・リクエストによるレスポンスのバックアップ・ホーム・イメージが停止します**

原因: ユーザーが NO を入力したため、ホームのバックアップ操作を停止します。

処置: なし。

**15707、ユーザー・リクエストによるレスポンスのリストア・ホーム・イメージが停止します**

原因: ユーザーが NO を入力したため、ホームのリストア操作を停止します。

処置: なし。

## B.2.13 OracleAS Guard 同期化エラー・メッセージ

OracleAS Guard 同期化エラー・メッセージは、次のとおりです。

**15721、DUF データベース・オブジェクトの初期化に失敗しました。**

原因: Duf データベース・オブジェクトの初期化に失敗しました。

処置: 2 次エラーを参照してください。

**15722、トポロジ操作の実行に使用できるトポロジ情報がありません**

原因: トポロジ操作の実行に使用できるトポロジ情報がありません。

処置: 2 次エラーを参照してください。

**15723、トポロジのバックアップ・リストにインスタンスが見つかりません**

原因: トポロジのバックアップ・リストが空です。

処置: 2 次エラーを参照してください。

**15724、スタンバイ・ホストのリストの取得に失敗しました**

原因: スタンバイ・ホストのリストの取得に失敗しました。

処置: 2 次エラーを参照してください。

**15725、トポロジ {0} の OracleAS 構成データのバックアップに失敗しました**

原因: OracleAS トポロジの構成データのバックアップに失敗しました。

処置: 2 次エラーを参照してください。

**15726、トポロジ {0} の OracleAS 構成データのリストアに失敗しました**

原因: OracleAS トポロジの構成データのリストアに失敗しました。

処置: 2 次エラーを参照してください。

**15727、OracleAS インフラストラクチャ・データベース {0} のバックアップに失敗しました**

原因: OracleAS トポロジのインフラストラクチャ・データベースのバックアップに失敗しました。

処置: 2 次エラーを参照してください。

**15728、OracleAS インフラストラクチャ・データベース {0} のリストアに失敗しました**

原因: OracleAS トポロジのインフラストラクチャ・データベースのリストアに失敗しました。

処置: 2 次エラーを参照してください。

**15729、トポロジの同期化操作の実行に失敗しました**

原因: トポロジの同期化操作の実行に失敗しました。

処置: 2 次エラーを参照してください。

## B.2.14 OracleAS Guard インスタンス化エラー・メッセージ

OracleAS Guard インスタンス化エラー・メッセージは、次のとおりです。

**15751、トポロジのインスタンス化操作の手順 {0} を実行中にエラーが発生しました**

原因: トポロジのインスタンス化操作が、指定された手順で失敗しました。

処置: 2 次エラーを参照してください。

**15752、リモート・トポロジ情報のロードに失敗しました**

原因: リモート・トポロジ情報のロードに失敗しました。

処置: ユーザーがトポロジの正しいホスト名を指定していること、OPMN プロセスがトポロジで実行されていることを確認してください。

**15753、ホスト {0} でトポロジのインスタンス化の準備中にエラーが発生しました**

原因: トポロジのインスタンス化の準備中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**15754、データベース {0} のインスタンス化中にエラーが発生しました。**

原因: データベースをインスタンス化中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**15755、データベース {0} のインスタンス化を完了中にエラーが発生しました。**

原因: データベースのインスタンス化を完了中にエラーが発生しました。

処置: 2 次エラーを参照してください。

**15756、トポロジのインスタンス化操作の初期設定中にエラーが発生しました**

原因: トポロジのインスタンス化操作の初期設定中にエラーが発生しました。

処置: 2 次エラーを参照してください。

- 15757、トポロジのスイッチオーバー操作を初期設定中にエラーが発生しました**  
原因：トポロジのスイッチオーバー操作の初期設定中にエラーが発生しました。  
処置：2 次エラーを参照してください。
- 15770、トポロジ・ファイルに指定されたインスタンス {0} が、ホーム {2} のインスタンス {1} と一致しません**  
原因：トポロジ・ファイルが正しくありません。  
処置：asgctl から discover topology コマンドを実行します。
- 15771、トポロジ・ファイル {0} はバージョンが間違っています。ファイルを削除して、トポロジを再検出してください**  
原因：トポロジ・ファイルが正しくありません。  
処置：asgctl から discover topology コマンドを実行します。
- 15772、トポロジ・ファイル {0} には、検出ホスト {1} のエントリがありません**  
原因：トポロジ・ファイルが正しくありません。  
処置：asgctl から discover topology コマンドを実行します。
- 15773、スタンバイ・トポロジに、プライマリ・インスタンス {0} のエントリがありません**  
原因：トポロジ・ファイルが正しくありません。  
処置：asgctl から discover topology コマンドを実行します。
- 15774、データベース {1} のスタンバイ・トポロジ・ネット記述子のホスト名 {0} はプライマリ・ホスト・アドレス {2} で解決します**  
原因：トポロジ・ファイルが正しくありません。  
処置：asgctl から discover topology コマンドを実行します。
- 15775、インスタンス {1} のスタンバイ・トポロジのホスト名 {0} はプライマリ・ホスト・アドレスで解決します**  
原因：トポロジ・ファイルが正しくありません。  
処置：asgctl から discover topology コマンドを実行します。
- 15776、OID サーバーのアクセス中にエラーが発生しました**  
原因：OID サーバーにアクセスできません。  
処置：正しい OID 情報を指定して、OID サーバーが実行されていることを確認します。
- 15777、エラー：サーバーにアクセスするために必要な OID 情報が指定されませんでした**  
原因：OID サーバーにアクセスできません。  
処置：正しい OID 情報を指定します。
- 15778、SID{0} のデータベース情報をホスト {1} から取得中にエラーが発生しました。このインスタンスは topology.xml ファイルから除外されます**  
原因：トポロジ・データベースのデータベース情報を取得できません。  
処置：なし。
- 15779、インスタンス {0} のインスタンス情報をホスト {1} から取得中にエラーが発生しました。このインスタンスは topology.xml ファイルから除外されます**  
原因：インスタンスの情報を取得できません。  
処置：なし。
- 15780、インスタンス {0} がトポロジに見つかりません**  
原因：トポロジ・ファイルにインスタンス名が存在しません。  
処置：asgctl から discover topology コマンドを実行します。

**15781、警告: ホーム {1} のホスト {0} でトポロジ・ファイルを更新できません**

**原因:** トポロジ・ファイルをホームに書き込むことができません。

**処置:** OracleAS Guard サーバーがそのホームで実行されていることを確認してください。

**15782、エラー: インスタンス {0} はトポロジにすでに存在しています**

**原因:** インスタンスがトポロジにすでに存在します。

**処置:** asgctl を使用してインスタンスを削除してください。

**15783、エラー: サーバーにアクセスするために必要な OID ホストおよびポート情報が指定されませんでした**

**原因:** OID サーバーにアクセスできません。

**処置:** 正しい OID 情報を指定します。

---

---

# Oracle Notification Server (ONS) を使用せずに 管理する Oracle Application Server Cluster

この付録では、Oracle Notification Server (ONS) を使用せずに OracleAS Cluster を設定および実行する方法を説明します。この方法では、Oracle Application Server インスタンスのグループをクラスタとして手動で作成および管理します。自動化した方法については、[第 3 章「アクティブ / アクティブ・トポロジ」](#) で説明しています。

この付録の項目は次のとおりです。

- [第 C.1 項「ONS を使用せずに管理する OracleAS Cluster について」](#)
- [第 C.2 項「構成作業」](#)

## C.1 ONS を使用せずに管理する OracleAS Cluster について

Oracle Application Server インスタンスのグループのクラスタリングは、ルーティングを目的として行います。クラスタ内の 1 つのインスタンスに障害が発生すると、そのクラスタ内の別のインスタンスにリクエストがルーティングされます。ONS を使用せずに OracleAS Cluster を管理する場合、ユーザーは管理者としてクラスタを手動で構成する必要があります。これを行うには、構成ファイルを編集します。

ONS を使用しない OracleAS Cluster の管理は、Oracle Application Server Java Edition のユーザーを対象としています。Oracle Application Server Java Edition では、ONS を使用して管理する OracleAS Cluster を利用できません。

ONS を使用して管理する OracleAS Cluster で利用でき、ONS を使用せずに管理する OracleAS Cluster では利用できない機能の一覧を次に示します。

- 動的検出: ONS を使用して管理する OracleAS Cluster では、マルチキャスト・アドレスを使用してクラスタを定義します。同じマルチキャスト・アドレスを使用するインスタンスが、自動的に 1 つのクラスタに結合されます。

ONS を使用せずに管理する OracleAS Cluster では、各インスタンスで構成ファイルを編集して、クラスタにインスタンスを追加または削除する必要があります。

- アプリケーションのマウント・ポイントの自動構成: ONS を使用して管理する OracleAS Cluster では、アプリケーションのマウント・ポイントが自動的に管理されます。

ONS を使用せずに管理する OracleAS Cluster では、各インスタンスで構成ファイルを編集して、アプリケーションのマウント・ポイントを追加または削除する必要があります。

- OPMN が管理する動的な peer-to-peer レプリケーション: ONS を使用して管理する OracleAS Cluster では、`orion-application.xml` ファイルにタグを追加することで、レプリケーションの対象となるピアのリストが OPMN によって動的に提供されるように指定できます。

ONS を使用せずに管理する OracleAS Cluster では、静的な peer-to-peer レプリケーションを使用できます。

## C.2 構成作業

ONS を使用せずに管理する OracleAS Cluster を設定するには、次の構成手順を実行します。

- 静的な peer-to-peer レプリケーションの構成

静的な peer-to-peer レプリケーションは、対象となるピアを `orion-application.xml` ファイルにリストして構成します。

詳細は、『Oracle Containers for J2EE 構成および管理ガイド』の第 9 章「OC4J でのアプリケーションのクラスタリング」の「静的 Peer-to-Peer レプリケーションの構成」を参照してください。

- マウント・ポイントの作成

マウント・ポイントは、`ORACLE_HOME/Apache/Apache/conf/mod_oc4j.conf` ファイルを編集して作成します。`Oc4jMount` ディレクティブを使用してマウント・ポイントを定義します。ONS を使用して管理する OracleAS Cluster では、マウント・ポイントは自動的に管理され、`mod_oc4j.conf` ファイルには出力されません。

マウント・ポイントの作成の詳細は、『Oracle HTTP Server 管理者ガイド』の第 7.32 項「`mod_oc4j`」を参照してください。

---

- URL ベースでの特定のインスタンスへのルーティング構成

ORACLE\_HOME/Apache/Apache/conf/mod\_oc4j.conf ファイルを編集すると、リクエストを特定のインスタンスに転送するように `mod_oc4j` を構成できます。Oc4jMount ディレクティブの `destination` パラメータを使用して、アプリケーションのリクエストの転送先を指定します。

転送先の指定の詳細は、『Oracle HTTP Server 管理者ガイド』の第 7.32 項「`mod_oc4j`」を参照してください。

ONS を使用せずに管理する OracleAS Cluster では、ルーティングまたはレプリケーションに OPMN のメンバーシップは使用されません。これは、ルーティングおよびセッションのレプリケーションの両方がハードコードされているためです。



## 数字

---

1 次ノード, 1-6

2 次ノード, 1-6

## A

---

Active Data Cache (Oracle Business Activity Monitoring), 5-16, 5-19

admin\_client.jar

クラスタへのアプリケーションのデプロイ, 3-20

グループへのアプリケーションのデプロイ, 3-10

asgctl コマンド

add instance, 7-7

asgctl, 7-9

clone instance, 6-25, 6-35, 7-10

clone topology, 6-25, 6-35, 7-13

connect asg, 7-16

create standby database, 7-17

disconnect, 7-19

discover topology, 6-24, 6-31, 6-66, 7-20

discover topology within farm, 6-24, 7-22

dump farm (廃止済), 7-58

dump policies, 6-25, 7-23

dump topology, 6-26, 6-31, 7-24

exit, 7-26

failover, 6-25, 6-45, 7-27

help, 6-65, 7-28

instantiate farm (廃止済), 7-59

instantiate topology, 6-25, 7-29

quit, 7-31

remove instance, 7-32

run, 7-34

set asg credentials, 6-24, 6-31, 6-68, 7-35

set echo, 7-37

set new primary database, 6-24, 6-31, 7-38

set noprompt, 7-39

set primary database, 6-24, 6-30, 6-66, 6-69, 7-40

set trace, 7-42

show env, 7-43

show operation, 6-26, 7-44

shutdown, 6-25, 7-46

shutdown farm (廃止済), 7-60

shutdown topology, 7-47

startup, 7-48

startup farm (廃止済), 7-61

startup topology, 6-25, 7-49

stop operation, 6-26, 7-50

switchover farm (廃止済), 7-62

switchover topology, 6-25, 6-43, 7-51

sync farm (廃止済), 7-64

sync topology, 6-25, 7-54

verify farm (廃止済), 7-65

verify topology, 6-25, 6-32, 7-56

一部に特有の情報, 7-4

共通報, 7-3

asgctl コマンドに共通な情報, 7-3

asgctl スクリプト

作成と実行, 6-67

asgctl スクリプトの作成と実行, 6-67

asgctl 操作の監視, 6-61

asgctl 操作の停止, 6-63

asgctl タスクのトレース, 6-63

## B

---

Backup and Recovery Tool, 8-4

## C

---

CFC 環境

障害リカバリ構成に関する特別な考慮事項, 7-5

特別な考慮事項

インスタンス化およびフェイルオーバー機能

, 7-5

スイッチオーバー操作, 7-6

clone instance コマンド, 6-35, 6-37, 7-10

clone topology コマンド, 6-25, 6-35, 6-37, 7-13

config.inp, 8-4

connect asg コマンド, 7-16

create standby database コマンド (OracleAS Disaster Recovery), 7-17

createinstance コマンド, 3-7

## D

---

disconnect コマンド (OracleAS Disaster Recovery), 7-19

discover topology (OracleAS Disaster Recovery)

, 6-24, 6-31, 6-66

コマンド, 7-20

discover topology within farm (OracleAS Disaster Recovery), 6-24

コマンド, 7-22

DNS, 6-18, 6-20

DNS 解決 (OracleAS Disaster Recovery の場合)

, 6-20  
OracleAS Disaster Recovery での使用, 6-17  
スイッチオーバー (OracleAS Disaster Recovery の場合), 6-45  
本番サイトおよびスタンバイ・サイトの DNS サーバー (OracleAS Disaster Recovery の場合), 6-20  
マッピング (OracleAS Disaster Recovery の場合), 6-65  
dump farm コマンド (廃止済), 7-58  
dump policies コマンド, 6-25, 6-32, 7-23  
dump topology コマンド, 6-26, 6-31, 6-64, 7-24

## E

EJB  
クライアント・ルーティング, 3-25  
セッション状態レプリケーション, 3-4  
レプリケーション, 3-5  
Enterprise Link (Oracle Business Activity Monitoring), 5-17, 5-20, 5-25  
ESB ランタイム・サーバー, 5-14  
ESB リポジトリ・サーバー, 5-14  
Event Engine (Oracle Business Activity Monitoring), 5-17, 5-19  
exit コマンド (OracleAS Disaster Recovery), 7-26

## F

failover コマンド (OracleAS Disaster Recovery), 6-8, 6-25, 6-45, 7-27

## H

help コマンド (OracleAS Disaster Recovery), 7-28

## I

IIS の設定、Oracle Business Activity Monitoring, 5-25  
Infrastructure データベースの認証  
新しい本番サイトへのフェイルオーバー, 6-24, 6-31  
本番サイト, 6-24, 6-30  
instantiate farm コマンド (廃止済), 7-59  
instantiate topology (OracleAS Disaster Recovery), 6-25, 6-40  
セカンダリ・サイト, 6-40  
ポリシー・ファイル, 6-40, 6-41  
instantiate topology コマンド, 7-29

## J

J2CA ベースのアダプタ, 5-7  
J2EE and Web Cache インストール・タイプ, 6-5  
Java Message Service, 3-24  
Java Object Cache, 3-25  
jgroups-protocol.xml ファイル, 5-5  
JNDI ネームスペースのレプリケーション, 3-25

## L

LVS ロード・バランサ, 2-6

## M

Microsoft Cluster Server, 4-2  
Oracle Business Activity Monitoring に対する構成, 5-20  
mod\_oc4j  
デフォルトのロード・バランシング・アルゴリズム, 3-10  
ロード・バランシング・アルゴリズムの選択, 3-23  
ロード・バランシング・オプション, 3-21  
mod\_oc4j.conf  
Oc4jRoutingWeight ディレクティブ, 3-22  
Oc4jSelectMethod ディレクティブ, 3-22  
ロード・バランシングの構成, 3-22

## N

NIS, 6-18

## O

OC4J  
Java Object Cache, 3-25  
JNDI ネームスペースのレプリケーション, 3-25  
mod\_oc4j, 3-21  
Oracle HTTP Server からのリクエストの受信, 3-10  
分散キャッシュ, 3-25  
ロード・バランシング, 3-21  
Oc4jRoutingWeight ディレクティブ, 3-22  
Oc4jSelectMethod ディレクティブ, 3-22  
OC4J インスタンス  
作成, 3-7  
opmn: 接頭辞, A-8  
opmnctl config topology コマンド, 3-14  
Oracle BPEL Process Manager, 5-2  
アクティブ / アクティブ・トポロジ, 5-3  
アクティブ / パッシブ・トポロジ, 5-7  
アダプタ, 5-7  
異なるサブネットにあるマシンでの実行, 5-5  
Oracle Business Activity Monitoring, 5-16  
Active Data Cache, 5-16, 5-19  
Enterprise Link, 5-17, 5-20, 5-25  
Event Engine, 5-17, 5-19  
IIS の設定, 5-25  
Microsoft Cluster Server の構成, 5-20  
Plan Monitor, 5-17, 5-20  
Report Cache, 5-17, 5-19  
Web アプリケーション, 5-17, 5-20  
既知の問題とトラブルシューティング, 5-25  
メッセージ整合の設定, 5-28  
Oracle Business Activity Monitoring に対する Web ガーデン (IIS 6) の設定, 5-25  
Oracle Business Rules, 5-31  
Real Application Clusters データベース, 5-32  
リポジトリのタイプ, 5-32  
リポジトリへの WebDAV URL, 5-32  
Oracle Data Guard, 6-1, 6-8, 6-22, 8-3, 11-2  
Oracle Enterprise Service Bus, 5-12  
Oracle ホーム, 5-14  
Real Application Clusters データベース, 5-15  
アクティブ / アクティブ・トポロジ, 5-12  
アダプタ, 5-16  
アプリケーションの登録, 5-16

- インストールおよび構成手順, 5-15
- 仮想ホスト名の検証, 5-15
- ランタイム・サーバー, 5-14
- リポジトリ・サーバー, 5-14
- Oracle Fail Safe, 4-2
- Oracle HTTP Server
  - OC4J へのリクエストのルーティング, 3-10
  - リリース 2 (10.1.2), 3-13
- Oracle Identity Management
  - アクティブ / アクティブ・トポロジでの使用, 3-11
- Oracle Notification Server を使用しないクラスタリング, C-1
- Oracle Service Registry, 5-31
- Oracle SOA Suite, 5-1
- Oracle Web Services Manager, 5-32
- OracleAS Cluster, 2-9
  - Oracle Notification Server を使用しない管理, C-1
  - peer-to-peer レプリケーション, 3-16
  - アクティブ / アクティブ・トポロジ内, 3-3
  - 検出サーバー方法, 3-14
  - 設定, 3-14
  - データベースへのレプリケーション, 3-18
  - 動的検出方法, 3-14
  - マルチキャスト・レプリケーション, 3-15
  - メリット, 2-10
- OracleAS Cluster (OC4J)
  - EJB の状態レプリケーション, 3-4
  - Web アプリケーションのセッション状態レプリケーション, 3-5
  - 状態レプリケーション, 2-2
- OracleAS Cluster (OC4J-EJB), 3-5
  - 複数の NIC, A-7
- OracleAS Cold Failover Cluster, 2-10, 6-7, 6-17, 6-21
  - 環境, 1-6
  - クラスタレベルのメンテナンス, 6-43
  - メリット, 2-11
  - 「アクティブ / パッシブ・トポロジ」も参照
- OracleAS Disaster Recovery, 6-1
  - Infrastructure のみの非対称のスタンバイ, 6-9
  - 同じ場所にある OID および MR と別の場所にある MR, 6-10
  - 同じ場所のない OID および MR と分散したアプリケーション MR, 6-11
  - 構成手順
    - Real Application Clusters データベースを使用, 6-49
    - Real Application Clusters データベースを使用しない, 6-47
    - プライマリ・サイトのみで Real Application Clusters データベースを使用, 6-56
    - サイトのフル・アップグレード手順, 9-2
    - サイトのフル・アップグレードの前提条件, 9-2
    - サポートされているトポロジ, 6-5
    - スイッチオーバー手順
      - Real Application Clusters データベースを使用, 6-53
      - Real Application Clusters データベースを使用しない, 6-48
      - プライマリ・サイトのみで Real Application Clusters データベースを使用, 6-59
    - スイッチバック手順
      - Real Application Clusters データベースを使用, 6-54
  - Real Application Clusters データベースを使用しない, 6-49
  - プライマリ・サイトのみで Real Application Clusters データベースを使用, 6-60
  - スタンバイ・サイトが同期化されない, A-2
  - 中間層の数が少ない非対称スタンバイ, 6-8
  - ディレクトリが見つからないためスイッチオーバー操作に失敗する, A-3
  - 非対称スタンバイ構成, 6-8
  - ファーム検証操作に失敗する, A-4
  - ファームの同期操作に失敗する, A-5
  - フェイルオーバー後にスタンバイが起動しない, A-2
  - フェイルオーバー手順
    - Real Application Clusters データベースを使用, 6-56
    - Real Application Clusters データベースを使用しない, 6-49
    - プライマリ・サイトのみで Real Application Clusters データベースを使用, 6-61
    - 本番とスタンバイが対称になっている構成, 6-6
- OracleAS Disaster Recovery サイトのフル・アップグレードの前提条件, 9-2
- OracleAS Disaster Recovery のアップグレード手順, 9-2
- OracleAS Guard, 6-28, 6-40
  - asgctl の起動, 7-9
  - asgctl コマンドの説明の要約, 7-1
  - asgctl スクリプトの作成と実行, 6-67
  - asgctl 操作の監視, 6-61
  - asg 資格証明の設定, 6-68
  - run, 7-34
  - 一般的な asgctl セッション, 6-65
  - インスタンスの削除, 7-32
  - インスタンスの追加, 7-7
  - エラー・メッセージ, 6-64
  - クライアント, 6-23
  - 現在の操作の表示, 6-62
  - サーバー, 6-24
  - サポートされる OracleAS リリース, 6-5
  - サポートされる障害時リカバリ構成, 6-28
  - スタンバイ・トポロジへのスイッチオーバー, 6-43
  - スタンバイ・トポロジへのフェイルオーバー, 6-45
  - セカンダリ・サイトでのインスタンスのクローニング, 6-37
  - セカンダリ・サイトでのトポロジのインスタンス化, 6-40
  - セカンダリ・サイトでのトポロジのクローニング, 6-37
  - セカンダリ・サイトをプライマリ・サイトに同期化, 6-41
  - 操作, 6-24
  - 操作の停止, 6-63
  - 操作履歴の表示, 6-63
  - タスクのトレース, 6-63
  - トポロジの詳細情報の表示, 6-64
  - 廃止された asgctl コマンドの説明の要約, 7-1
  - プライマリ・データベースの指定, 6-66
  - プライマリ・トポロジ構成の検証, 6-61
  - プライマリ・トポロジの Infrastructure データベースの指定, 6-69
  - プライマリ・トポロジの検証, 6-61
  - ヘルプの参照, 6-65

OracleAS Guard サーバーから OracleAS Guard クライアントへの認証, 6-24, 6-31  
OracleAS Recovery Manager, 8-4, 8-5  
OracleAS Web Cache  
スタンバイ・サイトで起動しない, A-4  
リパース・プロキシ・サーバーとして機能, 3-13  
リリース 2 (10.1.2), 3-13  
oracle.j2ee.naming.cache.timeout プロパティ, A-8

## P

Plan Monitor (Oracle Business Activity Monitoring), 5-17, 5-20  
Portal and Wireless インストール・タイプ, 6-5  
PROVIDER\_URL プロパティ, A-8

## Q

quit コマンド (OracleAS Disaster Recovery), 7-31

## R

Real Application Clusters データベース  
Oracle BPEL Process Manager のデハイドレーション・ストア, 5-4  
Oracle Business Rules, 5-32  
Oracle Enterprise Service Bus, 5-15  
OracleAS Disaster Recovery トポロジ, 6-49, 6-56  
Recovery Manager, 6-1, 8-2  
Report Cache (Oracle Business Activity Monitoring), 5-17, 5-19  
Rule Author, 5-32

## S

SCN, 8-3, 8-6  
Secure Shell ポート・フォワーディング, 11-2  
set asg credentials コマンド, 6-24, 6-31, 6-68, 7-35  
set echo コマンド, 7-37  
set new primary database コマンド, 6-24, 6-31, 7-38  
set noprompt コマンド, 7-39  
set primary database コマンド, 6-24, 6-30, 6-66, 6-69, 7-40  
set trace off コマンド, 6-63  
set trace on コマンド, 6-63  
set trace コマンド, 7-42  
show env コマンド, 7-43  
show operation full コマンド, 6-62  
show operation history コマンド, 6-63  
show operation コマンド, 6-26, 7-44  
shutdown farm コマンド (廃止済), 7-60  
shutdown topology コマンド, 6-25, 7-47  
shutdown コマンド, 7-46  
SSH トンネリング, 11-2  
startup farm コマンド (廃止済), 7-61  
startup topology コマンド, 6-25, 7-49  
startup コマンド, 7-48  
staticports.ini ファイル, 6-22  
stop operation コマンド, 6-26, 6-63, 7-50  
switchover farm コマンド (廃止済), 7-62  
switchover topology to コマンド, 6-8  
switchover topology コマンド, 6-25, 6-43, 7-51  
sync farm コマンド, 7-64

sync topology コマンド, 6-25, 7-54  
synchronize topology コマンド, 6-41

## T

TNS 名, 6-22  
trigger 属性 (replication-policy 用), 3-18  
TTL, 6-64

## U

UDDI  
「Oracle Service Registry」を参照

## V

verify farm コマンド (廃止済), 7-65  
verify topology コマンド, 6-25, 6-32, 6-61, 7-56

## W

WebDAV URL、Oracle Business Rules リポジトリ, 5-32  
Web アプリケーション (Oracle Business Activity Monitoring), 5-17, 5-20  
Web アプリケーションのセッション状態レプリケーション, 3-5  
Wide Area Network, 6-3  
Windows ネットワーク・ロード・バランサ, 2-6  
write-quota 属性 (cluster タグ内), 3-19

## あ

アーカイブ・ログ, 6-2  
手動転送, 8-3  
アクティブ / アクティブ・トポロジ, 1-3, 2-9, 3-1  
Oracle BPEL Process Manager, 5-3  
Oracle Enterprise Service Bus, 5-12  
Oracle HTTP Server リリース 2 (10.1.2) の使用, 3-13  
Oracle Identity Management の使用, 3-11  
OracleAS Cluster, 3-3  
OracleAS Web Cache の使用, 3-13  
アプリケーションのデプロイ, 3-20  
インスタンスの削除, 3-21  
インスタンスの追加, 3-21  
管理, 3-13  
グループ, 3-6  
コンポーネントの起動と停止, 3-20  
アクティブ / アクティブ・トポロジ内のグループ, 3-6  
アプリケーションのデプロイ, 3-10  
インスタンスの管理, 3-9  
アクティブ化エージェント、無効, 5-10  
アクティブ / パッシブ・トポロジ, 1-3, 2-10, 4-1  
Application Server Control コンソール、使用, 4-5  
Oracle BPEL Process Manager, 5-7  
アプリケーションのデプロイ, 4-5  
管理, 4-5  
欠点, 4-3  
コンポーネントの起動と停止, 4-5  
メリット, 4-2  
「OracleAS Cold Failover Cluster」も参照  
アダプタ, 5-7

- J2CA ベース, 5-7
- Oracle Enterprise Service Bus, 5-16
  - アクティブ / アクティブ・トポロジ, 5-8
  - アクティブ / パッシブ・トポロジ, 5-10
  - 同時実行性のサポート, 5-8
  - 変更済アクティブ / アクティブ・トポロジ, 5-9
- アップグレードの前提条件
- OracleAS Disaster Recovery, 9-2
- アプリケーション
  - admin\_client.jar を使用した、グループへのデプロイ, 3-10
  - アクティブ / アクティブ・トポロジへのデプロイ, 3-20
  - アクティブ / パッシブ・トポロジへのデプロイ, 4-5
- 一部の asgctl コマンドに特有の情報, 7-4
- 一般的な asgctl セッション, 6-65
- インスタンス
  - アクティブ / アクティブ・トポロジからの削除, 3-21
  - アクティブ / アクティブ・トポロジへの追加, 3-21
- インスタンス管理 (OracleAS Disaster Recovery), 6-25
- エラー・メッセージ (OracleAS Disaster Recovery), 6-64

## か

- 仮想 IP, 1-7, 2-6, 2-10
  - OracleAS Disaster Recovery, 6-17, 6-21
- 仮想ホスト名, 1-7, 6-15, 6-20
  - OracleAS Disaster Recovery, 6-7, 6-14, 6-17, 6-18, 6-19, 6-20, 6-23
  - OracleAS Disaster Recovery の要件, 6-4
  - 検証 (Oracle Enterprise Service Bus), 5-15
- 共有記憶域, 1-6, 2-10
- クラスターアドミニストレータ (Microsoft Cluster Server), 5-20
- クラスターウェア, 1-6, 4-2
- クラスターエージェント, 1-6
- クラスターリング、アプリケーションレベル, 2-2
- peer-to-peer レプリケーション, 3-16
- セッション状態レプリケーション, 3-4
- データベースへのレプリケーション, 3-18
- マルチキャスト・レプリケーション, 3-15
- 計画外停止, 6-45
- 検出サーバー方法 (OracleAS Cluster 用), 3-14
- 検証操作, 6-25, 6-32
- 現在の操作の表示, 6-62
- 構成ファイル
  - OracleAS Disaster Recovery, 6-7
  - OracleAS Disaster Recovery によるバックアップ, 6-1
  - OracleAS Disaster Recovery のバックアップ, 8-4
- コンポーネント
  - アクティブ / アクティブ・トポロジでの起動と停止, 3-20
  - アクティブ / パッシブ・トポロジでの起動と停止, 4-5
  - ステータスのチェック, 3-20

## な

- サイトのアップグレード手順

- OracleAS Disaster Recovery, 9-1
- サポートされているトポロジ
- OracleAS Disaster Recovery, 6-5
- 障害時リカバリ, 6-1
- 順序番号, 8-3, 8-6
- 冗長性, 2-9
- スイッチオーバー, 1-7
- スイッチオーバー操作, 6-25
- スイッチバック, 1-7
- スケジューリングした停止, 6-42
- スタンバイ・サイト, 6-3, 6-4, 6-20
  - clone instance コマンド, 6-25
  - OracleAS Web Cache が起動しない, A-4
  - インスタンス化, 6-25
  - 中間層の間違ったホスト名, A-4
  - ディレクトリが見つからないためスイッチオーバー操作に失敗する, A-3
  - 同期, 6-25
  - 同期化されない, A-2
  - フェイルオーバー後に起動しない, A-2
  - リストア, 8-4
- ステータス、チェック, 3-20
- すべてのノードの操作履歴の表示, 6-63
- 静的な peer-to-peer レプリケーション, 3-16
- セカンダリ・サイトでのインスタンスのクローニング, 6-37
- セカンダリ・サイトでのトポロジのクローニング, 6-37
- セカンダリ・サイトをプライマリ・サイトに同期化, 6-41
- セッション状態レプリケーション, 3-4
- ソフトウェア・ロード・バランサ, 2-5

## た

- データベースのレプリケーション, 3-18
- デハイドレーション・ストア, 5-4
- 特別な考慮事項
  - CFC 環境での障害時リカバリ構成, 7-5
  - インスタンス化およびフェイルオーバー機能, 7-5
  - スイッチオーバー操作, 7-6
  - スイッチオーバー操作, 6-45
- トポロジのスイッチオーバー・ポリシー・ファイル, 6-43
- トポロジの同期化ポリシー・ファイル, 6-42
- トポロジのクローン・ポリシー・ファイル, 6-37
- トポロジの検証ポリシー・ファイル, 6-32, 6-41, 6-62
- トポロジの詳細情報の表示, 6-64
- トポロジのダンプ・ポリシー・ファイル, 6-32
- トラブルシューティング
  - 操作の停止, 6-26
  - 操作の表示, 6-26
  - トポロジのファイルへのダンプ, 6-26, 6-31
- 動的検出方法 (OracleAS Cluster 用), 3-14
- 動的な peer-to-peer レプリケーション, 3-16

## な

- ネットワーク・インタフェース・カード (NIC)
  - OracleAS Cluster (OC4J-EJB) での複数の使用, A-7
- ネットワーク・ホスト名, 1-6, 6-17
- ネットワーク・ロード・バランサ, 2-6

## は

---

ハードウェア・クラスタ, 1-5, 4-2  
ハードウェア・ロード・バランサ, 2-5  
ファーム検証操作に失敗する, A-4  
ファームの同期操作に失敗する, A-5  
フェイルオーバー, 1-5  
フェイルオーバー・ポリシー・ファイル, 6-46  
フェイルバック, 1-5  
フォルト・トレラント・モード, 2-6  
物理ホスト名, 1-6, 6-15, 6-16, 6-18  
プライマリ・データベースの指定, 6-66  
プライマリ・トポロジの Infrastructure データベースの指定, 6-69  
プライマリ・トポロジの検証, 6-61  
ヘルプの参照 (OracleAS Disaster Recovery), 6-65  
ホスト名  
  仮想, 1-7, 6-15  
  仮想 (OracleAS Disaster Recovery), 6-17, 6-18  
  ネットワーク, 1-6, 6-17  
  物理, 1-6, 6-15, 6-16, 6-18  
ホスト名解決, 6-18  
本番サイト, 6-3, 6-20  
  バックアップ, 8-2  
ポリシー・ファイル, 6-32  
  clone topology, 6-37  
  dump topology, 6-32  
  failover, 6-46  
  instantiate topology, 6-40, 6-41  
  switchover topology, 6-43  
  sync topology, 6-42  
  verify topology, 6-32, 6-41, 6-62  
  書込み, 6-25  
  成功要件の属性, 6-32  
  編集, 6-33  
ポリシー・ファイルのダンプ, 6-32  
ポリシー・ファイルの編集, 6-33

## ま

---

メッセージ整合の設定 (Oracle Business Activity Monitoring), 5-28

## ら

---

リバース・プロキシ・サーバー (OracleAS Web Cache), 3-13  
ルーティング・アルゴリズム  
  mod\_oc4j 用, 3-10  
レプリケーション  
  peer-to-peer, 3-16  
  peer-to-peer (静的), 3-16  
  peer-to-peer (動的), 3-16  
  データベース, 3-18  
  ノード数の指定, 3-19  
  マルチキャスト, 3-15  
  レプリケーション・ポリシーの設定, 3-18  
ロード・バランサ, 2-5  
  LVS, 2-6  
  OracleAS Disaster Recovery, 6-6  
  Windows ネットワーク・ロード・バランサ, 2-6  
  仮想サーバーの構成, 2-6  
  ソフトウェア・ロード・バランサ, 2-5

タイプ, 2-5  
ネットワーク・ロード・バランサ, 2-6  
ハードウェア・ロード・バランサ, 2-5  
フォルト・トレラント・モード, 2-6  
プロセス障害の検出, 2-6  
ポートの監視, 2-6  
ポートの構成, 2-6  
要件, 2-6  
  リソースの監視, 2-6  
ロード・バランサでの仮想サーバーの構成, 2-6  
ロード・バランサでのプロセス障害の検出, 2-6  
ロード・バランサでのポートの監視, 2-6  
ロード・バランサでのポートの構成, 2-6  
ロード・バランサでのリソースの監視, 2-6  
ロード・バランサの要件, 2-6  
ロード・バランシング・アルゴリズム  
  mod\_oc4j, 3-10, 3-21  
ログの適用サービス, 8-2