

Oracle Fail Safe for Windows NT and Windows 2000

概要および管理ガイド

リリース 3.1.2

2001 年 5 月

部品番号 : J03698-01

Oracle Fail Safe for Windows NT and Windows 2000 概要および管理ガイド, リリース 3.1.2

部品番号 : J03698-01

原本名 : Oracle Fail Safe Concepts and Administration Guide, Release 3.1.2 for Windows NT and Windows 2000

原本部品番号 : A87498-01

Copyright © 1996, 2001, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記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に関する法律により保護されています。

当プログラムのリバース・エンジニアリング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ことがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、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を用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	xvii
------------	------

第 I 部 概要

1 Oracle Fail Safe の基礎知識

1.1	Oracle Fail Safe とは	1-1
1.2	Oracle Fail Safe の利点	1-2
1.2.1	高い可用性を備えたリソースおよびアプリケーション	1-2
1.2.2	使いやすさ	1-4
1.2.3	アプリケーションとの統合のしやすさ	1-6
1.3	典型的な Oracle Fail Safe の構成	1-7
1.4	Oracle Fail Safe ソリューションの展開	1-9

2 クラスタの概念

2.1	クラスタ・テクノロジー	2-2
2.1.1	クラスタによる高い可用性の実現方法	2-3
2.1.2	システム・レベルの構成	2-3
2.1.3	ディスク・レベルの構成	2-4
2.1.4	クォラム・リソース	2-5
2.2	リソース、グループおよび高可用性	2-5
2.2.1	リソース	2-6
2.2.2	グループ	2-6
2.2.3	リソースの依存性	2-7
2.2.4	リソース・タイプ	2-8

2.3	グループ、仮想アドレスおよび仮想サーバー	2-10
2.4	仮想アドレスへの IP アドレス割当て	2-13
2.5	クラスタ・グループとクラスタ別名	2-13
2.6	フェイルオーバー	2-14
2.6.1	計画外フェイルオーバー	2-15
2.6.1.1	リソース障害による計画外フェイルオーバー	2-15
2.6.1.2	ノードの障害または使用不能状態による計画外フェイルオーバー	2-17
2.6.1.3	計画的フェイルオーバー	2-18
2.6.2	フェイルオーバーに影響するグループおよびリソースのポリシー	2-19
2.6.3	リソース障害の検出方法	2-21
2.6.4	リソース再起動ポリシー	2-22
2.6.5	リソースのフェイルオーバー・ポリシー	2-22
2.6.6	リソースの「可能所有者ノード」リスト	2-22
2.6.7	グループのフェイルオーバー・ポリシー	2-25
2.6.8	リソース再起動ポリシーおよびグループ・フェイルオーバー・ポリシーによる フェイルオーバー機能への影響	2-27
2.6.9	グループ・フェイルオーバーと「優先ノード」リスト	2-28
2.6.10	グループのフェイルオーバー・ノードの決定	2-28
2.7	フェイルバック	2-30
2.7.1	グループ・フェイルバックと「優先ノード」リスト	2-31
2.7.2	フェイルオーバー後のクライアントの再接続	2-33

3 Oracle Fail Safe ソリューションの設計

3.1	構成のカスタマイズ	3-1
3.1.1	アクティブ / パッシブ構成	3-2
3.1.2	アクティブ / アクティブ構成	3-3
3.1.3	作業負荷のパーティション化構成	3-5
3.1.4	階層化構成	3-6
3.2	クライアントとアプリケーションの統合	3-9

第 II 部 管理と総合管理

4 高い可用性を実現するための管理

4.1	フェイルオーバーを構成する意味	4-1
4.2	ウィザードの入力項目の Oracle Fail Safe での処理	4-2

4.3	クラスタ・セキュリティの管理	4-5
4.3.1	Oracle Fail Safe Server	4-6
4.3.1.1	Oracle Fail Safe Security Setup ツールを使用したアカウント更新	4-6
4.3.2	Oracle Fail Safe Manager	4-7
4.4	スタンドアロン・リソースの検出	4-8
4.5	リソース名の変更	4-8
4.6	複数の Oracle ホーム環境での Oracle Fail Safe の使用方法	4-8
4.7	複数仮想アドレスを使用する構成	4-9
4.8	既存クラスタへのノードの追加	4-10

5 FSCMD コマンドライン・インタフェース

6 トラブルシューティング・ツール

6.1	検証操作	6-1
6.1.1	クラスタの検証	6-3
6.1.2	グループの検証	6-5
6.1.3	スタンドアロン・データベースの検証	6-7
6.2	クラスタのダンプ	6-10
6.3	その他のトラブルシューティング情報の検索	6-12

第 III 部 Oracle Fail Safe ソリューションの展開

7 可用性を高めるためのデータベースの構成

7.1	スタンドアロン・データベースの検出	7-2
7.2	Oracle データベースのグループへの追加	7-3
7.2.1	構成前に	7-3
7.2.2	構成手順	7-4
7.2.3	Oracle データベースの構成データ	7-4
7.2.3.1	ノードの選択	7-5
7.2.3.2	仮想アドレス	7-6
7.2.3.3	データベース識別情報	7-8
7.2.3.4	データベース認証	7-9
7.3	Oracle Net8 リスナー・リソースの作成および構成	7-10
7.3.1	Net8 リスナーの定義	7-11
7.4	可用性の高いデータベースへのクライアント接続	7-12
7.4.1	データベースをグループに追加すると更新される Net8 構成	7-12

7.4.1.1	Oracle Fail Safe により行われる TNSNAMES.ORA ファイルの更新内容	7-12
7.4.1.2	Oracle Fail Safe により行われる LISTENER.ORA ファイルの更新内容	7-14
7.4.2	Database Configuration Assistant を使用して作成した Net8 構成の作成および更新	7-14
7.4.3	Oracle8i Database サーバー・ソフトウェアへのアップグレードに伴って必要な SID リスト・エントリ (SID_DESC)	7-15
7.4.4	複数のリスナーを伴うノードでの Net8 (または SQL*Net) の構成	7-16
7.4.5	高い可用性を提供するように構成されたデータベースでの外部プロシージャの使用	7-17
7.4.6	マルチスレッド・サーバーを使用するデータベースのサポート	7-18
7.4.6.1	Oracle8 データベース用のマルチスレッド・サーバー	7-18
7.4.6.2	Oracle8i データベース用のマルチスレッド・サーバー	7-20
7.4.6.2.1	スタンドアロンの Oracle8i データベース	7-20
7.4.6.2.2	グループ内の Oracle8i データベース	7-21
7.4.6.3	マルチスレッド・サーバー構成でのデータベースのオフライン化	7-22
7.4.6.4	マルチスレッド・サーバー構成でのグループからのデータベースの削除	7-22
7.5	データベースのセキュリティ要件	7-22
7.5.1	クラスタ・ノード上のパスワード・ファイルの同期化	7-23
7.5.2	データベースの INTERNAL パスワードの変更	7-24
7.5.3	Oracle Data Migration Assistant を使用したフェイルセーフ・データベースの移行	7-25
7.6	可用性を高めるための ctxsrv サーバーの構成	7-26
7.7	Oracle Enterprise Manager との統合	7-27
7.7.1	Oracle Intelligent Agent のグループへの追加	7-28
7.8	データベース・リカバリの最適化	7-30
7.9	フェイルセーフ・データベースに対する管理作業の実行	7-31
7.10	クライアントおよびクラスタ対応のアプリケーション	7-32
7.10.1	アプリケーションに対して透過的なフェイルオーバーの実現	7-32
7.10.1.1	クラスタ対応のアプリケーション	7-32
7.10.1.2	クラスタ対応ではないアプリケーション	7-33
7.10.1.3	クラスタ対応の ODBC および OCI アプリケーション	7-33
7.10.2	Oracle7 クライアント・アプリケーション	7-33
7.10.3	Oracle8 および Oracle8i クライアント・アプリケーション	7-34
7.10.3.1	透過的アプリケーション・フェイルオーバーのためのアプリケーション再接続	7-36
7.10.3.2	TNSNAMES.ORA ファイルでの再接続の構成	7-38
7.10.3.2.1	クライアント接続を使用可能にするには	7-38
7.10.3.2.2	フェイルオーバーの構成	7-39
7.10.3.3	Oracle8 および Oracle8i の ODBC クライアント・アプリケーションの再接続	7-40
7.10.3.4	Oracle8 OCI クライアント・アプリケーションの再接続	7-41
7.10.3.4.1	OCI プログラムの構造	7-42

7.10.3.4.2	透過的アプリケーション・フェイルオーバーのコールバック関数	7-43
7.10.3.4.3	フェイルオーバー・コールバック関数の作成	7-43
7.10.3.4.4	フェイルオーバー・コールバック関数の登録	7-44
7.10.3.5	透過的アプリケーション・フェイルオーバーのデモ・アプリケーション	7-44
7.11	可用性の高い Oracle データベースに関連するエラー処理と問題のトラブル シューティング	7-44
7.11.1	データベースをオンラインにする際に発生するエラーの処理	7-45
7.11.2	問題のトラブルシューティング	7-46
7.11.3	データベースのグループへの追加の問題	7-47
7.11.4	グループのオンライン化の問題	7-48
7.11.5	データベース管理者の認証（Internal としての接続）	7-49
7.11.6	サンプル・データベースの問題	7-50
7.11.7	仮想サーバーの構成で検出される問題	7-51
7.11.7.1	仮想アドレスの構成で発生する問題	7-51
7.11.7.2	リスナー作成時の問題	7-53
7.11.7.3	アーカイブされた LISTENER.ORA または TNSNAMES.ORA ファイル	7-53
7.11.7.4	ロールバック・ファイル	7-53
7.11.8	セキュリティ・アクセスおよび認証の問題	7-54
7.11.9	クライアントがデータベースにアクセスできない	7-54

8 可用性を高めるための Oracle WebDB リスナーの構成

8.1	概要	8-2
8.2	スタンドアロン Oracle WebDB リスナーの検出	8-4
8.3	Oracle WebDB リスナーのグループへの追加	8-4
8.3.1	構成前に	8-4
8.3.2	Oracle WebDB リスナーの構成手順	8-4
8.3.3	Oracle WebDB リスナーの構成データ	8-6
8.3.3.1	ノードの選択	8-6
8.3.3.2	Oracle WebDB リスナーの仮想アドレス	8-7
8.3.3.3	Oracle WebDB リスナーの識別情報	8-8
8.3.3.4	Oracle WebDB リスナーのデータベース	8-10
8.3.3.5	Oracle WebDB リスナーのアカウント	8-12
8.4	Oracle WebDB リスナーへのクライアント接続	8-13
8.5	Oracle WebDB リスナーまたは PL/SQL 設定の変更	8-13
8.6	Oracle WebDB リスナーのセキュリティ要件	8-13
8.7	Oracle WebDB のインストールに関するヒント	8-13
8.8	可用性の高い Oracle WebDB リスナーに関連する問題のトラブルシューティング	8-15

9 可用性を高めるための Oracle Forms Load Balancer Server の構成

9.1	概要	9-2
9.2	スタンドアロン Oracle Forms Load Balancer Server の検出	9-5
9.3	Oracle Forms Load Balancer Server のグループへの追加	9-5
9.3.1	構成前に	9-6
9.3.2	構成手順	9-6
9.3.3	Oracle Forms Load Balancer Server の構成データ	9-7
9.3.3.1	ノードの選択	9-8
9.3.3.2	Oracle Forms Load Balancer Server リスナーの仮想アドレス	9-10
9.3.3.3	Oracle Forms Load Balancer Server の識別情報	9-11
9.4	Oracle Forms Load Balancer Server のセキュリティ要件	9-13
9.5	起動パラメータの変更	9-13
9.5.1	Forms CGI-bin 実行可能ファイルの起動パラメータの変更	9-14
9.5.2	Oracle Forms Load Balancer Client の起動パラメータの変更	9-14
9.5.3	Oracle Forms Server の起動パラメータの変更	9-14
9.5.4	Oracle Forms Load Balancer Server の起動パラメータの変更	9-15
9.6	可用性の高い Oracle Forms Load Balancer Server に関連する問題のトラブルシューティング	9-15

10 可用性を高めるための Oracle Forms Server の構成

10.1	概要	10-2
10.2	スタンドアロン Oracle Forms Server の検出	10-3
10.3	Oracle Forms Server のグループへの追加	10-4
10.3.1	構成前に	10-4
10.3.2	構成手順	10-5
10.3.3	Oracle Forms Server の構成データ	10-6
10.3.3.1	ノードの選択	10-7
10.3.3.2	Forms Server の仮想アドレス	10-8
10.3.3.3	Forms Server の識別情報	10-9
10.3.3.4	Forms のパラメータ	10-10
10.3.3.5	Oracle Forms アプリケーション・ファイルのパス	10-12
10.3.3.6	Oracle Forms アプリケーションによりアクセスされる Oracle データベース	10-13
10.4	Oracle Forms Server のセキュリティ要件	10-15
10.5	Oracle Forms Server へのクライアント接続	10-16
10.6	可用性の高い Oracle Forms Server に関連する問題のトラブルシューティング	10-16
10.6.1	ユーザーが Oracle Forms Server にアクセスできない	10-17

10.6.2	Web ブラウザで Java アプレットをダウンロードできない	10-17
10.6.3	Java アプレットを実行できない	10-17
10.6.4	Web ブラウザでフェイルオーバー後に Java アプレットを再ロードできない	10-18
10.6.5	アプリケーションで Forms アプリケーションが見つからない	10-18
10.6.6	Oracle Forms アプリケーションが正しく実行されない	10-19
10.6.7	Web ブラウザでマルチフォーム・アプリケーションの追加ページにアクセス できない	10-19

11 可用性を高めるための Oracle Reports Server の構成

11.1	概要および実装のオプション	11-2
11.1.1	Web ベースの実装	11-2
11.1.2	クライアント / サーバー実装	11-3
11.1.3	マスター / スレーブ実装	11-5
11.2	スタンドアロン Oracle Reports Server の検出	11-5
11.3	Oracle Reports Server のグループへの追加	11-6
11.3.1	構成前に	11-6
11.3.2	構成手順	11-6
11.3.3	Oracle Reports Server の構成データ	11-8
11.3.3.1	ノードの選択	11-8
11.3.3.2	仮想アドレス	11-10
11.3.3.3	Oracle Reports Server の識別情報	11-12
11.3.3.4	ソース・ディレクトリ、キャッシュ・ディレクトリおよびジョブ・ ディレクトリの場所	11-13
11.3.3.5	Oracle Reports Server によりアクセスされるデータベース	11-14
11.3.3.6	Oracle Reports Server の実行に使用されるアカウント	11-15
11.4	Oracle Reports Server へのクライアント接続	11-16
11.4.1	Oracle Reports Server へのクライアント・アクセスのための TNSNAMES.ORA ファイルの更新	11-18
11.5	マスター / スレーブ実装のための構成後の手順	11-19
11.5.1	マスター / スレーブ例の概要	11-19
11.5.2	マスター / スレーブ間の通信を使用可能にする方法	11-20
11.5.2.1	マスター・サーバーのための調整	11-20
11.5.2.2	スレーブ・サーバーのための調整	11-21
11.5.3	既存のマスター / スレーブ実装へのスレーブ・サーバーの追加	11-22
11.5.3.1	マスター Oracle Reports Server のための調整	11-22
11.5.3.2	スレーブ・サーバーのための調整	11-23

11.6	可用性の高い Oracle Reports Server へのユーザー・アクセスの制御	11-24
11.7	Oracle Reports のスケジューリング	11-25
11.7.1	ジョブ・リポジトリを可用性の高い Oracle Reports Server とともに使用	11-25
11.8	可用性の高い Oracle Reports Server に関連する問題のトラブルシューティング	11-26
11.8.1	Web サーバーで CGI の RWCGI60.EXE ファイルを見つけられない	11-26
11.8.2	Oracle Reports Server で RWCGI60.EXE ファイルをダウンロード	11-27
11.8.3	ユーザーが Oracle Reports Server と通信できない	11-27
11.8.4	Web ブラウザで Oracle Reports 定義ファイルが見つからない	11-27
11.8.5	Oracle Reports アプリケーションが正しく実行されない	11-27
11.8.6	ユーザーが Reports を出力できない	11-28

12 可用性を高めるための Web サーバーの構成

12.1	概要	12-2
12.2	スタンドアロン Oracle Applilcation Server Web リスナーの検出	12-4
12.3	Oracle Application Server Web リスナーのグループへの追加	12-5
12.3.1	構成前に	12-6
12.3.2	Oracle Application Server の構成手順	12-7
12.3.3	Oracle Application Server の構成データ	12-8
12.3.3.1	ノードの選択	12-9
12.3.3.2	Oracle Application Server Web リスナーの識別情報	12-11
12.3.3.3	Oracle Application Server Web リスナーのデフォルト・ファイル	12-12
12.3.3.4	Oracle Application Server Web リスナーの仮想アドレス	12-14
12.4	Oracle Application Server Web リスナーへのクライアント接続	12-15
12.5	Oracle Application Server Web リスナーのグループからの削除	12-16
12.6	Oracle Application Server のセキュリティ要件	12-16
12.7	Oracle Application Server のインストールに関するヒント	12-17
12.7.1	インストールの一般的なヒント	12-17
12.7.2	インストールされる Oracle Application Server Web リスナー	12-17
12.7.2.1	ノードの IP アドレスのみをリスニングするようにデフォルト・リスナーを再構成	12-18
12.7.2.2	デフォルト Web リスナーの停止	12-19
12.8	可用性の高い Oracle Application Server Web リスナーに関連する問題のトラブルシューティング	12-19
12.9	Oracle HTTP Server の概要	12-19
12.10	スタンドアロン Oracle HTTP Server の検出	12-20
12.11	Oracle HTTP Server のグループへの追加	12-20
12.11.1	構成前に	12-20

12.11.2	Oracle HTTP Server の構成手順	12-21
12.11.3	Oracle HTTP Server の構成データ	12-22
12.11.3.1	ノードの選択	12-22
12.11.3.2	Oracle HTTP Server の識別情報	12-24
12.11.3.3	Oracle HTTP Server のディレクトリ	12-26
12.11.3.4	Oracle HTTP Server の仮想アドレス	12-27
12.12	Oracle HTTP Server へのクライアント接続	12-29
12.13	Oracle HTTP Server のグループからの削除	12-29
12.14	Oracle HTTP Server のセキュリティ要件	12-30
12.15	可用性の高い Oracle HTTP Server に関連する問題のトラブルシューティング	12-30
12.15.1	Oracle HTTP Server のグループへの追加の問題	12-31
12.15.2	ユーザーが Oracle HTTP Server にアクセスできない	12-31
12.15.3	ユーザーが Oracle HTTP Server Web サイトに接続できない	12-32
12.15.4	ユーザーが Web サイト上のドキュメントにアクセスできない	12-32

13 可用性を高めるための Oracle Applications Concurrent Manager の構成

13.1	概要	13-2
13.2	スタンドアロン Oracle Applications Concurrent Manager の検出	13-4
13.3	Oracle Applications Concurrent Manager のグループへの追加	13-4
13.3.1	構成前に	13-4
13.3.2	構成手順	13-5
13.3.3	Oracle Applications Concurrent Manager の構成データ	13-6
13.3.3.1	ノードの選択	13-7
13.3.3.2	Oracle Applications Concurrent Manager 仮想アドレス	13-9
13.3.3.3	Concurrent Manager の識別情報	13-10
13.3.3.4	Concurrent Manager の認証	13-12
13.3.3.5	Concurrent Manager のデータベース	13-13
13.4	Oracle Applications Concurrent Manager のセキュリティ要件	13-14
13.5	可用性の高い Oracle Applications Concurrent Manager に関連する問題のトラブル シューティング	13-15
13.5.1	Oracle Applications Concurrent Manager のグループへの追加の問題	13-16
13.5.2	グループのオンライン化の問題	13-16
13.5.3	Concurrent Manager 用の Windows サービスが誤って削除された	13-17
13.5.4	Concurrent Manager がオンライン化できない	13-17

14 可用性を高めるための Oracle MTS Service の構成

14.1	概要	14-2
14.2	スタンドアロン Oracle MTS Service の検出	14-4
14.3	Oracle MTS Service のグループへの追加	14-4
14.3.1	構成前に	14-4
14.3.2	構成手順	14-5
14.3.3	Oracle MTS Service の構成データ	14-7
14.3.3.1	ノードの選択	14-8
14.3.3.2	Oracle MTS Service 仮想アドレス	14-10
14.3.3.3	Oracle MTS Service 識別情報	14-11
14.3.3.4	Oracle MTS Service データベース	14-12
14.3.3.5	Oracle MTS Service 認証	14-13
14.4	Oracle MTS Service のセキュリティ要件	14-15
14.5	可用性の高い Oracle MTS Service に関連する問題のトラブルシューティング	14-15
14.5.1	Oracle MTS Service のグループへの追加の問題	14-16
14.5.2	グループのオンライン化の問題	14-16
14.5.3	削除された Oracle MTS Service の Windows サービス	14-17

15 可用性を高めるための汎用サービスの構成

15.1	概要	15-2
15.1.1	Oracle Fail Safe を使用する利点	15-2
15.1.2	可用性を高める構成にしない汎用リソース	15-3
15.2	スタンドアロン汎用サービスの検出	15-3
15.3	汎用サービスのグループへの追加	15-3
15.3.1	構成手順	15-3
15.3.2	汎用サービス用構成データ	15-5
15.3.2.1	ノードの選択	15-5
15.3.2.2	汎用サービスの識別情報	15-7
15.3.2.3	汎用サービスの起動パラメータ	15-9
15.3.2.4	汎用サービスにより使用されるディスク	15-11
15.3.2.5	汎用サービスの依存性	15-13
15.3.2.5.1	汎用サービスの依存性の指定	15-13
15.3.2.5.2	汎用サービスと仮想アドレスの依存性	15-14
15.3.2.6	汎用サービスのレジストリ・キー	15-15
15.4	汎用サービスのセキュリティ要件	15-16
15.5	サンプル汎用サービスの構成	15-16

15.6	汎用サービスに関する問題のトラブルシューティング	15-17
------	--------------------------------	-------

第 IV 部 付録、用語集および索引

A ODBC および OCI のデモ

B ネットワーク構成の要件

B.1	ホスト名および IP アドレスの登録	B-1
B.2	クラスタ内の正しい名前解決の有効性検査	B-2
B.3	不適切な名前解決に関する問題のトラブルシューティング	B-3
B.3.1	Windows NT Service Pack 3 に推奨されるソリューション	B-3
B.3.2	Windows NT Service Pack 4 以降に推奨されるソリューション	B-5
B.3.3	Windows 2000 に推奨されるソリューション	B-5

C オラクル社カスタマ・サポート・センターへの連絡

C.1	問題の報告	C-1
C.2	バージョン情報の検出	C-2
C.3	Oracle Fail Safe の問題のトレース	C-2
C.4	トレース・ファイルとアラート・ファイルの検索	C-4

用語集

索引

例リスト

7-1 コールバックを登録する OCI の構造 7-44

図リスト

1-1	Microsoft クラスタでの Oracle Fail Safe によるフェイルオーバー	1-3
1-2	Oracle Fail Safe Manager	1-5
1-3	Oracle Fail Safe Manager のメニューと内容	1-6
1-4	Oracle Fail Safe によって構成されるハードウェアおよびソフトウェア・コンポーネント	1-9
2-1	Microsoft クラスタ・システム	2-2
2-2	シェアード・ナッシング構成	2-4
2-3	クラスタ・グループの設計	2-7
2-4	可用性の高い Oracle Reports Server のリソース・タイプ	2-9
2-5	「リソースをグループに追加 - 仮想アドレス」ウィザード・ページ	2-11
2-6	仮想サーバーを介したクラスタ・リソースへのアクセス	2-12
2-7	「クラスタに接続」ダイアログ・ボックスの「クラスタ別名」	2-14
2-8	リソースのフェイルオーバー	2-16
2-9	ノードのフェイルオーバー	2-18
2-10	グループの「フェイルオーバー」タブ・ページ	2-20
2-11	リソースの「ポリシー」タブ・ページ	2-21
2-12	「ノード」タブ・ページ	2-25
2-13	フェイルオーバーしきい値とフェイルオーバー期間の関係を示す時間軸	2-26
2-14	グループの「フェイルバック」タブ・ページ	2-31
3-1	2 ノードのアクティブ / パッシブ (スタンバイ) 構成	3-2
3-2	4 ノードのアクティブ / パッシブ (スタンバイ) 構成	3-3
3-3	アクティブ / アクティブ構成	3-4
3-4	アクティブ / アクティブの作業負荷のパーティション化構成	3-6
3-5	Oracle Reports Server の階層化構成	3-8
4-1	Oracle Fail Safe 環境内の仮想サーバーとアドレッシング	4-4
4-2	Oracle Fail Safe Server の Windows ユーザー・アカウント設定	4-7
4-3	スタンドアロン・リソースの検出	4-8
6-1	「トラブルシューティング」メニューの検証コマンド	6-2
6-2	「クラスタの検証」のクラスタワイド操作ウィンドウ	6-4
6-3	「グループの検証」のクラスタワイド操作ウィンドウ	6-6
6-4	「スタンドアロン・データベースの検証」ダイアログ・ボックス	6-8
6-5	「スタンドアロン・データベースの検証」のクラスタワイド操作ウィンドウ	6-9
6-6	グループを検証するクラスタワイド操作ウィンドウ	6-11
7-1	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	7-5
7-2	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	7-6
7-3	「データベース仮想アドレス」ウィザード・ページ	7-7
7-4	「データベースの識別情報」ウィザード・ページ	7-9
7-5	「データベースの認証」ウィザード・ページ	7-10
7-6	「リソースをグループに追加 - リソース」ウィザード・ページ	7-29
7-7	フェイルオーバーのフロー・チャート	7-36
7-8	ODBC クライアントの自動フェイルオーバーを使用可能にする	7-41
7-9	OCI ハンドルの階層	7-42
8-1	可用性の高い Oracle WebDB 構成のための 2 層アーキテクチャ	8-2
8-2	可用性の高い Oracle WebDB 構成のための 3 層アーキテクチャ	8-3

8-3	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	8-7
8-4	Oracle WebDB リスナーの「仮想アドレスの選択」ウィザード・ページ	8-8
8-5	「Oracle WebDB リスナーの識別情報」ウィザード・ページ	8-10
8-6	「Oracle WebDB リスナーのデータベース」ウィザード・ページ	8-11
8-7	「Oracle WebDB リスナーのアカウント」ウィザード・ページ	8-12
9-1	Oracle Forms Load Balancer を使用した Oracle Forms の 3 層アーキテクチャ	9-4
9-2	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	9-9
9-3	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	9-10
9-4	「仮想アドレスの選択」ウィザード・ページ	9-11
9-5	「Load Balancer Server の識別情報」ウィザード・ページ	9-13
10-1	Oracle Forms の 3 層アーキテクチャ	10-3
10-2	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	10-7
10-3	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	10-8
10-4	Oracle Forms の「仮想アドレスの選択」ウィザード・ページ	10-9
10-5	「Forms の識別情報」ウィザード・ページ	10-10
10-6	「Forms のパラメータ」ウィザード・ページ	10-12
10-7	「Forms パス」ウィザード・ページ	10-13
10-8	「Forms のデータベース」ウィザード・ページ	10-14
10-9	Web ベースの Forms を示すツリー・ビュー	10-15
11-1	Oracle Reports の Web 実装のための 3 層アーキテクチャ	11-3
11-2	Oracle Reports のクライアント / サーバー実装のための 3 層アーキテクチャ	11-4
11-3	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	11-9
11-4	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	11-10
11-5	Oracle Reports の「仮想アドレスの選択」ウィザード・ページ	11-11
11-6	「Reports の識別情報」ウィザード・ページ	11-12
11-7	「Reports のディスク」ウィザード・ページ	11-14
11-8	「Reports のデータベース」ウィザード・ページ	11-15
11-9	「Reports のアカウント」ウィザード・ページ	11-16
12-1	可用性の高い Web 構成のための 3 層アーキテクチャ	12-3
12-2	スタンドアロン Oracle Application Server Web リスナー	12-5
12-3	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	12-10
12-4	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	12-11
12-5	「OAS Web リスナーの識別情報」ウィザード・ページ	12-12
12-6	「OAS Web リスナーのデフォルト・ファイル」ウィザード・ページ	12-14
12-7	「OAS Web リスナーの仮想アドレス」ウィザード・ページ	12-15
12-8	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	12-23
12-9	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	12-24
12-10	「Oracle HTTP Server の識別情報」ウィザード・ページ	12-25
12-11	「Oracle HTTP Server のディレクトリ」ウィザード・ページ	12-27
12-12	「Oracle HTTP Server の仮想アドレス」ウィザード・ページ	12-28
13-1	可用性の高い Oracle Applications Concurrent Manager 構成のための 3 層アーキテクチャ	13-3
13-2	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	13-8
13-3	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	13-9

13-4	Oracle Applications Concurrent Manager の「仮想アドレスの選択」ウィザード・ページ	13-10
13-5	「Concurrent Manager の識別情報」ウィザード・ページ	13-11
13-6	「Concurrent Manager の認証」ウィザード・ページ	13-12
13-7	「Concurrent Manager のデータベース」ウィザード・ページ	13-14
14-1	可用性の高い Oracle MTS Server のための 3 層アーキテクチャ	14-3
14-2	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	14-9
14-3	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	14-10
14-4	Oracle MTS Service 「仮想アドレス」ウィザード・ページ	14-11
14-5	「Oracle MTS Service の識別情報」ウィザード・ページ	14-12
14-6	「Oracle MTS Service のデータベース」ウィザード・ページ	14-13
14-7	「Oracle MTS Service の認証」ウィザード・ページ	14-14
15-1	全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ	15-6
15-2	使用できないノードがある場合の「ノードの選択」ウィザード・ページ	15-7
15-3	「汎用サービスの識別情報」ウィザード・ページ	15-9
15-4	「汎用サービスのアカウント」ウィザード・ページ	15-11
15-5	「汎用サービスのディスク」ウィザード・ページ	15-12
15-6	依存性ツリー	15-13
15-7	「汎用サービスの依存性」ウィザード・ページ	15-14
15-8	「汎用サービスのレジストリ」ウィザード・ページ	15-15

表リスト

2-1	Test_Group グループ内のリソースの可能所有者の例	2-29
4-1	許可と権限	4-5
6-1	トラブルシューティングのための検証コマンド	6-2
7-1	データベースを構成する手順	7-4
7-2	FAILOVER_MODE パラメータの TYPE サブパラメータ・オプション	7-39
8-1	Oracle WebDB リスナーの構成手順	8-5
9-1	Oracle Forms Load Balancer Server の構成手順	9-6
10-1	Oracle Forms Server の構成手順	10-5
11-1	Oracle Reports Server の構成手順	11-7
11-2	マスター / スレーブ実装の例	11-19
12-1	Oracle Application Server の構成手順	12-7
12-2	Oracle HTTP Server の構成手順	12-21
13-1	Oracle Application Concurrent Manager の構成手順	13-5
14-1	Oracle MTS Service を構成する手順	14-6
15-1	汎用サービスの構成手順	15-4
15-2	サンプル汎用サービスの構成手順	15-16
C-1	クラスタのサーバー・ノードが使用するトレース・フラグ	C-3

はじめに

このマニュアルでは、Microsoft クラスタ・システム上で稼働している Oracle Fail Safe を使用して、次のものの可用性が高まるように構成する方法を説明します。

- Oracle データベース・サーバー
- Oracle WebDB リスナー
- Oracle Forms Load Balancer Server
- Oracle Forms Server
- Oracle Reports Server
- Oracle Application Server
- Oracle HTTP Server
- Oracle Applications Concurrent Manager
- Oracle MTS Service
- Microsoft Windows サービスとしてインストールされたアプリケーション

対象読者

このマニュアルは、Microsoft クラスタ・システム上で実行しているソフトウェア・コンポーネントのダウン時間を最小にするために、Oracle Fail Safe を考慮している方を対象としています。

読者は、Microsoft Cluster Server (MSCS)、SQL*Net または Net8 ネットワーキング、および高い可用性を実現する他のアプリケーションに精通している必要があります。

構成

このマニュアルには、15 の章、3 つの付録、用語集および索引が含まれています。

第 1 章	クラスタおよび Oracle Fail Safe の概要を示します。
第 2 章	Microsoft クラスタおよび Oracle Fail Safe の概念と用語を紹介します。
第 3 章	高い可用性で業務上の要求に合致した、Oracle Fail Safe 構成のカスタマイズと最適化について説明します。
第 4 章	複数の Oracle ホームでの Oracle Fail Safe Manager の使用などについて、セキュリティ上の考慮点を説明します。
第 5 章	FSCMD コマンドの参照情報を掲載しています。
第 6 章	Oracle Fail Safe Manager ファミリのトラブルシューティング・ツールについて説明します。
第 7 章	Oracle データベース・サーバーに高い可用性を与える構成方法、および Oracle Call Interface (OCI) や ODBC クライアント、アプリケーションとの統合方法について説明します。
第 8 章	Oracle WebDB リスナーに高い可用性を与える構成方法を説明します。
第 9 章	Oracle Forms Load Balance Server に高い可用性を与える構成方法を説明します。
第 10 章	Oracle Forms Server に高い可用性を与える構成方法を説明します。
第 11 章	Oracle Reports Server に高い可用性を与える構成方法を説明します。
第 12 章	Oracle Application Server または Oracle HTTP Server に高い可用性を与える構成方法を説明します。
第 13 章	Oracle Applications Concurrent Manager に高い可用性を与える構成方法を説明します。
第 14 章	Oracle Service for Microsoft Transaction Server (Oracle MTS Service と呼ばれる) に高い可用性を与える構成方法を説明します。
第 15 章	Windows 汎用サービスに高い可用性を与える構成方法を説明します。
付録 A	Oracle Fail Safe CD-ROM に含まれているサンプルのデモ・ソフトウェアをダウンロードして使用する方法を示します。
付録 B	クラスタの適切なネットワーク構成を検証するための情報を掲載しています。
付録 C	オラクル社カスタマ・サポート・センターへの連絡方法、およびサポート担当者に提示する必要がある情報について説明します。

関連資料

Oracle Fail Safe の詳細は、次の資料を参照してください。

- ソフトウェアの更新、オンライン・マニュアルへのアクセス、その他のリリース固有の情報は、『Oracle Fail Safe リリース・ノート』を参照してください。
- インストール、削除およびアップグレードの方法は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
- オンライン・ヘルプは、Oracle Fail Safe Manager に提供されているオンライン・ヘルプの項目を参照してください。オンライン・ヘルプの項目にアクセスするには、Oracle Fail Safe Manager のメニュー・バーにある「ヘルプ」をクリックしてください。

関連製品に関する詳細は、次の資料を参照してください。

- クラスタ・システムの詳細は、Microsoft Cluster Server (MSCS) のマニュアルを参照してください。
- Oracle サーバーとその動作の仕組みの詳細は、『Oracle8i 概要』を参照してください。
- Net8 の詳細は、『Net8 管理者ガイド』を参照してください。
- ODBC の詳細は、『Microsoft ODBC Software Development Kit and Programmer's Reference』を参照してください。また、ODBC ドライバの Oracle 固有の実装の詳細は、Oracle ODBC オンライン・ヘルプを参照してください。(たとえば、Microsoft により定義されていないフェイルオーバーのセットアップ方法などを説明しています。)
- Oracle Call Interface (OCI) の詳細は、『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を参照してください。
- Oracle Enterprise Manager の詳細は、Oracle Enterprise Manager 製品のマニュアルを参照してください。
- Oracle WebDB リスナーの詳細は、Oracle WebDB のドキュメント・セットを参照してください。
- Oracle Forms Load Balancer Server、Oracle Forms および Oracle Reports の詳細は、Oracle Developer 6i のドキュメント・セットを参照してください。
- Oracle Application Server の詳細は、『Oracle Application Server 概要』を参照してください。
- Oracle HTTP Server の詳細は、Oracle HTTP Server のマニュアルを参照してください。
- Oracle MTS Service の詳細は、『Oracle8 と Microsoft Transaction Server の連携』を参照してください。
- Oracle Applications Concurrent Manager の詳細は、Oracle Applications 11i のドキュメント・セットを参照してください。

表記規則

このマニュアルでは、次の表記規則を使用しています。

表記規則	意味
.	例の中の縦向きの省略記号は、例に直接関係しない情報が省略されていることを示す。
...	文またはコマンドの中の横向きの省略記号は、例に直接関係しない文またはコマンドの一部が省略されていることを示す。
イタリック体	イタリック体は、変数を示すために使用する。変数とは、システム・メッセージ内（たとえば、内部エラー番号 <i>mmn</i> ）、コマンドライン内（たとえば、 <i>/Producer=name</i> ）、および本文中でのコマンド・パラメータ内（現在のホスト名として <i>cluster-node-name</i> を指定しているような場合）で変化する情報を持ちます。
太字体	本文中の太字体は、本文や用語集でその用語が定義されていることを示す。
< >	ユーザー指定の名前を囲む記号。
[]	オプション修飾子を囲む記号。この中から 1 つ選択するか、または選択せずに省略します。
大文字	必須のキーワードまたはパラメータを示す。
小文字	ユーザーが選択するキーワードまたはパラメータを示す。ただし、選択リストからのみ選択が可能です。
「メニュー」 → 「オプション」 → 「サブメニュー・オプション」	右矢印は、メニュー・オプションまたはサブメニュー・オプションを選ぶことを示す記号。次の例は、「表示」メニューのプルダウンを表示し、ポインタを移動して「イメージ」サブメニューのプルダウンを表示した後、「リフレッシュ」オプションを選択することを示します。 「表示」 → 「イメージ」 → 「リフレッシュ」を選択します。

第I部

概要

第I部では、汎用クラスタおよび Oracle Fail Safe の概念について説明します。

第I部は、次の章から構成されています。

- 第1章「Oracle Fail Safe の基礎知識」
- 第2章「クラスタの概念」
- 第3章「Oracle Fail Safe ソリューションの設計」

Oracle Fail Safe の基礎知識

ビジネスにおいて、24 時間 365 日利用できる製品やサービスがますます求められるようになっていきます。100% の可用性を保証できるソリューションはありませんが、Oracle Fail Safe により、Oracle データベースのみならず、Microsoft Cluster Server (MSCS) によって構成され、Microsoft クラスタ上で稼働する多くのアプリケーションのダウン時間を最小限に抑えることが可能です。

この章では、次の項目について説明します。

項目	参照
Oracle Fail Safe とは	1.1 項
Oracle Fail Safe の利点	1.2 項
典型的な Oracle Fail Safe の構成	1.3 項
Oracle Fail Safe ソリューションの展開	1.4 項

1.1 Oracle Fail Safe とは

Oracle Fail Safe は使いやすいソフトウェア・オプションで、Microsoft Cluster Server (MSCS) とともに、Microsoft クラスタ上で高い可用性を実現するビジネス・ソリューションです。**クラスタ**は、ネットワーク・ユーザーからは可用性の高い単一システムのように見える 2～4 つの Windows システムで構成されます。クラスタ内の各システムを**クラスタ・ノード**と呼びます。

Oracle Fail Safe は MSCS クラスタ・ソフトウェアとともに、クラスタ上で実行されるアプリケーションおよびデータベースの高い可用性を実現します。あるクラスタ・ノードに障害が発生した場合、Oracle Fail Safe を使用して構成したパラメータに基づき、クラスタ・ソフトウェアがその作業負荷を正常に機能しているノードに移します。この操作を**フェイルオーバー**と呼びます。

Oracle Fail Safe によって、Oracle データベース、Oracle WebDB リスナー、Oracle Forms Server、Oracle Reports Server、Oracle Applications、および Microsoft Windows サービスとして構成可能なほとんどすべてのアプリケーションのダウン時間を短縮できます。

Oracle Fail Safe は Oracle Fail Safe Server と Oracle Fail Safe Manager から構成されます。

- Oracle Fail Safe Server は MSCS ソフトウェアとともに、可用性を高めるように構成されたリソースが計画的に、また予想外に停止したときの高速自動フェイルオーバーを構成します。これらのリソースには、Oracle データベース・サーバー、Oracle Forms Server、Oracle Reports Server、Oracle MTS Service、Web サーバーまたはその他の Windows サービス（また、これらの項目が依存するソフトウェアおよびハードウェア）があります。また、Oracle Fail Safe Server は障害が発生したソフトウェア・リソースの再起動を試行できるため、あるクラスタ・ノードから別のノードへのフェイルオーバーが不要になることもあります。
- Oracle Fail Safe Manager には、クラスタ・リソースを構成し管理するための使いやすいインタフェースとウィザード、さらに問題を診断するためのトラブルシューティング・ツールが用意されています。

これらのコンポーネントをともに使用することで、高い可用性を備えたデータベース、アプリケーションおよびインターネット・ビジネス・ソリューションを速やかに展開できます。

1.2 Oracle Fail Safe の利点

Oracle Fail Safe には、主に次のような利点があります。

- 高い可用性を備えたデータベースおよびアプリケーション
- 使いやすさ
- アプリケーションとの統合のしやすさ

1.2.1 高い可用性を備えたリソースおよびアプリケーション

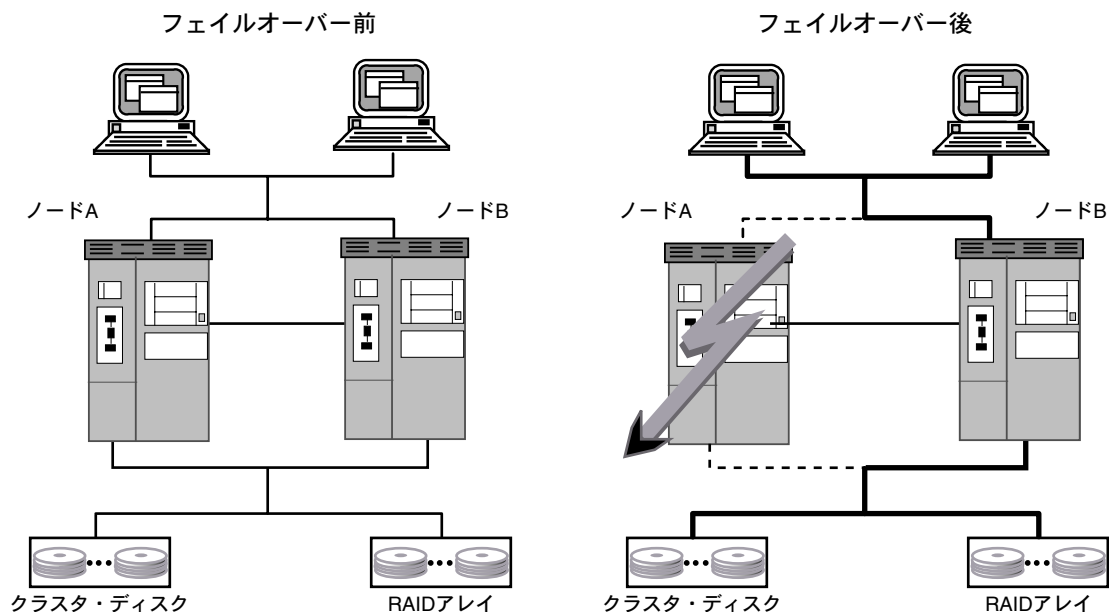
Oracle Fail Safe は MSCS とともに、ハードウェア・リソースとソフトウェア・リソースを両方の可用性が高くなるように構成します。構成後、エンド・ユーザーおよびクライアントからは、クラスタ内の複数のノードが単一の仮想サーバーのように見えます。エンド・ユーザーとクライアント・アプリケーションは単一の固定ネットワーク・アドレス（**仮想アドレス**）に接続し、その基礎となるクラスタに関する知識は必要ありません。その後、クラスタ内のあるノードが使用不能になった場合、MSCS は障害が発生したノードの作業負荷（およびクライアントの要求）を他のノードに移します。

例として、[図 1-1](#) の左側に、2つのノードから成るクラスタ構成を示します。ここでは両方のノードが使用可能であり、トランザクションが能動的に処理されています。表面的には、この構成は2つの独立したサーバーのセットアップと同じように見えますが、**共有記憶装置インターコネクト**によって2つのノードに1セットのディスクが物理的に接続されるよう、記憶装置サブシステムが構成されている点が異なります。同じディスクに物理的に2つの

ノードが接続されますが、MSCS により、各ディスクをある時点で所有しアクセスできるノードは必ずどちらか 1 つのノードとなります。

図 1-1 の右側に、あるノードのハードウェアまたはソフトウェアが使用不能になった場合に、管理者が介入することなく、正常に機能しているノードへ自動的に作業負荷を移して（フェイルオーバー）再起動する方法を示します。フェイルオーバー中、クラスタ・ディスクの所有権は障害が発生したサーバー（ノード A）から解放され、正常に機能しているサーバー（ノード B）がこの所有権を獲得します。Oracle データベース・インスタンスがノード A で稼働していた場合、Oracle Fail Safe によってノード B のデータベース・インスタンスが再起動されます。クライアントは、ノード A がホスト・サーバーであったときにデータベースへのアクセスに使用したのと同じ仮想アドレスを使用し、ノード B を介してデータベースにアクセスします。

図 1-1 Microsoft クラスタでの Oracle Fail Safe によるフェイルオーバー



1.2.2 使いやすさ

クラスタ内で動作するソフトウェアおよびその依存コンポーネント（たとえば、ディスク、IP アドレス、ネットワーク）の構成作業には、数多くのハードウェア・コンポーネントおよびソフトウェア・コンポーネントが関係しているため、複雑なプロセスになることがあります。そのため、Oracle Fail Safe はインストール、管理および使用が簡単にできるように設計されています。

インストール： Oracle Fail Safe は、Oracle Universal Installer を使用して対話的にインストールすることも、サイレント・モードでインストールすることもできます。**サイレント・モード**は、レスポンス・ファイルを使用して Oracle Universal Installer に入力データを提供することによりソフトウェアをインストールするインストール方法です。また、オペレーティング・システムとアプリケーション・ソフトウェアの両方を**ローリング・アップグレード**することも可能です。1つのシステムがローリング・アップグレードによってアップグレードされている間、もう1つのクラスタ・ノードが継続してクラスタの作業負荷のホストとなることが可能になり、ダウン時間が最小になります。詳細は、『Oracle Fail Safe インストール・ガイド』を参照してください。

管理と使用： Oracle Fail Safe Manager には、クラスタ上のアプリケーションとデータベースをセットアップ、構成および管理するための使いやすいインタフェースが用意されています。Oracle Fail Safe Manager には、構成手順を自動化し、その構成をクラスタ・ノード間で矛盾なく複製するウィザードもあります。

Oracle Fail Safe Manager には、次のものが含まれます。

- 情報を効率的に検索できるように同じデータの複数のビューが表示された、オブジェクトのツリー・ビュー
- リソース構成を自動化および簡易化するウィザードと、作業負荷のバランスをとるためにノード間でリソースを移動するなど、定期的なシステム・メンテナンスを速やかに実行するためのドラッグ・アンド・ドロップ機能
- 構成の前後に一般的な構成の問題を自動的に診断して修正する、統合検証ツール・ファミリ
- HTML や PDF 形式で使用可能なクイック・ツアー、チュートリアル、ヘルプおよびマニュアルなどのオンライン・マニュアル
- バッチ・プログラムやスクリプトからのクラスタ管理に便利なコマンドライン・インタフェース（FSCMD）

図 1-2 に、Oracle Fail Safe Manager のウィンドウを示します。左側のペインには、クラスター・リソースの複数のビュー（および現行の状態）が示されたツリーが表示されます。右側のペインには、クラスター上のすべてのグループとそれらの現行状態をリストするプロパティ・ページが表示されます。ツリー・ビューから選択したフォルダによって、右側のペインの表示が異なります。特定のグループまたはリソースを選択すると、そのグループまたはリソースのプロパティ・シートが表示されます。

図 1-2 Oracle Fail Safe Manager

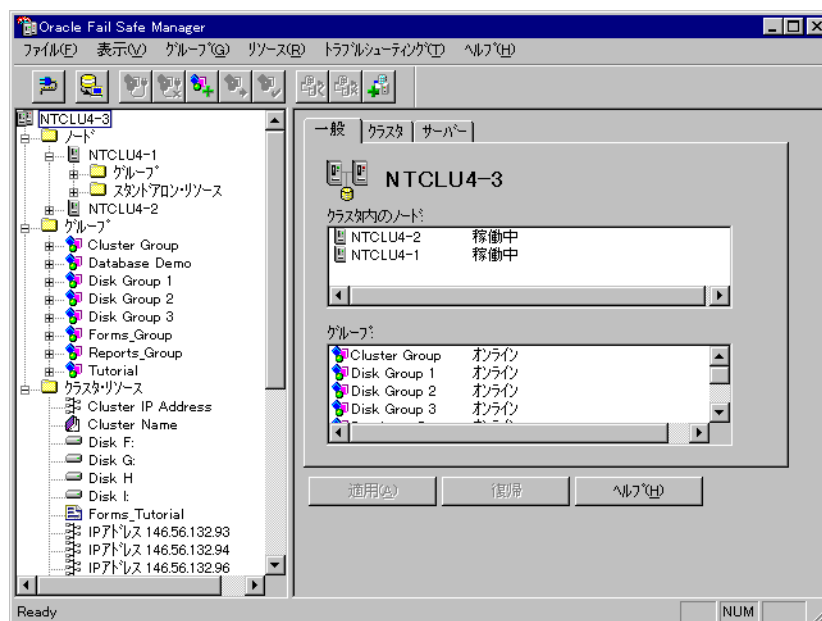


図 1-3 に、Oracle Fail Safe のメニューと各メニューの項目を示します。

図 1-3 Oracle Fail Safe Manager のメニューと内容



1.2.3 アプリケーションとの統合のしやすさ

Oracle Fail Safe によって構成された、データベースまたはその他のアプリケーションにアクセスするよう既存のアプリケーションを構成する場合、変更はほとんど必要ありません。アプリケーションは常に同じ仮想アドレスにあるクラスタ・リソースにアクセスするため、アプリケーションはフェイルオーバーを短時間のノード・リブートとして扱います。

フェイルオーバーの発生後、データベース・クライアントおよびユーザーは再接続して、まだ実行されていないトランザクション（インスタンスのリカバリ中にロールバックされたデータベース・トランザクションなど）があればそれを再実行する必要があります。

Oracle8 または Oracle8i OCI によって開発されたアプリケーション（Oracle ODBC ドライバを使用する ODBC クライアントを含む）では、フェイルオーバー後の自動再接続を利用できます。Oracle7 ODBC および OCI によって開発されたアプリケーションの場合は、接続と状態を再確立するための追加コーディングが必要です。詳細は、7.10 項を参照してください。

1.3 典型的な Oracle Fail Safe の構成

Microsoft ハードウェア互換性リストの要件を満たす Microsoft クラスタ・システムには、MSCS を使用して Oracle Fail Safe ソリューションを展開することができます。有効なクラスタ構成のリストは、次の URL で、各クラスタ・タイプ別に確認できます。

<http://www.microsoft.com/hcl/>

多くのクラスタの構成はほとんど同じで、記憶装置インターコネクトの選択（SCSI またはファイバ・チャネル）と、クラスタ・ノード間のアプリケーション配置方式のみが異なります。

典型的なクラスタ構成には、次のハードウェアおよびソフトウェアが含まれます。

- ハードウェア
 - 実行可能アプリケーション・ファイルがインストールされているローカル（プライベート）。ディスクをそれぞれ1つ以上持っている Microsoft クラスタ・ノード。
 - クラスタ内通信のためのノード間のプライベート（ハートビート）・インターコネクト。
 - ローカル・エリア・ネットワーク（LAN）またはワイド・エリア・ネットワーク（WAN）に接続するパブリック・インターコネクト（インターネットまたはイントラネット、あるいはその両方）。
 - **共有記憶装置インターコネクト**（SCSI またはファイバ・チャネル）上の NTFS 形式のディスク。あるノードから別のノードにフェイルオーバーする必要があるすべてのデータ・ファイル、ログ・ファイルおよびその他のファイルは、これらのクラスタ・ディスク上に置かれます。

注意： さらに高い可用性を与えるために冗長性の高いハードウェア（RAID など）を使用する場合の情報は、クラスタ・ハードウェアのマニュアルを参照してください。

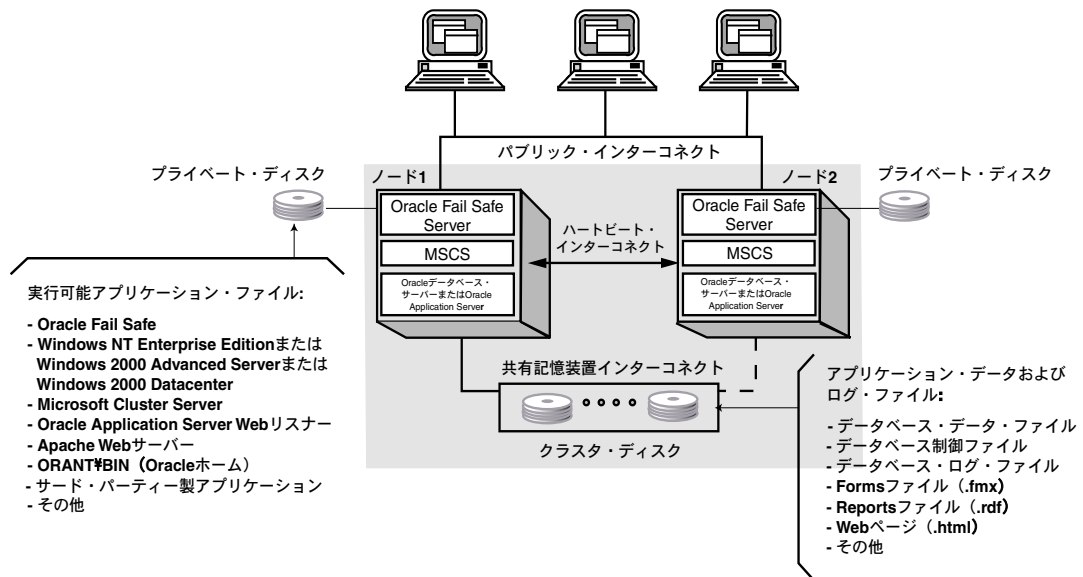
- 追加の冗長コンポーネント（UPS、ネットワーク・カード、ディスク・コントローラなど）。
- ソフトウェア（各ノードにインストール）
 - Windows NT Enterprise Edition、Windows 2000 Advanced Server または Windows 2000 Datacenter Server
 - Oracle Fail Safe Server
 - Oracle Fail Safe Manager（1 つ以上のクラスタ・ノードまたは 1 台以上のクライアント・ワークステーション、あるいはその両方にインストール）

- 次のリソースのうち、可用性を高める 1 つ以上のリソース
 - * Oracle データベース・サーバー
 - * Oracle WebDB リスナー
 - * Oracle Forms Load Balancer Server
 - * Oracle Forms Server
 - * Oracle Reports Server
 - * Oracle HTTP Server
 - * Oracle Applications Concurrent Manager
 - * Oracle MTS Service
 - * Windows サービスとして構成可能な Oracle またはサード・パーティー製のアプリケーション

これらのコンポーネントのリリース別のサポートに関する詳細は、リリース・ノートを参照してください。

図 1-4 に、2 つのノードから成るクラスタを Oracle Fail Safe で構成した場合のハードウェアおよびソフトウェア・コンポーネントを示します。実行可能アプリケーション・ファイルは各クラスタ・ノードのプライベート・ディスク上にインストールされ、アプリケーション・データおよびログ・ファイルは共有クラスタ・ディスクに常駐することに注意してください。

図 1-4 Oracle Fail Safe によって構成されるハードウェアおよびソフトウェア・コンポーネント



1.4 Oracle Fail Safe ソリューションの展開

Oracle Fail Safe は MSCS とともに、クラスタ上で稼働するリソースを構成し、高速フェイルオーバーを実現して、計画的な停止（システム・アップグレード）および計画外の停止（ハードウェアまたはソフトウェアの障害）によるダウン時間を最小限に抑えます。

クラスタは次の事項を管理するよう設計されており、高い可用性を提供します。

■ 計画外フェイルオーバー

クラスタにより、**計画外フェイルオーバー**（ハードウェアまたはソフトウェア・コンポーネントの障害）がユーザーに対して透過的な方法で管理されます。クラスタ上の1つのノードが使用不能になった場合、一時的に別のノードがそれ自体の作業負荷と、障害のあったノードの作業負荷の両方を処理します。リソースに障害が発生し、現在のノード上で再起動できなくなった場合、別のノードがそのリソース（およびそれが依存しているすべてのリソース）の所有権を受け取り、その再起動を試みます。

■ 計画的フェイルオーバー

クラスタにより、**計画的フェイルオーバー**（クラスタ上のソフトウェアをアップグレードする場合などに意図的に発生させるフェイルオーバー）が管理されます。別のノードにリソースをフェイルオーバーし、ソフトウェアまたはハードウェアのアップグレードを実行した後、そのリソースを元のノードに戻すことができます。（これをリソースの

フェイルバックと呼びます。) その後、クラスタの他のノードでも同じアップグレード手順を実行します。

Oracle Fail Safe では次の事項が管理され、クラスタ環境のリソースを効率的に使用できます。

- 独立した作業負荷

クラスタ・ノードは個別の作業負荷を処理できます。たとえば、1つのノードが Oracle データベースのホストとなり、残りのノードがアプリケーションのホストとなることが可能です。

- ロード・バランス

クラスタ・ノード間でリソースのバランスをとることができます。たとえば、負荷の大きいノードから容量に余裕があるノードにデータベースを移すことが可能です。

Oracle Fail Safe には多様な配置オプションがあり、幅広いフェイルオーバー要件を満たします。第3章では、アクティブ / パッシブ、アクティブ / アクティブ、作業負荷分散の各ソリューションなど、業務要件に合わせて Oracle Fail Safe ソリューションを構成する方法を説明します。

クラスタの概念

Oracle Fail Safe の高可用性ソリューションでは、Microsoft クラスタ・ハードウェアと Microsoft Cluster Server (MSCS) ソフトウェアを使用しています。

- **Microsoft クラスタ**は、2～4つの独立したコンピューティング・システム（ノードと呼ばれる）が同じディスク・サブシステムに接続された構成になっています。
- **Microsoft Cluster Server (MSCS)** は、Microsoft Windows NT Enterprise Edition、Microsoft Windows 2000 Advanced Server および Microsoft Windows 2000 Datacenter Server に組み込まれているソフトウェアです。Microsoft Cluster Server によって、Microsoft クラスタに配置されたアプリケーションおよびハードウェアのコンポーネント（リソースと呼ばれる）を構成、監視および制御できます。Microsoft Windows NT および Microsoft Windows 2000 Advanced Server では、1つのクラスタで最高2つまでのノードがサポートされます。Microsoft Windows 2000 Datacenter Server では、1つのクラスタで最高4つまでのノードがサポートされます。

Oracle Fail Safe が提供する高可用性機能を利用するためには、MSCS の概念を理解することが重要になります。

この章では、次の項目について説明します。

項目	項
クラスタ・テクノロジー	2.1 項
リソース、グループおよび高可用性	2.2 項
グループ、仮想アドレスおよび仮想サーバー	2.3 項
仮想アドレスへの IP アドレス割当て	2.4 項
クラスタ・グループとクラスタ別名	2.5 項
フェイルオーバー	2.6 項
フェイルバック	2.7 項

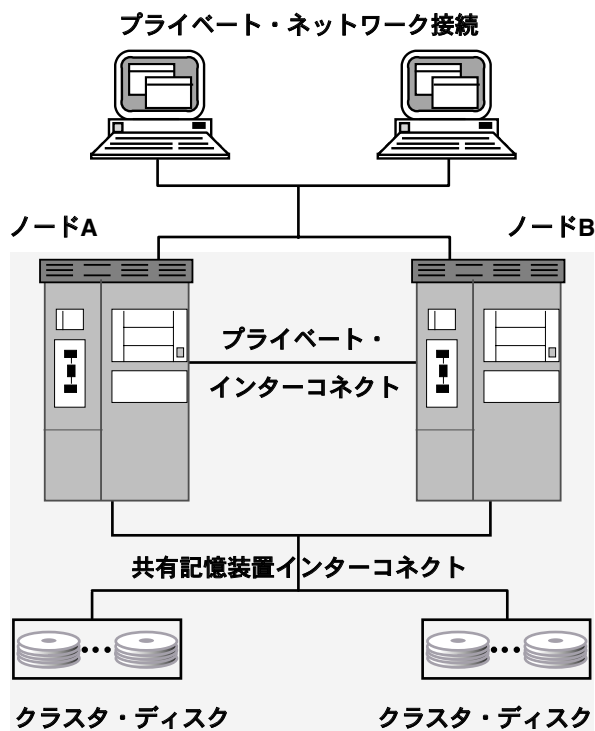
2.1 クラスタ・テクノロジー

クラスタのメンバーである Windows システムを**クラスタ・ノード**と呼びます。クラスタ・ノードは、ノード間のネットワーク接続と同様の共有記憶装置インターコネクトを介して結合されます。

ノード間ネットワーク接続（ハートビート接続とも呼ばれる）では、あるノードが使用可能か不可能かを別のノードから検出できます。この通信には、プライベート・インターコネクト（ユーザーおよびクライアント・アプリケーションのアクセスに使用されるパブリック・ネットワーク接続と区別される）を使用することが一般的です。あるノードに障害が発生した場合、クラスタ・ソフトウェアは使用不能になったノードの作業負荷をただちに使用可能なノードに**フェイルオーバー**し、障害発生ノードが所有していたあらゆるクラスタ・リソースを使用可能なノードに再マウントします。クライアントは、何も変更することなく、クラスタ・リソースに継続してアクセスします。

図 2-1 に、2つのノードから成る Microsoft クラスタ構成でのネットワーク接続を示します。

図 2-1 Microsoft クラスタ・システム



2.1.1 クラスタによる高い可用性の実現方法

クラスタ・テクノロジーが使用可能になるまで、PC システムの信頼性は RAID やミラー化ドライブ、および電源の二重化といったハードウェアの冗長性によって達成されていました。可用性の高いシステムを作成する上でディスク冗長性は重要ですが、この方法のみではシステムとアプリケーションの可用性を保証できません。

MSCS ソフトウェアを使用して Microsoft クラスタ内のサーバーに接続すると、通常の操作中は各サーバー（ノード）がクラスタ・ディスクのサブセットに排他的アクセスすることになり、サーバーの冗長化を実現します。クラスタは、互いに依存しないスタンドアロン・システムよりはるかに効率的です。なぜなら、各ノードはそれぞれ有効な作業を実行できる上に、障害が発生したクラスタ・ノードのディスク・リソースおよび作業負荷を引き継ぐことができるためです。

クラスタは、その設計によって、コンポーネントの障害を管理し、コンポーネントの追加と削減をユーザーに認識させず、高い可用性を実現します。障害の検出、リカバリ、およびクラスタ・ノードを単一のシステムとして管理する能力などのサービスの提供を含む、付加的な利点があります。

注意： 可用性が高まるように冗長性の高いハードウェア（RAID など）を使用する場合の情報は、ハードウェアのマニュアルを参照してください。

2.1.2 システム・レベルの構成

クラスタ構成の設定と使用には、様々な方法があります。Oracle Fail Safe は、アクティブ / パッシブとアクティブ / アクティブの両方のシステム構成をサポートしています。

- アクティブ / パッシブ・クラスタ

アクティブ / パッシブ・クラスタ構成では、1 つ以上のノードがクラスタの**プライマリ・ノード**として動作し、別のノードがバックアップ（つまり **2 次ノード**）として動作します。プライマリ・ノードのみが有効な作業を実行し、2 次ノードは使用不能になったプライマリ・ノードの作業負荷を引き継ぐために待機します。

- アクティブ / アクティブ・クラスタ

アクティブ / アクティブ・クラスタ構成では、クラスタのすべてのノードが同時に有効な作業を実行します。1 つのノードが使用不能になった場合、別のノードがそれ自身の作業負荷に加え、使用不能なノードの作業負荷を引き継ぎます。

第 3 章で、これらの構成に関する詳細を説明します。

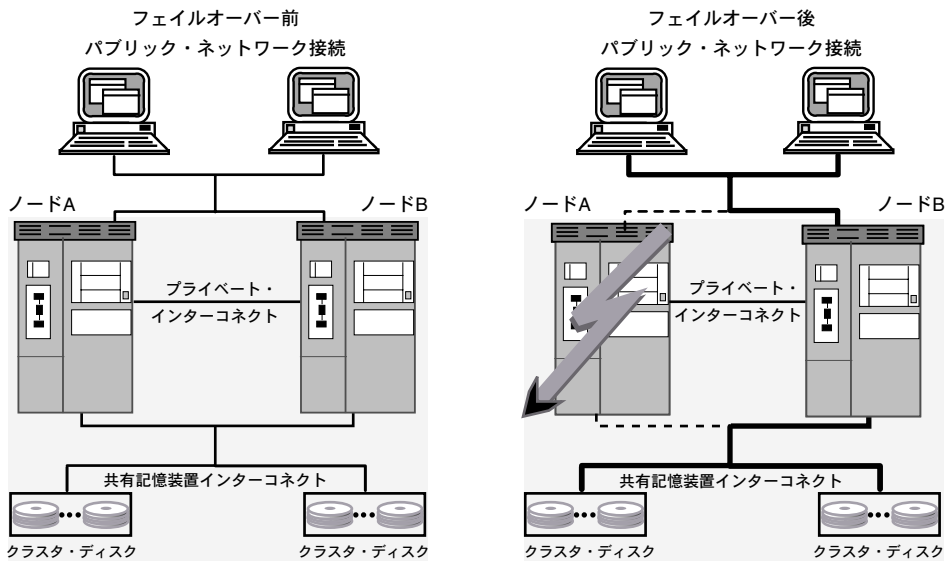
2.1.3 ディスク・レベルの構成

Microsoft クラスタが障害からリカバリする際には、正常に機能しているノードが、シェアード・ナッシング構成を介して障害ノードのディスク・データへのアクセスを獲得します。

シェアード・ナッシング構成では、すべてのノードが物理的に同一ディスクにケーブル接続されていますが、ある時点で特定のディスクにアクセスできるのは1つのノードのみです。すべてのノードが物理的にディスクに接続されていても、そのディスクを所有しているノードしか、それらにアクセスできません。

図 2-2 では、2つのノードから成るクラスタで1つのノードが使用不能になった場合に、障害ノードが所有していたディスクおよびアプリケーション作業負荷の所有権を、もう一方のクラスタ・ノードが引き受け、両方のノードに対する操作の処理を続行することを示しています。

図 2-2 シェアード・ナッシング構成



2.1.4 クオラム・リソース

クオラム・リソースでは、停電やメモリ内のデータ損傷が起こった場合にクラスタのリカバリに必要な構成データ（メタデータ）が保持されます。クオラム・リソースは他のクラスタのリソースにもアクセスできるため、すべてのクラスタ・ノードがクラスタ・メタデータにアクセスすることになります。クオラム・リソースは、次のサービスを実行します。

- どのクラスタ・ノードがそのクラスタを制御するかを決定します。
- 障害からクラスタをリカバリするのに必要なログ情報を格納します。
- 最新のクラスタ・メタデータへのアクセスをメンテナンスします。

ある時点でクオラム・リソースを所有できるクラスタ・ノードは1つのみです。クラスタ・ノードが孤立した場合（たとえば、ネットワーク障害により他のクラスタ・ノードと通信できない場合）、現在クオラム・リソースを所有しているノードが、フェイルオーバー発生の場合と同じように、孤立ノードの作業負荷を受け継ぎます。2つのノードから成るクラスタでこのような状態になると、クオラム・リソースを所有しているノードが単一ノード・クラスタになります。

クオラム・リソースを現在所有しているノードに障害が発生した場合、正常に機能しているノードの中から、クオラム・リソースの制御を獲得するノードが選ばれます。クオラム・リソースは、ここで選択されたノードにフェイルオーバーします。

MSCS をインストールするとき、クオラム・リソースの場所を指定します。クオラム・リソースの場所を忘れた場合は、Oracle Fail Safe Manager で「クラスタのダンプ」コマンドを使用するとその場所を表示できます。「クラスタのダンプ」コマンドを発行するには、次のようにします。

1. 「Oracle Fail Safe Manager」メニューで「トラブルシューティング」をクリックします。
2. 「クラスタのダンプ」をクリックします。
3. クオラム・リソースの場所に関する情報は、「クラスタのダンプ」コマンド出力の最初の部分に表示されます。

2.2 リソース、グループおよび高可用性

あるサーバー・ノードが使用不能になった場合、可用性が高まるように構成されたクラスタ・リソース（ディスク、Oracle データベースとアプリケーション、および IP アドレスなど）は、グループと呼ばれる単位で使用可能なノードに移されます。次の項ではリソース、グループおよび可用性が高まるようにそれらを構成する方法を説明します。

2.2.1 リソース

クラスタ・リソースとは、コンピューティング・システムで使用可能で、次のような特徴を備えた物理的または論理的なコンポーネントです。

- オンライン化とオフライン化が可能です。
- クラスタ内で管理できます。
- ある時点でホストになるノードはクラスタ内で1つのみですが、潜在的には他のクラスタ・ノードによっても所有されます。(たとえば、あるリソースが特定のノードによって所有されている場合です。フェイルオーバーの後、そのリソースは別のクラスタ・ノードによって所有されます。ただし、特定の時点においてリソースにアクセスできるクラスタ・ノードはいずれか1つのみです。)

2.2.2 グループ

グループは、フェイルオーバーの最小単位を形成するクラスタ・リソースの論理的な集合です。フェイルオーバーの際には、グループ単位のリソースが別のノードに移されます。グループは、ある時点では1つのクラスタ・ノードにのみ所有されます。特定の作業負荷（データベース、ディスクおよび他のアプリケーション）のために必要なリソースはすべて、同じグループに常駐している必要があります。

たとえば、Oracle Fail Safe を使用して可用性の高い Oracle データベースを構成するために作成されたグループには、次のようなリソースが含まれる可能性があります。

- Oracle データベースで使用する全ディスク
- Oracle データベース・サーバーのインスタンス
- 次の項目で構成されている1つ以上の仮想アドレス
 - IP アドレス
 - ネットワーク名
- グループ内のデータベースへの接続要求をリスニングする Net8（または SQL*Net）ネットワーク・リスナー
- Oracle Enterprise Manager とグループ内のデータベースの間の通信を管理する Oracle Intelligent Agent

データベースをグループに追加する場合、そこで使用されるディスクは常に同じグループに含まれることに注意してください。そのため、同一のディスクを使用する2つのリソースを異なるグループに含めることはできません。両方のリソースをフェイルセーフにするには、同じグループに含めてください。

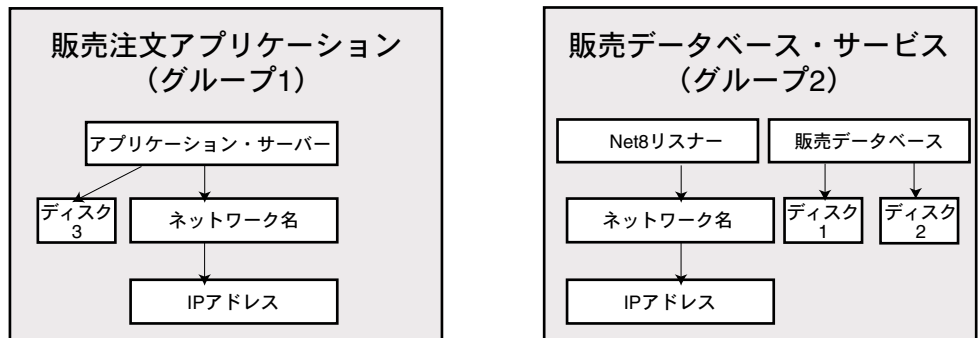
Oracle Fail Safe は、グループを作成し、アプリケーションを実行するために必要なリソースを追加する際に役立ちます。

2.2.3 リソースの依存性

図 2-3 は、注文入力アプリケーションの可用性が高まるように作成されたグループ、販売注文アプリケーションおよび販売データベース・サービスを示します。販売アプリケーションおよびデータベースを作成し、そのデータベース・ファイルをクラスタ・ディスクに置いた後で、Oracle Fail Safe Manager を使用して、そのアプリケーション用とデータベース用にグループを 1 つずつ作成します。

リソースをグループに追加すると、追加したリソースが依存しているリソースが Oracle Fail Safe Manager によって自動的に追加されます。これはリソースの**依存性**と呼ばれます。たとえばデータベースをグループに追加すると、Oracle Fail Safe により、データベースに使用されるディスクが追加され、Net8（または SQL*Net）ファイルがそれぞれのグループとともに動作するように構成されます。また、Oracle Fail Safe では、各グループがそれぞれのノードにフェイルオーバーできるかどうかテストされます。

図 2-3 クラスタ・グループの設計



クラスタ内のノードはそれぞれ 1 つ以上のグループを所有できます。各グループは、依存するリソースまたは関連付けられたリソースで構成されます。リソースの依存関係によって、クラスタ・ソフトウェアがリソースをオンライン化またはオフライン化する順序が定義されます。たとえば、障害が発生するとまず Oracle アプリケーションまたはデータベースがオフライン化され、続いて物理ディスク、ネットワーク名、IP アドレスがオフライン化されます。フェイルオーバー・ノードでは、順序が逆になり、MSCS は最初に IP アドレスをオンライン化し、次にネットワーク名、物理ディスク、最後に Oracle データベースと Ne8 リスナー、またはアプリケーションがオンライン化します。

2.2.4 リソース・タイプ

各リソース（汎用サービス、物理ディスク、Oracle データベースなど）は、リソースの動的リンク・ライブラリ（DLL）と関連付けられており、クラスタ環境ではそのリソース DLL を使用して管理されます。カスタム Oracle リソース DLL に加えて、標準 MSCS リソース DLL があります。同じリソース DLL で異なるリソース・タイプをサポートできます。

- MSCS には、IP アドレス、物理ディスク、汎用サービス、およびその他多数のサポートしているリソース・タイプ用のリソース DLL があります。（**汎用サービス**・リソースは、MSCS が提供するリソース DLL によってサポートされる Windows サービスです。）
- Oracle Fail Safe では、カスタム・サポートが提供されているリソース・タイプ（Oracle Forms、Oracle Reports、Oracle Forms Load Balancer Server、Oracle Applications Concurrent Manager、Oracle WebDB、Oracle HTTP Server、Oracle MTS Service、汎用サービスなど）を監視する際に、多数の MSCS リソース DLL を使用します。
- オラクル社では、Oracle データベースおよび Oracle Application Server のリソース・タイプに対するカスタム DLL を提供しています。MSCS は、Oracle リソース DLL を使用して Oracle リソースを管理（オンライン化およびオフライン化）し、リソースの可用性を監視します。

Oracle Fail Safe Server には次のリソース DLL ファイルが用意されています。この DLL を通じて MSCS は Oracle データベース・リソースおよび Oracle Application Server リソースと通信し、それらを監視できます。

- Oracle データベース・サーバー
 - * FsResOdbbs.DLL が提供する機能によって、MSCS は Oracle データベースをオンライン化またはオフライン化し、Is Alive ポーリングを介してデータベースの状態をチェックできます。
 - * FsResOdbbsEx.DLL によって、MSCS クラスタ アドミニストレータが Oracle データベース・リソースのプロパティ表示に使用するリソース管理拡張 DLL ファイルが提供されます。
- Oracle Application Server
 - * FsResOas.DLL が提供する機能によって、MSCS は Oracle Application Server をオンライン化し、その状態（オフライン、オンラインなど）をチェックできます。
 - * FsResOasEx.DLL によって、MSCS クラスタ アドミニストレータが Oracle Application Server リソースのプロパティ表示に使用するリソース管理拡張 DLL ファイルが提供されます。

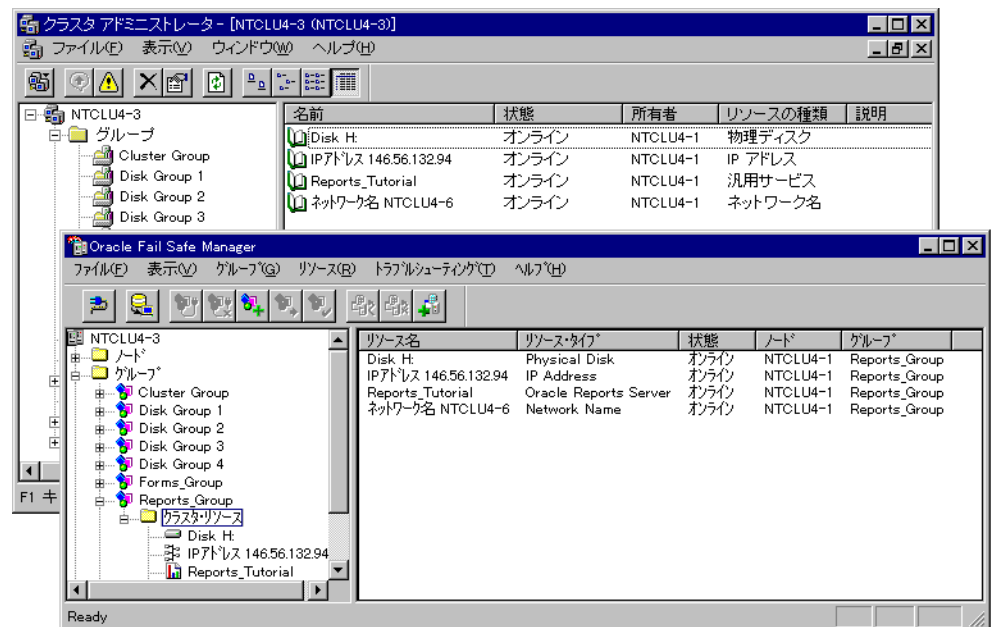
たとえば、Oracle Fail Safe Manager を使用して Oracle データベースをグループに追加すると、Oracle Fail Safe によってデータベース・リソースと Oracle リスナー・リソースが作成されます。

Oracle データベース・リソース DLL (FsResOdbbs.DLL および FsResOdbbsEx.DLL) が提供する機能によって、MSCS は Oracle データベースの状態をチェックし、Oracle データベースをオンライン化またはオフライン化することができます。

MSCS 汎用サービス DLL (clusres.dll) が提供する機能によって、MSCS はリスナーの状態をチェックし、リスナーをオンライン化およびオフライン化することができます。MSCS では、Oracle リスナーが汎用サービスであるとみなされます。ただし、リスナーの追加情報は Oracle Fail Safe Server により保持されるため、Oracle Fail Safe Manager でのこのリソース・タイプは "Oracle TNS Listener" と表示されます。

図 2-4 に、Oracle Fail Safe Manager でのリソース・タイプの表示方法を示します。Oracle Reports Server のリソース・タイプは、Oracle Fail Safe Manager では Oracle Reports Server として表示され、MSCS クラスタ アドミニストレータでは汎用サービスとして表示されるため注意してください。これは Oracle Forms Server、Oracle リスナー、Oracle HTTP Server および Oracle Forms Load Balancer Server の場合も同様です。

図 2-4 可用性の高い Oracle Reports Server のリソース・タイプ



関連項目：

- Oracle Fail Safe によって提供されるカスタム・リソース DLL の完全な情報は、『Oracle Fail Safe インストール・ガイド』を参照してください。
- 標準リソース・タイプおよび標準リソース DLL の詳細は、MSCS のマニュアルを参照してください。

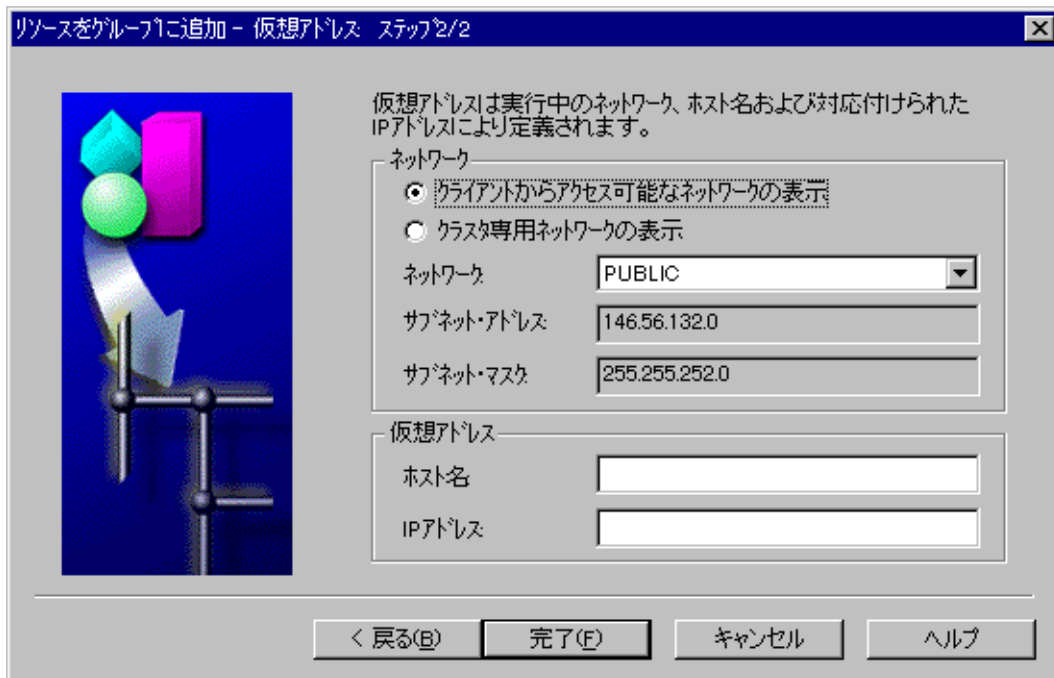
2.3 グループ、仮想アドレスおよび仮想サーバー

仮想アドレスは、グループ内のリソースのホストになっているクラスタ・ノードとは無関係に、リソースにアクセスできるネットワーク・アドレスです。仮想アドレスによって、ノードに依存しない一定のネットワーク位置が提供されます。これにより、リソースのホストになっている物理的クラスタ・ノードを知る必要なく、クライアントが容易にリソースにアクセスできます。

障害が起きている間にグループが使用不能ノードから使用可能ノードに移されるため、クライアントは、1 つのノードによってしか識別されないアドレスを使用しているアプリケーションには接続できません。Oracle Fail Safe Manager 内のグループの仮想アドレスを識別するには、グループに一意のネットワーク名と IP アドレスを付加します。

[図 2-5](#) に、1 つ以上の仮想アドレスをグループに追加するための Oracle Fail Safe Manager のウィザード・ページを示します。

図 2-5 「リソースをグループに追加 - 仮想アドレス」ウィザード・ページ

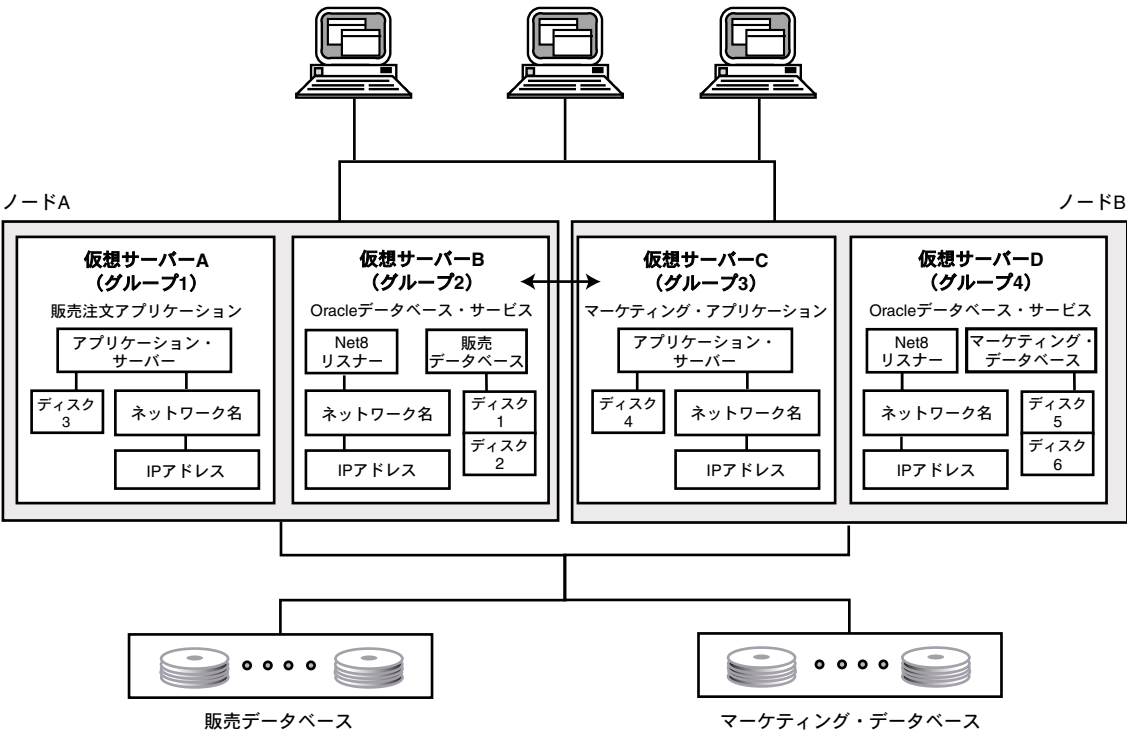


あるグループに仮想アドレスを追加すると、そのグループは**仮想サーバー**になります。クライアント・アクセスのためにグループには少なくとも1つの仮想アドレスが必要ですが、グループに複数の仮想アドレスを割り当てることができます。帯域幅を増やしたり、グループ内のリソースのセキュリティを区分化するために、複数の仮想アドレスを割り当てることができます。

ユーザーやクライアント・アプリケーションからは、各グループは特定のノードの物理的識別情報に依存しない、可用性の高い仮想サーバーのように見えます。グループ内のリソースにアクセスする場合、クライアントは常にグループの仮想アドレスに接続してグループにアクセスします。クライアントにとって、仮想サーバーはクラスタ・リソースへのインタフェースであり、物理ノードのように見えます。

図 2-6 に、各ノードに 2 つのグループを構成した 2 ノード・クラスタを示します。クライアントは仮想サーバー A、B、C および D を通じてこれらのグループにアクセスします。各ノードの物理アドレスではなく、グループの仮想アドレスを介してクラスタ・リソースにアクセスすることで、どのクラスタ・ノードがグループのホストになっているかに関係なく、確実にリモート接続ができます。

図 2-6 仮想サーバーを介したクラスタ・リソースへのアクセス



2.4 仮想アドレスへの IP アドレス割当て

クラスタをセットアップする際に、少なくとも次の数の IP アドレスを割り当てます。

- 各クラスタ・ノードに対して 1 つの IP アドレス
- 各クラスタ別名 (2.5 項を参照) に対して 1 つの IP アドレス
- 各仮想サーバー (グループ) に対して 1 つの IP アドレス

たとえば、図 2-6 の構成では、2 つのクラスタ・ノードのそれぞれに 1 つずつ、クラスタ別名に 1 つ、そして各仮想サーバーに 4 つで、合計 7 つの IP アドレスが必要です。

Oracle Fail Safe 環境での IP アドレス割当ての詳細は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。

2.5 クラスタ・グループとクラスタ別名

クラスタ別名はノードに依存しないネットワーク名で、クラスタを識別し、クラスタ関連のシステム管理に使用されます。MSCS は「クラスタ・グループ」と呼ばれるグループを作成し、クラスタ別名はこのグループの仮想アドレスです。Oracle Fail Safe Server は、クラスタ・グループ内のリソースです。クラスタ・グループは、MSCS によって作成される特別な仮想サーバーです。

Oracle Fail Safe 環境では、クラスタ別名はシステム管理の目的でしか使用されません。Oracle Fail Safe Manager は、クラスタ別名を使用してクラスタ・コンポーネントおよび MSCS と対話します。

Oracle Fail Safe Manager を使用してクラスタに接続する際には、図 2-7 に示すようにクラスタ別名を指定します。クラスタへの接続時にクラスタ別名を指定することにより、Oracle Fail Safe Manager は確実に Oracle Fail Safe Server の稼働している仮想サーバーに接続されます。クラスタ別名は、常に Oracle Fail Safe Server と同じグループに属するためです。

図 2-7 「クラスタに接続」ダイアログ・ボックスの「クラスタ別名」



クライアント・アプリケーションは、仮想サーバーとの通信の際にクラスタ別名を使用しません。かわりに、クライアントのアプリケーションのために構成されたグループの仮想アドレスを使用します。

2.6 フェイルオーバー

あるノード上のグループをオフライン化し、それを別のノード上でオンライン化するプロセスを**フェイルオーバー**と呼びます。フェイルオーバーの発生後、このグループ内のリソースを実行するように構成されたクラスタ・ノードのいずれかが使用可能であれば、そのリソースにアクセスできます。MSCS は、クラスタ内のクラスタ・ノードおよびリソースの状態を継続的に監視します。

フェイルオーバーには、計画的なものと計画外のものがあります。

- 計画外フェイルオーバーは、クラスタ・ソフトウェアがノードまたはリソースの障害を検出した場合に、自動的に行われます。
- 計画的フェイルオーバーは、ロード・バランスやソフトウェアのアップグレードなどの機能を実行する場合に行う手動の操作です。

次の各項で、これら 2 タイプのフェイルオーバーについて詳細に説明します。

2.6.1 計画外フェイルオーバー

計画外フェイルオーバーには、原因が次のいずれかによって、2つのタイプがあります。

- 可用性が高まるように構成されたリソースの障害
- クラスタ・ノードの障害または使用不能状態

2.6.1.1 リソース障害による計画外フェイルオーバー

リソース障害による計画外フェイルオーバーは、次に説明する手順で検出および実行されます。

1. クラスタ・ソフトウェアが、リソースに障害が発生したことを検出します。

リソース障害を検出するために、クラスタ・ソフトウェアはリソースが稼働状態であるかどうかを（リソース DLL 経由で）定期的に問い合わせます。詳細は、[2.6.3 項](#)を参照してください。

2. クラスタ・ソフトウェアが、**リソース再起動ポリシー**を実装します。リソース再起動ポリシーでは、クラスタ・ソフトウェアが現在のノード上でリソース再起動を試行するかどうか、そして再起動を試行する場合には一定時間内に何回それを試行するかを指定します。たとえば Oracle Fail Safe がリソースの再起動を 900 秒間に 3 回試行するなど指定します。

リソースが再起動された場合、クラスタ・ソフトウェアがソフトウェアの監視を再開し（手順 1）、フェイルオーバーは回避されます。

リソースが現在のノード上で再起動されない、またはできない場合、クラスタ・ソフトウェアはリソース・フェイルオーバー・ポリシーを適用します。

リソース・フェイルオーバー・ポリシーでは、リソース障害がグループ障害を引き起こすかどうかを指定します。グループがフェイルオーバーしないというリソース・フェイルオーバー・ポリシーを指定した場合、リソースは障害発生の状態のままとなり、フェイルオーバーは発生しません。

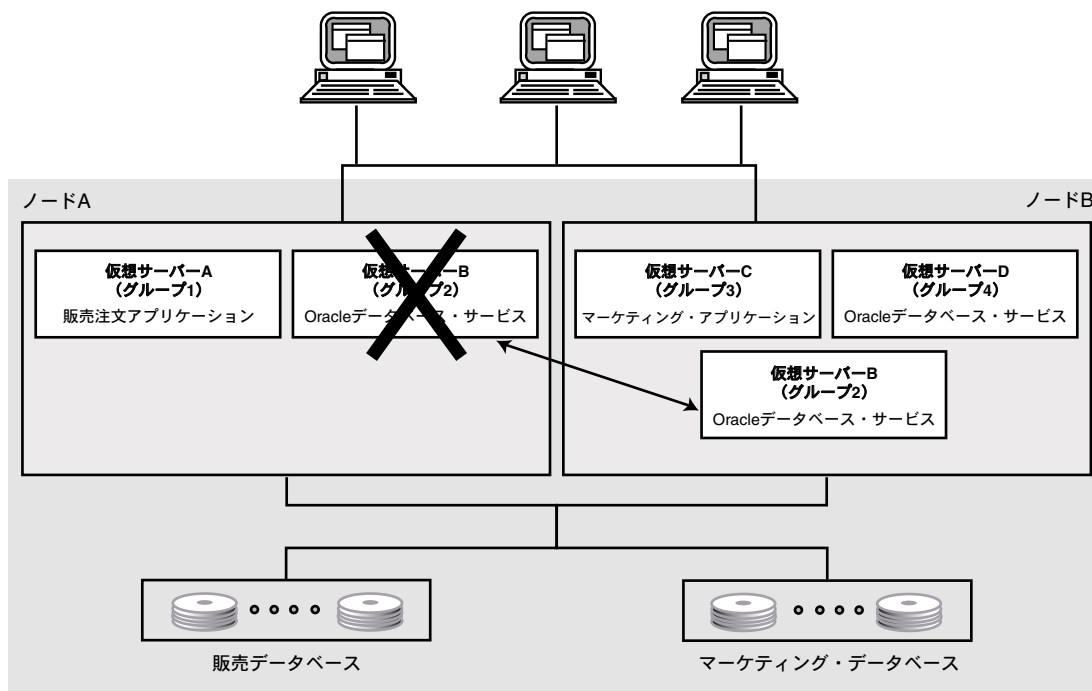
[図 2-11](#) に示すタブ付きのページで、リソースの再起動ポリシーおよびフェイルオーバー・ポリシーを表示または修正できます。

3. リソースが再起動しないまたはできない場合には、グループがフェイルオーバーするというリソース・フェイルオーバー・ポリシーを指定すると、そのグループは別のノードにフェイルオーバーされます。グループのフェイルオーバー先となるノードは、稼働しているノード、そのリソースの「可能所有者ノード」リスト、およびグループの「優先ノード」リストによって決定されます。リソースの「可能所有者ノード」リストの詳細は [2.6.6 項](#)、グループの「優先ノード」リストの詳細は [2.6.9 項](#)を、それぞれ参照してください。

4. グループがフェイルオーバーすると、そのグループのフェイルオーバー・ポリシーが適用されます。**グループ・フェイルオーバー・ポリシー**では、そのグループがオフライン化されるまでに、クラスタ・ソフトウェアが一定時間内に何回のフェイルオーバーを許可するかを指定します。グループ・フェイルオーバー・ポリシーを使用すると、グループが何度もフェイルオーバーすることを防ぐことができます。グループ・フェイルオーバー・ポリシーの詳細は、[2.6.7 項](#)を参照してください。
5. (障害または意図的な再起動のために) 所定のノードがオフライン化され、その後再度オンライン化される場合に、リソースとそれが属するグループがそのノードに戻されるかどうかは、**グループ・フェイルバック・ポリシー**によって決定されます。フェイルバック・ポリシーの詳細は、[2.7 項](#)を参照してください。

図 2-8 では、グループ 2 のリソースの 1 つに障害が発生したために、仮想サーバー B がノード B にフェイルオーバーしています。

図 2-8 リソースのフェイルオーバー



2.6.1.2 ノードの障害または使用不能状態による計画外フェイルオーバー

クラスタ・ノードが使用不能になったことによる計画外フェイルオーバーは、次に説明する手順で検出および実行されます。

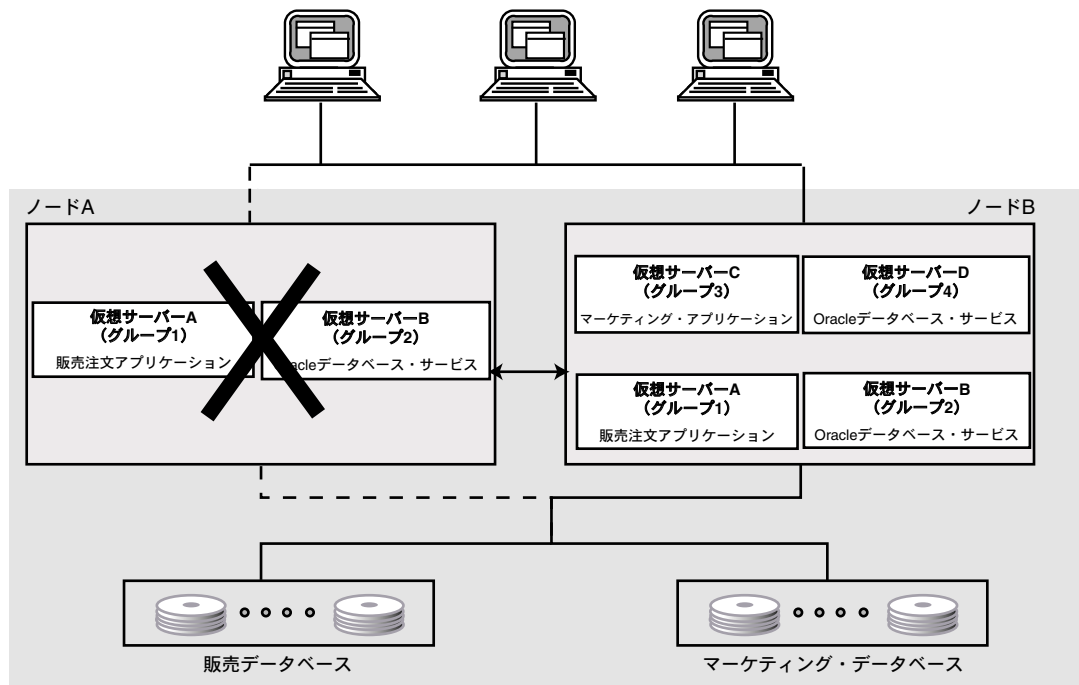
1. クラスタ・ソフトウェアによって、クラスタ・ノードが使用不能になったことが検出されます。

ノードの障害または使用不能状態を検出するために、クラスタ・ソフトウェアは（プライベート・インターコネクトを使用して）定期的にクラスタ内のノードに問合せを行います。
2. 障害の発生した、または使用不能になったノード上のグループが、1つ以上の他のノードにフェイルオーバーされます。フェイルオーバー先のノードは、クラスタ内で使用可能なノード、各グループの「優先ノード」リスト、および各グループ内のリソースの「可能所有者ノード」リストによって決定されます。リソースの「可能所有者ノード」リストの詳細は [2.6.6 項](#)、グループの「優先ノード」リストの詳細は [2.6.9 項](#) を、それぞれ参照してください。
3. グループがフェイルオーバーすると、そのグループのフェイルオーバー・ポリシーが適用されます。**グループ・フェイルオーバー・ポリシー**では、そのグループがオフライン化されるまでに、クラスタ・ソフトウェアが一定時間内に何回のフェイルオーバーを許容するかを指定します。グループ・フェイルオーバー・ポリシーの詳細は、[2.6.7 項](#)を参照してください。
4. リソースとそれが所属するグループが、再度使用可能になったノードに移されるかどうかは、**グループ・フェイルバック・ポリシー**によって決定されます。フェイルバック・ポリシーの詳細は、[2.7 項](#)を参照してください。

図 2-9 では、ノード A での障害発生時にグループ 1 および 2 がフェイルオーバーされることを示しています。クライアント・アプリケーション（障害が発生したサーバーに接続されていたもの）は、フェイルオーバー後、再びサーバーに接続する必要があります。アプリケーションが Oracle データベースに対する更新処理を実行中で、障害発生時に未コミットのデータベース・トランザクションが進行している場合、そのトランザクションはロールバックされます。

ここで説明した手順 3 および 4 は、前述したリスト（[2.6.1.1 項](#)）の手順 4 および 5 と同じであることを注意してください。フェイルオーバーの開始後は、フェイルオーバーの原因がリソース障害かノード障害かにかかわらず、プロセスは同じになります。

図 2-9 ノードのフェイルオーバー



2.6.1.3 計画的フェイルオーバー

計画的フェイルオーバーは、あるノードのクライアント・アプリケーションおよびクラスター・リソースをオフライン化し、別のノードでオンライン化する意図的なプロセスです。これによって、あるクラスター・ノードで管理者が定期的なメンテナンス作業（ハードウェアおよびソフトウェアのアップグレードなど）を実行している間も、ユーザーは別のノードで作業を継続できます。メンテナンス作業以外にも、クラスター内のノード間でロード・バランスを行うために、計画的フェイルオーバーの実行が必要な場合があります。つまり計画的フェイルオーバーは、ノード間でのグループの移動にも使用できるということです。実際、計画的フェイルオーバーを実装する際には、Oracle Fail Safe Manager 内でグループ移動操作を実行します（説明は Oracle Fail Safe Manager のオンライン・ヘルプを参照してください）。

計画的フェイルオーバーの際、Oracle Fail Safe Server は MSCS とともにグループをあるノードから別のノードに効率的に移します。クライアント接続は失われるので、アプリケーションがクラスター対応ではない場合、クライアントはアプリケーションの仮想サーバー・アドレスに手動で再接続する必要があります（クラスター対応アプリケーションの詳細は、[7.10 項](#)を参照してください）。また、Oracle Fail Safe によって、このダウン時間中も中断されることなくクライアントは別のクラスター・ノードで作業できるため、アップグレードをゆっくりと実行できます。（グループに Oracle データベースが含まれている場合、どの計画的フェ

イルオーバーよりも先にデータベースのチェックポイント取得が実行され、新しいノードでの高速データベース・リカバリが確実に行われます。)

2.6.2 フェイルオーバーに影響するグループおよびリソースのポリシー

リソースおよびグループの各フェイルオーバー・ポリシーの値は、Oracle Fail Safe Manager を使用してグループを作成またはグループにリソースを追加すると、デフォルトに戻されます。ただし、これらのポリシーの値は、グループの「フェイルオーバー」タブ・ページ、グループの「フェイルバック」タブ・ページおよびリソースの「ポリシー」タブ・ページで再設定が可能です。グループ・フェイルバック・ポリシーの値は、グループ作成時またはそれ以降に、グループの「フェイルバック」タブ・ページを使用して設定できます。

図 2-10 に、グループ・フェイルオーバー・ポリシーを設定するページを示します。このページにアクセスするには、Oracle Fail Safe Manager のツリー・ビューで該当のグループを選択し、「フェイルオーバー」タブをクリックします。

図 2-11 に、リソースのポリシーを設定するページを示します。このページにアクセスするには、Oracle Fail Safe Manager のツリー・ビューで該当のリソースを選択し、「ポリシー」タブをクリックします。

図 2-10 グループの「フェイルオーバー」タブ・ページ

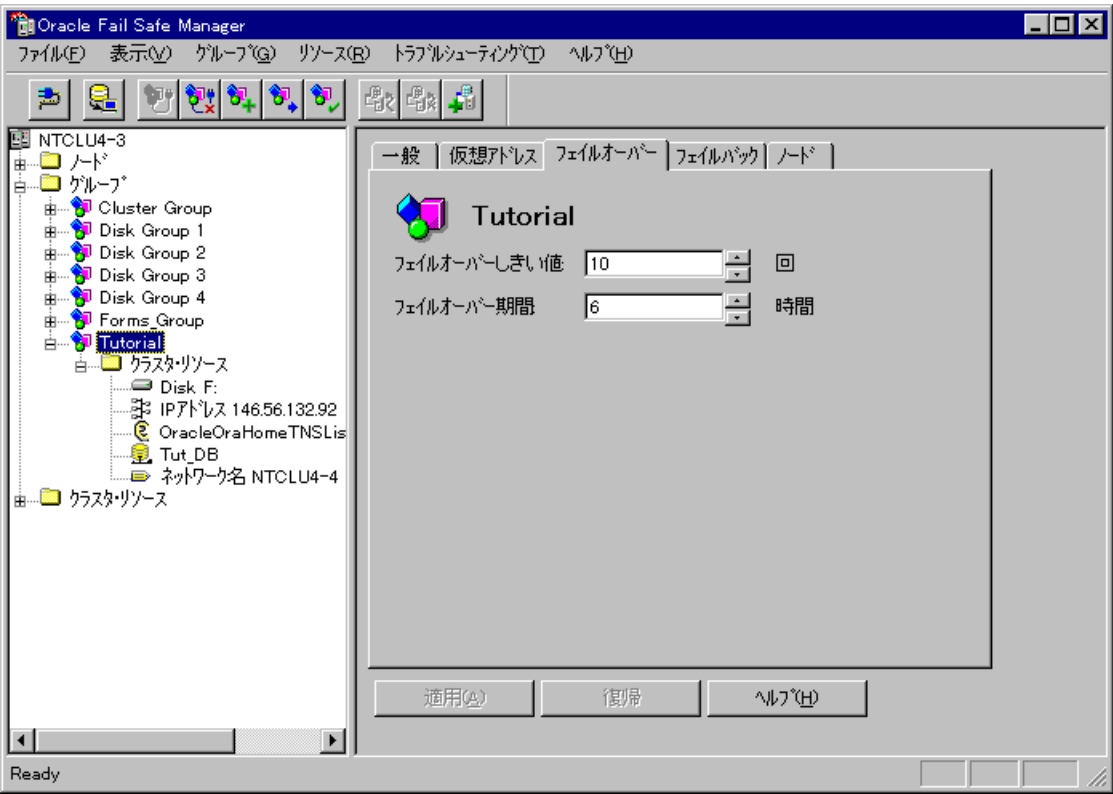


図 2-11 リソースの「ポリシー」タブ・ページ



2.6.3 リソース障害の検出方法

可用性が高まるように構成されたリソースは、すべてクラスタ・ソフトウェアによってその状態が監視されます。リソース障害は次の3つの値に基づいて検出されます。

- 保留タイムアウト値

保留タイムアウト値は、障害が発生したと判断する前に保留状態のリソースがオンライン化（またはオフライン化）されるまで、クラスタ・ソフトウェアが待機する時間の長さを指定します。デフォルト値は180秒です。

- Is Alive ポーリング間隔

Is Alive ポーリング間隔は、クライアント・ソフトウェアがリソースの状態をチェックする頻度を指定します。リソース・タイプに対するデフォルト値を使用することも、数

値を指定することもできます（ミリ秒単位）。このチェックは、Looks Alive ポーリング間隔で行われるチェックよりも完全ですが、消費する CPU リソースも多くなります。

- Looks Alive ポーリング間隔

Looks Alive ポーリング間隔は、クライアント・ソフトウェアがリソースの登録済の状態をチェックし、リソースがアクティブかどうかを判断する頻度を指定します。リソース・タイプに対するデフォルト値を使用することも、数値を指定することもできます（ミリ秒単位）。このチェックは、Is Alive ポーリング間隔で行われるチェックよりも不完全ですが、消費する CPU リソースは少なくなります。

2.6.4 リソース再起動ポリシー

リソースで障害が発生したことが確定すると、クラスタ・ソフトウェアによってそのリソースの再起動ポリシーが適用されます。リソース再起動ポリシーには、[図 2-11](#) で示すように 2 つのオプションがあります。

- クラスタ・ソフトウェアが、現在のノードでリソースの再起動を試みない。そのかわりに、リソース・フェイルオーバー・ポリシーをただちに適用します。
- クラスタ・ソフトウェアが、現在のノードでリソースの再起動を、一定時間内に指定した回数試みる。リソースが再起動できない場合、クラスタ・ソフトウェアはリソース・フェイルオーバー・ポリシーを適用します。

2.6.5 リソースのフェイルオーバー・ポリシー

リソース・フェイルオーバー・ポリシーは、現在のノードでリソースが再起動されない、または再起動できない場合に、そのリソースを含むグループがフェイルオーバーするかどうかを決定します。障害リソースを含むグループがフェイルオーバーしないというポリシーが指定されている場合、リソースは現在のノードで障害発生状態のままとなります。（このグループが、最終的にはフェイルオーバーする可能性はあります。このグループ内の別のリソースが、障害リソースを含むグループはフェイルオーバーするというポリシーを指定されていれば、そこでフェイルオーバーするためです。）障害リソースを含むグループがフェイルオーバーするというポリシーが指定されている場合、障害リソースを含むグループは、「優先ノード」リストによる指定に応じて別のクラスタ・ノードにフェイルオーバーします。（「優先ノード」リストの説明は、[2.6.9 項](#)および[2.7.1 項](#)を参照してください。）

2.6.6 リソースの「可能所有者ノード」リスト

「可能所有者ノード」リストは、指定されたリソースの実行が許可されているすべてのノードから成ります。リソースの実行が許可されるノードは、次のように定義されます。

- そのリソースの .dll がノード上にインストールされていること
- そのノードが「可能所有者ノード」リストから除外されるように指定していないこと

さらに、定義上の要件とはされていませんが、指定のノード上で実行を許可されるリソースはすべて、そのノード上で実行するために構成されている必要があります。そのように構成されていない場合、そのリソースを含むグループはノードにフェイルオーバーしますが、リソースの再起動はできなくなります。リソースは、次のいずれかを実行すると可能所有者ノードで実行するように構成されます。

- そのノードをグループの可能所有者ノードとして現在含んでいるグループに、リソースを追加します。

特定のリソース・タイプをグループに追加する方法の詳細は、第 III 部を参照してください。

- 「グループの検証」コマンドを実行します。

リソースをグループに追加した後でノードがそのグループの可能所有者ノードになった場合、「グループの検証」コマンドが、そのノード上でグループを構成するよう要求します。「グループの検証」コマンドによって、グループのすべての可能所有者ノードがそのグループ用に構成されます。「グループの検証」コマンドの詳細は、[6.1.2 項](#)を参照してください。

前述したように、ノードを「可能所有者ノード」リストから除外する指定が可能です。たとえば、4 つのノードから成るクラスタがあり、各ノードには Oracle8i データベースと、Oracle Fail Safe のデータベース・リソース .dll がインストールされているとします。4 つのノードすべてを、リソース用の可能所有者ノードに指定する方法もあります。しかしここでは、Node_3 にはデータベースとそれ以外の作業負荷を両方とも実行する十分なメモリーがないと仮定します。Node_3 は、データベース・リソース用の「可能所有者ノード」リストから除外することにします。

リソースをグループに追加する際に、そのリソース用の「可能所有者ノード」リストを指定してください。可用性を高めたリソース用の「可能所有者ノード」リストは、次のいずれかのタブ・ページを使用して調整できます。

- リソースの「一般」タブ・ページ

リソースの「一般」タブ・ページでは、リソースの「可能所有者ノード」リストへの変更によって、そのリソースが属しているグループがどのような影響を受けるかは示されません。このタブ・ページを使用してリソースの「可能所有者ノード」リストを変更する場合には、グループ内のリソースのいずれも、それらの「可能所有者ノード」リストに共通のノードを持たないという状態にならないよう注意してください。

- リソースを含むグループの「ノード」タブ・ページ

「ノード」タブ・ページには、グループの「可能所有者ノード」リストが表示されます。ただし、「可能所有者ノード」リストは、実際にはグループの属性ではありません。Oracle Fail Safe では、グループ内の各リソースの「可能所有者ノード」リストの共通点を検出することによって、グループの「可能所有者ノード」リストに表示されるノードが決定されます。このタブ・ページを使用して可能所有者ノードの 1 つを削除すると、グループの可能所有者ノードはまったくなくなることがわかります。[図 2-12](#) は、「ノード」タブ・ページの例です。グループの「可能所有者ノード」リストに変更を加える

と、同じグループ内のすべてのリソース（ディスク・リソースは除く）にその変更が適用されることに注意してください。

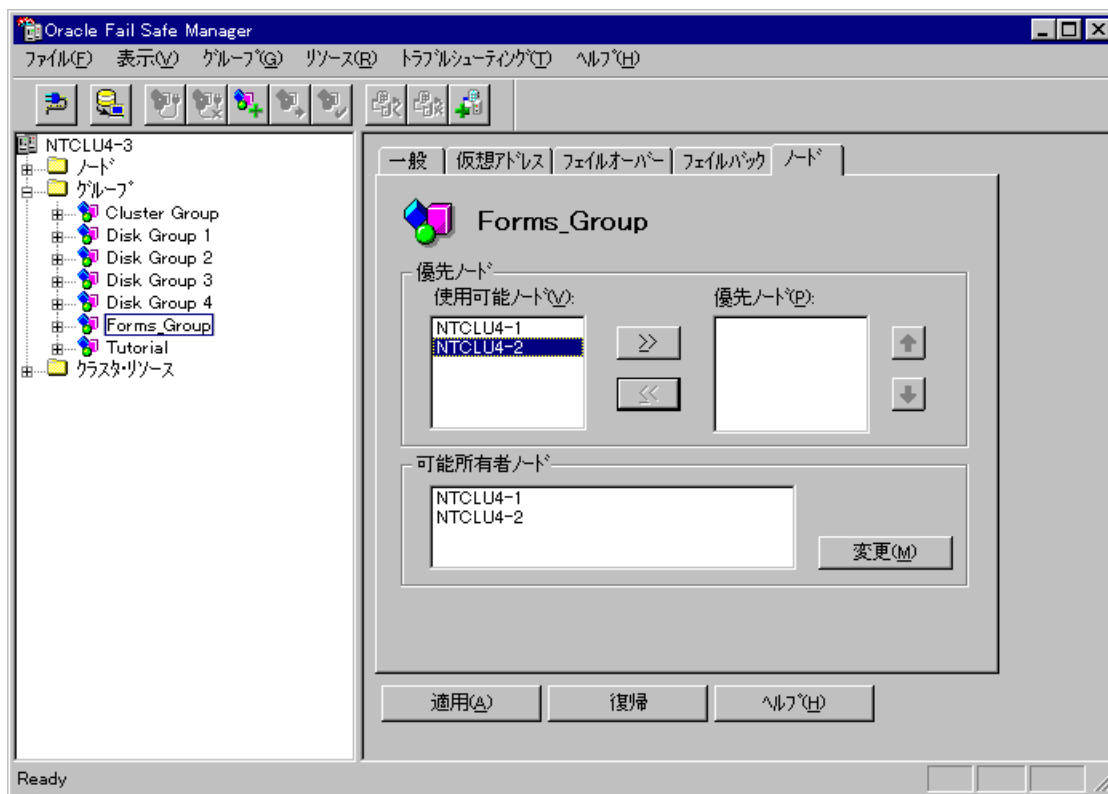
2つのノードから成るクラスタでは、すべてのリソースの「可能所有者ノード」リストに両方のノードが含まれるのが普通です。フェイルオーバー機能を使用するには、少なくとも2つのクラスタ・ノードがリソースの可能所有者ノードである必要があります。

注意： クラスタに新規のノードを追加し、そのノードには Oracle Fail Safe または MSCS DLL（あるいはその両方）がインストールされているとします。このノードは、インストールされている DLL でサポートされるリソースの可能所有者になります。リソースがまだ、そのノード上での可用性が高まるように構成されていない場合、グループはそのノードにフェイルオーバーし、リソースを再起動することはできません。

ただし、「グループの検証」コマンドを実行すると、特定グループ内のリソースが、そのグループの可能所有者となっている各ノードで実行するように構成されているかどうか、Oracle Fail Safe によってチェックされます。グループ内のリソースが実行するように構成されていない可能所有者ノードが検出されると、Oracle Fail Safe がそれを構成します。

このような理由からオラクル社では、新規のノードが可能所有者としてリストされた各グループについては、「グループの検証」コマンドを実行することを強くお勧めします。「グループの検証」コマンドは、[6.1.2 項](#)で説明します。

図 2-12 「ノード」タブ・ページ



2.6.7 グループのフェイルオーバー・ポリシー

リソースが現在のノード上で再起動できない場合には、そのリソースを含むグループがフェイルオーバーするというリソース・フェイルオーバー・ポリシーを指定すると、そのグループはフェイルオーバーし、グループ・フェイルオーバー・ポリシーが適用されます。同様に、ノードが使用不能になった場合にも、そのノード上のグループがフェイルオーバーし、グループ・フェイルオーバー・ポリシーが適用されます。

グループ・フェイルオーバー・ポリシーでは、そのグループがオフライン化されるまでに、クラスタ・ソフトウェアが一定時間内に何回のフェイルオーバーを許容するかを指定します。グループ・フェイルオーバー・ポリシーを使用すると、グループが何度もフェイルオーバーするのを防ぐことができます。

グループ・フェイルオーバー・ポリシーは、フェイルオーバーしきい値とフェイルオーバー期間とで構成されます。

- フェイルオーバーしきい値

フェイルオーバーしきい値では、クラスタ・ソフトウェアがグループ・フェイルオーバーの試行を停止するまでに（フェイルオーバー期間内で）発生するフェイルオーバーの最大回数を指定します。

- フェイルオーバー期間

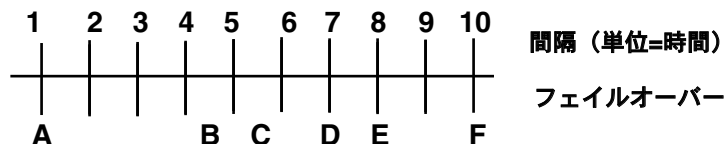
フェイルオーバー期間は、クラスタ・ソフトウェアがフェイルオーバー発生回数をカウントする時間です。フェイルオーバーの頻度が、フェイルオーバー期間に指定した時間内に、フェイルオーバーしきい値に指定した回数を超えると、クラスタ・ソフトウェアはグループ・フェイルオーバーの試行を停止します。

たとえば、フェイルオーバーしきい値が3でフェイルオーバー期間が5とした場合、クライアント・ソフトウェアがそのグループのフェイルオーバーを中止するまでに、5時間以内で3回のフェイルオーバーが許容されます。

最初のフェイルオーバーが発生すると、フェイルオーバー期間を測定するタイマーは0にリセットされ、フェイルオーバー回数を測定するカウンタは1に設定されます。フェイルオーバー期間を超過した時点では、タイマーは0にリセットされません。そのかわりに、フェイルオーバー期間を超過してから最初のフェイルオーバーが発生した時点で、タイマーは0にリセットされます。

たとえば、前述の例と同じくフェイルオーバー期間が5で、フェイルオーバーしきい値が3であるとして、[図 2-13](#)に示すように、Aの時点で最初のフェイルオーバーが発生すると、タイマーが0にリセットされます。2度目のフェイルオーバーが4.5時間後のBの時点で発生し、3度目のフェイルオーバーはCの時点で発生すると仮定します。フェイルオーバー期間を超過するのは、3度目のフェイルオーバー発生時点ということになります。したがってCの時点ではフェイルオーバーの続行が許可され、タイマーが0に、カウンタは1にそれぞれリセットされます。

図 2-13 フェイルオーバーしきい値とフェイルオーバー期間の関係を示す時間軸



Dの時点でもう1回フェイルオーバーが発生するとします。時間軸全体を見ると、フェイルオーバーは停止することが予想されます。B、CおよびDの時点でのフェイルオーバーは、5時間以内に発生しています。しかし、フェイルオーバー期間を測定するタイマーはCの時点で0にリセットされており、フェイルオーバーしきい値は超えていないため、このグループのフェイルオーバーはクラスタ・ソフトウェアによって許可されます。

次のフェイルオーバーがEの時点で発生するとします。通常はフェイルオーバーになるような問題がFの時点で発生しても、クラスタ・ソフトウェアはこのグループをフェイルオーバーしません。これは、Cの時点でタイマーが0にリセットされて以降の5時間以内に、3回のフェイルオーバーが発生したためです。このグループは、クラスタ・ソフトウェアによって障害発生状態のまま現在のノード上に残されます。

2.6.8 リソース再起動ポリシーおよびグループ・フェイルオーバー・ポリシーによるフェイルオーバー機能への影響

リソース再起動ポリシーと、リソースを含むグループのフェイルオーバー・ポリシーは、どちらもグループのフェイルオーバー機能に影響します。

たとえば、Customers というグループに Northeast データベースがあり、次のように指定するとします。

- Northeast データベースの「ポリシー」ページで次のように指定します。
 - 現在のノードで、600 秒（10 分）以内に 3 回データベースの再起動を試行
 - リソースに障害が発生し再起動が不可能な場合には、グループをフェイルオーバー
- Customers グループの「フェイルオーバー」ページで次のように指定します。
 - リソースを含むグループのフェイルオーバーしきい値は 20
 - リソースを含むグループのフェイルオーバー期間は 1 時間

データベース障害が発生したと仮定します。Oracle Fail Safe では、現在のノードでデータベースの再起動が試みられます。データベース再起動の試みが、10 分間で 3 回失敗したとします。この場合、Customers グループは別のノードにフェイルオーバーします。

そのノードで、Oracle Fail Safe はデータベースの再起動を試みますが、10 分間に 3 回失敗したため、Customers グループは再度フェイルオーバーします。Oracle Fail Safe によるデータベース再起動の試みと Customers グループのフェイルオーバーは、データベースが再起動するか、あるいはグループのフェイルオーバーが 1 時間以内に 20 回を超えるまで継続します。データベースが再起動できず、グループのフェイルオーバーが 1 時間以内に 20 回より少ない場合、Customers グループは何度もフェイルオーバーを続けます。このような場合には、フェイルオーバーが何度も続かないようにフェイルオーバーしきい値を低くすることを検討してください。

2.6.9 グループ・フェイルオーバーと「優先ノード」リスト

グループを作成する際に、グループ・フェイルオーバーおよびフェイルバック両方の「優先ノード」リストを作成します。（クラスタ内のノードが2つのみの場合、このリストをフェイルバックのみに指定します。）リスト中の各ノードがグループを所有する優先順位を示すために、順序付けられたノードのリストを作成します。

たとえば4つのノードから成るクラスタで、あるデータベースを含むグループについて次のような「優先ノード」リストを指定するとします。

- ノード 1
- ノード 4
- ノード 3

このように指定すると、4つのノードすべてが稼働している場合にはグループが優先的にノード 1 で実行されます。ノード 1 が使用不能になると、グループは2番目にリストされているノード 4 で実行されます。ノード 1 もノード 4 も使用不能な場合、次にリストされているノード 3 でグループが実行されます。「優先ノード」リストからは、ノード 2 が除外されています。ただし、クラスタ・ソフトウェアで使用可能なノードが他にない（ノード 1、ノード 4 およびノード 3 のすべてで障害が発生したために）場合、グループはノード 2 にフェイルオーバーします。（ノード 2 が、グループ内のすべてのリソースの可能所有者とは限らない場合でも同様です。このような場合、グループはフェイルオーバーしますが、障害発生状態のままになります。）

フェイルオーバーが発生すると、クラスタ・ソフトウェアは「優先ノード」リストを使用して、グループのフェイルオーバー先となるノードを決定します。グループのフェイルオーバー先となるのは、リスト中のノードのうち、稼働しており、かつそのグループの可能所有者ノードである最上位のノードです。クラスタ・ソフトウェアがグループのフェイルオーバー先を決定する方法の詳細は、[2.6.10 項](#)で説明します。

「優先ノード」リストによるフェイルバックへの影響の詳細は、[2.7.1 項](#)を参照してください。

2.6.10 グループのフェイルオーバー・ノードの決定

グループのフェイルオーバー先となるノードは、次の3つのリストに基づいて決定されます。

- 使用可能なクラスタ・ノードのリスト

使用可能なクラスタ・ノードのリストは、グループ障害の発生時に稼働している全ノードから成ります。たとえば、4つのノードから成るクラスタがあるとして、グループのフェイルオーバー時に1つのノードが停止している場合、使用可能なクラスタ・ノードのリスト内は3つになります。

- グループ内の各リソースの「可能所有者ノード」リスト ([2.6.6 項](#)を参照)
- リソースを含むグループの「優先ノード」リスト ([2.6.9 項](#)を参照)

クラスタ・ソフトウェアでは、使用可能なクラスタ・ノードと、グループ内の全リソースに共通の可能所有者セットとの共通点を検出して、グループのフェイルオーバー先となる可能性があるノードを決定します。たとえば、4つのノードから成るクラスタで、ノード3に Test_Group というグループがあるとします。Test_Group 内のリソースには、表 2-1 に示すような可能所有者を指定してあります。

表 2-1 Test_Group グループ内のリソースの可能所有者の例

リソース 1 の可能所有者	リソース 2 の可能所有者	リソース 3 の可能所有者
ノード 1	ノード 1	ノード 1
ノード 2	ノード 3	ノード 2
ノード 3	ノード 4	ノード 3
ノード 4		ノード 4

表 2-1 を見ると、3つのリソースすべてに共通な可能所有者は次のノードであることがわかります。

- ノード 1
- ノード 3
- ノード 4

ノード 3（現在 Test_Group が常駐している）で障害が発生したと仮定します。使用可能なノードは次のようになります。

- ノード 1
- ノード 4

Test_Group のフェイルオーバー先となるノードを決定するために、クラスタ・ソフトウェアはグループ内の全リソースに共通する可能所有者ノードのリストと、使用可能なノードのリストとの共通点を検出します。この例では、これら2つのリストで共通するのはノード 1 およびノード 4 です。

Test_Group のフェイルオーバー先となるノードの決定には、そのグループの「優先ノード」リストが使用されます。Test_Group の「優先ノード」リストを次のように設定してあるとします。

- ノード 3
- ノード 4
- ノード 1

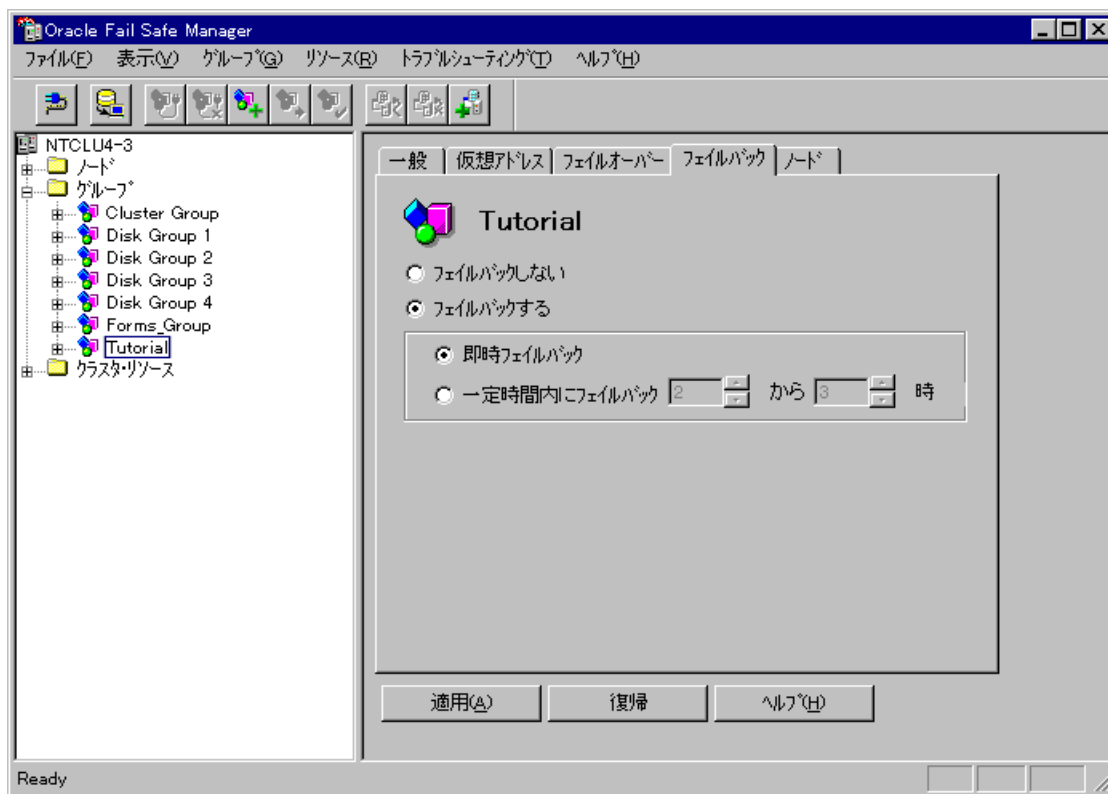
ノード 3 は障害が発生しているため、クラスタ・ソフトウェアは `Test_Group` をノード 4 にフェイルオーバーします。ノード 3 およびノード 4 がいずれも使用不能の場合、`Test_Group` はノード 1 にフェイルオーバーされます。もしノード 1、3 および 4 がすべて使用不能であれば、`Test_Group` はノード 2 にフェイルオーバーされます。ただし、ノード 2 は `Test_Group` 内のすべてのリソースの可能所有者ではないので、`Test_Group` はノード 2 で障害発生状態のままになります。

2.7 フェイルバック

フェイルバックは、優先ノードが動作可能状態に復帰した後で、クラスタ・リソースのグループがフェイルオーバー・ノードから優先ノードに自動的に戻るプロセスです。**優先ノード**とは、可能な場合（そのノードが使用可能である場合）にグループを常駐させるノードです。

フェイルバック・ポリシーを設定して、グループがフェイルオーバー・ノードから優先ノードにフェイルバックする条件（およびフェイルバックするかどうか）を指定します。たとえば、あるグループがただちにフェイルバックする、または選択した特定の時間内にフェイルバックするように設定が可能です。あるいは、グループがフェイルバックせずに現在常駐しているノードで稼働し続けるようなフェイルバック・ポリシーも設定できます。[図 2-14](#) は、グループのフェイルバック・ポリシーを設定するタブ・ページを示しています。

図 2-14 グループの「フェイルバック」タブ・ページ



2.7.1 グループ・フェイルバックと「優先ノード」リスト

クラスタ上にグループを作成する際に、グループ・フェイルオーバーおよびフェイルバックの「優先ノード」リストを作成します。(クラスタ内のノードが2つのみの場合、このリストをフェイルバックのみに指定します。) 順序付けられたノードのリストを作成して、グループを優先的に実行するノードを指定します。以前に使用不能だったノードが再度オンライン化されると、クラスタ・ソフトウェアはクラスタ上の各グループの「優先ノード」リストを読み込み、再オンライン化されたこのノードがいずれかのグループの優先ノードであるかどうかを判断します。「優先ノード」リスト中で、再オンライン化されたノードの順位が、現在グループの常駐しているノードよりも上位である場合、グループは再オンライン化されたこのノードにフェイルバックされます。

たとえば4つのノードから成るクラスタで、My_Group というグループについて次のような「優先ノード」リストを指定するとします。

- ノード 1
- ノード 4
- ノード 3

ノード 1 がオフライン化されたため、My_Group はノード 4 にフェイルオーバーし、現在そこで稼働中であると仮定します。ここでノード 1 が再オンライン化されます。クラスタ・ソフトウェアが My_Group (およびクラスタ上の他の全グループ) の「優先ノード」リストを読み込み、My_Group の優先ノードがノード 1 であることが検出されます。フェイルバックが有効であれば、My_Group はノード 1 にフェイルバックされます。

My_Group が現在ノード 3 で稼働中 (ノード 1 もノード 4 も使用不能なため) であり、ノード 4 が再オンライン化された場合、フェイルバックが有効であれば My_Group はノード 4 にフェイルバックします。この後でノード 1 が使用可能になった場合、My_Group は再び、今度はノード 1 にフェイルバックします。「優先ノード」リストを指定する際には、フェイルバックが不必要に何度も発生することのないように注意してください。大部分のアプリケーションでは、「優先ノード」リストのノードは2つあれば十分です。

グループが、あるノードに手動で移動された場合、予期しない結果になります。すべてのノードが使用可能で、My_Group は現在ノード 4 で稼働しているとします (これは、グループの移動操作で My_Group をノード 4 に移動したためです)。ノード 2 がリポートされると、たとえノード 1 (My_Group の「優先ノード」リストで最上位のノード) が稼働中であっても、My_Group はノード 2 にフェイルバックします。

ノードが再オンライン化されると、再オンライン化されたそのノードが、「優先ノード」リストの中で、現在各グループの常駐しているノードよりも上位であるかどうかクラスタ・ソフトウェアによってチェックされます。上位である場合、そのようなグループはすべて、再オンライン化されたノードに移動されます。

「優先ノード」リストによるフェイルオーバーへの影響の詳細は、[2.6.9 項](#)を参照してください。

2.7.2 フェイルオーバー後のクライアントの再接続

ノードの障害は、次のユーザーおよびアプリケーションにのみ影響します。

- 障害発生ノードがホストになっているアプリケーションに、直接接続しているもの
- ノードの障害発生時にトランザクション処理中であったもの

多くの場合、障害発生ノードに接続していたユーザーおよびアプリケーションは接続を失うことになり、処理を継続するためにはフェイルオーバー・ノードに（ノードに依存しない仮想アドレスを通じて）再接続する必要があります。Web アプリケーションの場合、コミットされていないフォームまたはレポートのコンテキストは失われます。ユーザーがアプリケーションに再接続するには、Web ブラウザで URL を再読み込みします。データベースの場合、障害発生時に処理中だった未コミット・トランザクションは、ロールバックされます。クラスタ対応のクライアント・アプリケーションではサービスが短時間中断され、クライアント・アプリケーションからはノードが即時にリポートされたように見えます。サービスはフェイルオーバー・ノードで自動的に再起動されます。オペレータの介入は必要ありません。

クラスタ対応アプリケーションの詳細は、[7.10 項](#)を参照してください。

Oracle Fail Safe ソリューションの設計

Oracle Fail Safe には多数の構成オプションがあり、どのようなアーキテクチャ要件またはフェイルオーバー要件にも対応します。

この章では、次の項目について説明します。

項目	参照
構成のカスタマイズ	3.1 項
クライアントとアプリケーションの統合	3.2 項

3.1 構成のカスタマイズ

可用性の高いソリューションを配置するには、主に次の 4 つの方法があります。

- アクティブ / パッシブ構成
- アクティブ / アクティブ構成
- 作業負荷のパーティション化構成
- 階層化構成

作業をクラスタ・ノード間で割り当てる方法はそれぞれの構成で異なりますが、次に示すことはすべての構成に共通しています。

- 1 つ以上の Oracle ホームが、各ノードのプライベート・ディスク（通常はシステム・ディスク）に作成されます。
- 必要な Oracle 製品の実行可能プログラムがすべて、各ノードの Oracle ホームにインストールされます。
- 可用性を高めようとするアプリケーションで必要なすべてのデータ・ファイル、構成ファイル、ログ・ファイル、HTML ファイルなどは、クラスタ・ディスク上に置かれるため、各クラスタ・ノードからアクセスが可能です。

Oracle Fail Safe Server ソフトウェアは、適切な構成とフェイルオーバーを保証するために、必要に応じて1つ以上のクラスタ・ノードで自動的に実行されます。

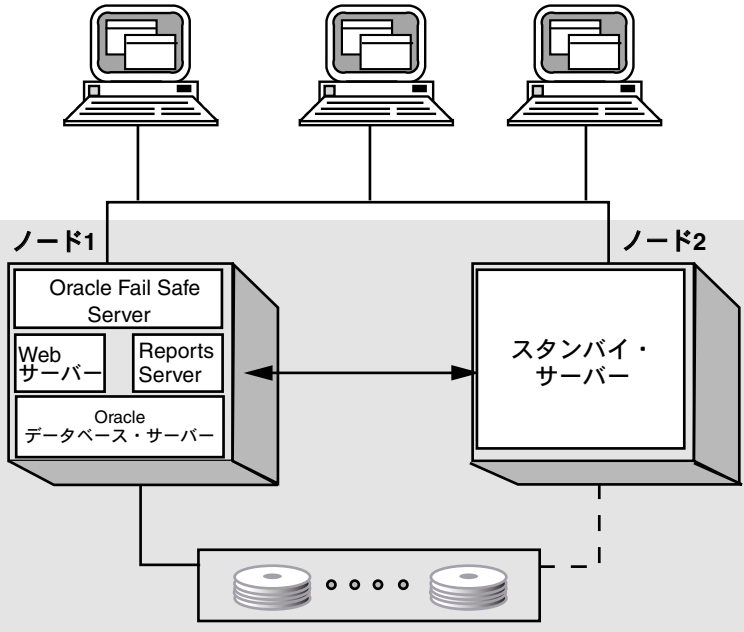
図 1-4 に、Oracle Fail Safe によって構成されたクラスタのソフトウェアおよびハードウェア・コンポーネントを示します。

3.1.1 アクティブ/パッシブ構成

最も単純な構成は、アクティブ / パッシブ構成です。この構成では、1つ以上のノードがクラスタ全体の作業負荷（Oracle データベース・サーバー、Forms Server、Reports Server または Web サーバーなど）のホストになりますが、1つのノードは（スタンバイ・サーバーとして）アイドル状態を維持し、アプリケーションを実行しているノードで障害が発生した場合に処理を引き継ぐために待機します。このソリューションでは、フェイルセーフな作業負荷のパフォーマンスがフェイルオーバーの前後で同じになります。

図 3-1 は、ノード 1 に Oracle データベース、Oracle Reports Server および Web サーバーを置き、ノード 2 をスタンバイ・ノードとした 2 ノードの構成です。

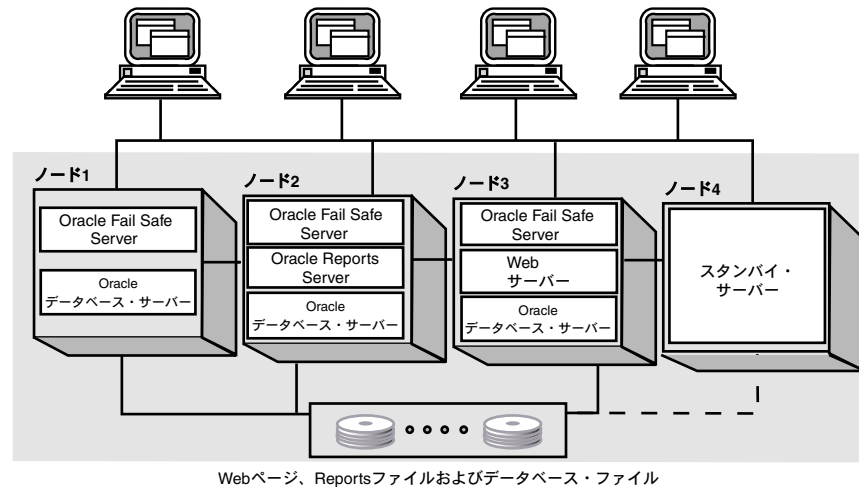
図 3-1 2 ノードのアクティブ/パッシブ（スタンバイ）構成



Webページ、Reportsファイルおよびデータベース・ファイル

図 3-2 は、ノード 1 に Oracle データベース、ノード 2 に Oracle Reports Server、ノード 3 に Web サーバーを置いた 4 ノードの構成です。ノード 4 がスタンバイ・ノードになります。

図 3-2 4 ノードのアクティブ/パッシブ（スタンバイ）構成



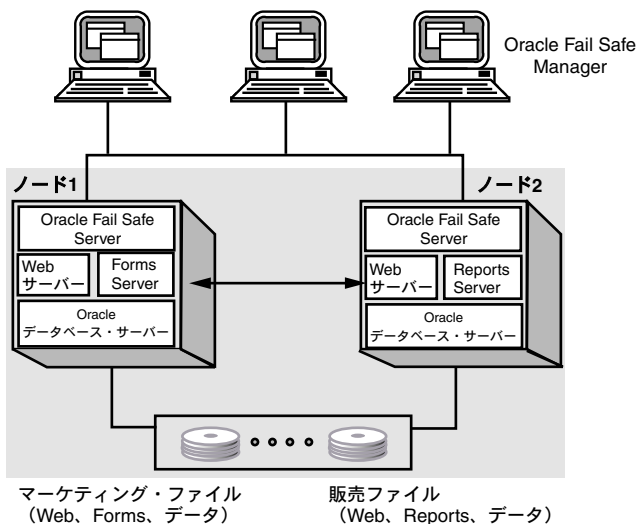
パッシブのスタンバイ・ノードにはそれ自体の作業負荷がないため、アクティブ / パッシブ構成は最高速のフェイルオーバー構成です。

3.1.2 アクティブ/アクティブ構成

アクティブ / アクティブ構成は、アクティブ / パッシブ構成よりもコスト効率に優れています。障害発生時には他のノードのバックアップを行う一方で、各ノードがアプリケーション処理作業も共有するためです。1つのノードに障害が発生した場合、別のノードが障害ノードからフェイルオーバーされたアプリケーションやサービスだけでなく、それ自体のアプリケーションおよびサービスも実行できる能力が必要です。この構成では、柔軟なアーキテクチャによって業務上の要求に最も合うように作業負荷を分割できます。

図 3-3 に、各クラスタ・ノードで1つの Oracle データベース・サーバーと、1つ以上の Forms Server、Reports Server および Web サーバーが稼働している 2 ノードのアクティブ / アクティブ構成を示します。図 3-3 では、ノード 1 でマーケティング用に Oracle データベース・サーバーが1つ使用され、ノード 2 で販売用にも1つの Oracle データベース・サーバーが使用されます。ノード 1 が所有するクラスタ・ディスクにはマーケティング・データベース・ファイルが格納され、ノード 2 が所有するクラスタ・ディスクには販売データベース・ファイルが格納されています。

図 3-3 アクティブ/アクティブ構成



アクティブ / アクティブ構成では、通常操作中すべてのノードがアクティブにアプリケーションを処理します。この構成では、すべてのノードが動作している場合に優れたパフォーマンス（より効果的なスループット）を実現しますが、フェイルオーバーは低速で、ノードに障害が発生するとパフォーマンスが低下する可能性があります。また、クライアント接続がすべてのノードに分配されます。

作業負荷のバランス調整とは、各システムの通常の作業負荷の大きさを調整することです。すべてのシステムがほぼ上限の能力で稼働している場合、停止した他のシステムの作業負荷を処理するためのリソースはほとんど使用できず、クライアント・システムではフェイルオーバー中およびフェイルオーバー後の応答時間が大幅に遅くなります。速やかに修復できるリソースや、障害システムに置き換えられるリソースがある場合は、片方のクラスタ・ノードが両方の作業負荷を担当する時間は短くなります。応答が遅くなる時間はより短い方が効率的です。実際、ダウン時間があるよりもアプリケーションの実行速度が遅くなる方がよいという業務もあります。

また、すべてのシステムが（クラスタ中のノード数に応じて）25%～50%より低い能力で稼働する場合、フェイルオーバー後のクライアントへの応答時間は長くなりませんが、通常の状態では1システム全体に相当する分がアイドル状態であり、アクティブ / パッシブ構成とほとんど同じになってしまいます。

Oracle Fail Safe では、このタイプの構成に伴うパフォーマンス上の問題を回避するような構成が可能です。たとえば、次のようなことが可能です。

- ミッション・クリティカルなアプリケーションに関してのみ、フェイルオーバーを可能にします。
- 各ノードごとに異なるデータベース・パラメータ・ファイルを使用して、フェイルオーバーの後はより少ないシステム・リソースを使用するように設定します。
- 各コンポーネント（Oracle データベース・サーバー、Oracle Forms Server、Oracle Reports Server など）を、独自のフェイルオーバー・ポリシーおよびフェイルバック・ポリシーを持つ個別のグループに構成します。

Oracle Fail Safe では各クラスタ・ノードがいくつかの仮想サーバーのホストになるように構成できるため、このことが可能になります。

- Oracle Fail Safe のスクリプト作成サポート（第 5 章で説明している FSCMD コマンドを使用）と、システム監視ツール（Oracle Enterprise Manager など）を併用して、ロード・バランスのためのグループ移行を自動化します。

すべてのノードが物理的に同等である必要はありませんが、十分な能力とメモリー、ディスク・ホスト・アダプタおよびディスク・ドライブを持つサーバーを選択し、忙しい時間帯にフェイルオーバーが発生した場合でも十分なレベルのサービスをサポートできるようにしておくことが賢明です。

3.1.3 作業負荷のパーティション化構成

作業負荷のパーティション化構成は、3.1.2 項で説明したアクティブ / アクティブ構成のバリエーションの 1 つです。作業負荷のパーティション化では、1 つのノードにデータベースを置き、別のノードにアプリケーション・サーバー（Oracle またはサード・パーティー製のアプリケーションなど）を置くことがあります。1 つのノードに障害が発生した場合、別のノードがそれ自身の作業負荷に加えて障害ノードの作業負荷も引き継ぎます。この利点と不利な点は、アクティブ / アクティブ構成での説明と同じです。

図 3-4 に、2 ノードの作業負荷パーティション化構成を示します。この例では、アプリケーションとデータベースの作業負荷を分割し、1 つのノードではデータベース作業を、もう一方のノードではアプリケーション作業をそれぞれ処理します。この構成は、Oracle アプリケーションなどのエンタープライズ・アプリケーションに高い可用性を保証するために使用されます。また、互いに依存しない作業負荷を異なるノード上で実行することにより、スループットを最大にできます。あるノードに障害が発生した場合は、クラスタ・サーバーのもう一方のノードがデータベース・サーバーとアプリケーション・サーバーの両方を担当します。

図 3-4 アクティブ/アクティブの作業負荷のパーティション化構成

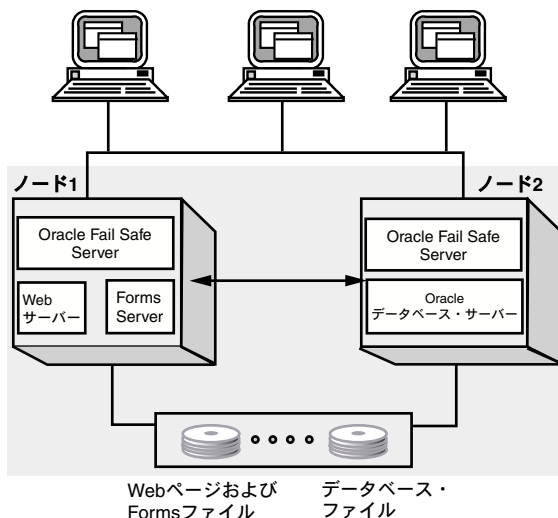


図 3-4 では、ノード 1 で Oracle Forms Server および Web サーバーが稼働し、ノード 2 で Oracle データベース・サーバーが稼働しています。ノード 1 が所有するクラスタ・ディスクには Web ページと Oracle Forms のファイルが格納され、ノード 2 が所有するクラスタ・ディスクにはデータベース・ファイルが格納されています。

プライベート・インターコネクト（ハートビート）が広い帯域幅を持っている場合、Forms Server では、データベースとの通信にパブリック・ネットワークよりもプライベート・ネットワークを使用して、データベース・トランザクションの処理を最適化できる可能性があります。ノード間通信に必要な帯域幅は小さいため、Web サーバーはプライベート・ネットワークをデータベースへの効率的な専用ネットワーク・リンクとして利用できます。

3.1.4 階層化構成

階層化構成では、可用性の高いリソースは別々のクラスタに常駐します。たとえば、複数のシステム上で稼働する多数の Forms Server および Reports Server を駆動する 1 つのバックエンド・データベースを構成することができます。次に、Oracle Forms Load Balancer Server およびマスター Oracle Reports Server を使用したこのような構成方法を示します。

- Oracle Forms Load Balancer Server

Oracle Forms Load Balancer Server は、複数の Oracle Forms Server 間での作業負荷の分割を担当します。階層化構成では Oracle Forms Load Balancer Server と Web サーバーの両方が障害発生箇所となり、それらを正常化するためには通常ユーザーの介入が必要になります。Oracle Forms Load Balancer Server と Web サーバーを、Oracle Fail

Safe により可用性を高めるように構成することで、どちらも障害発生箇所でなくなります。

Oracle Fail Safe を使用して、Oracle Forms Server がアクセスするデータベースをグループに追加すると、可用性をさらに高めることができます。

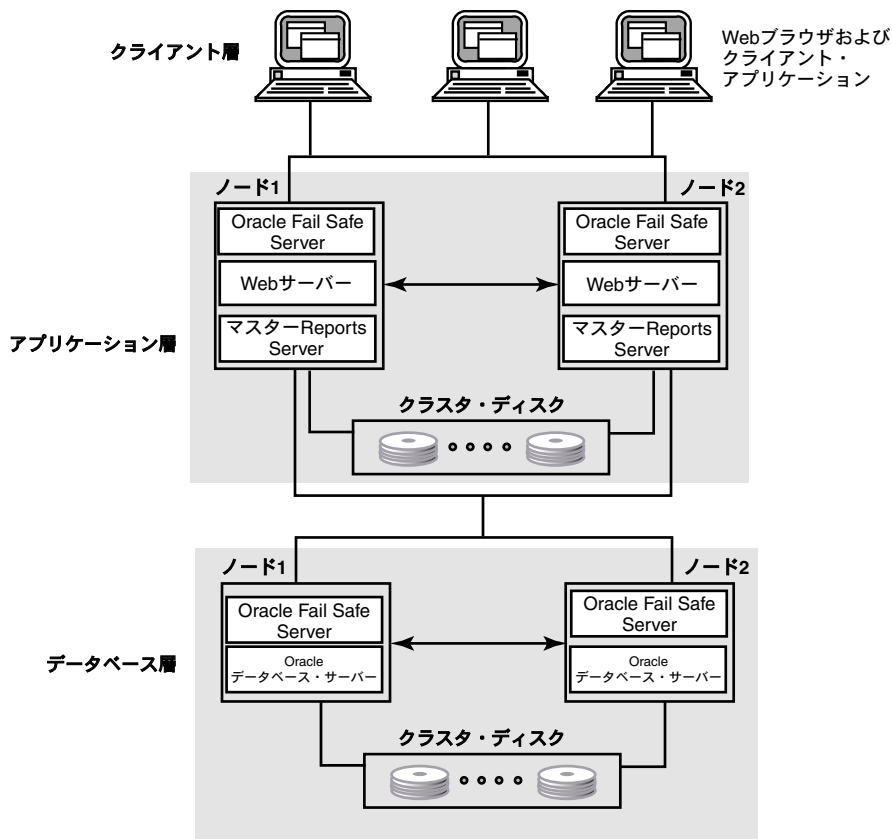
- マスター Oracle Reports Server

サーバーの 1 つがマスターに指定され、それ以外のサーバー間の作業負荷分割を担当します。階層化構成ではマスター Oracle Reports Server と Web サーバーの両方が障害発生箇所となり、それらを正常化するためには通常ユーザーの介入が必要になります。マスター Oracle Reports Server および Web サーバーを、Oracle Fail Safe により可用性を高めるように構成することで、どちらも障害発生箇所でなくなります。

Oracle Fail Safe を使用して、Oracle Reports Server がアクセスするデータベースをグループに追加すると、可用性をさらに高めることができます。

図 3-5 に、Oracle Fail Safe によってマスター Oracle Reports Server の可用性を高める 3 層の構成を示します。データベース層では、各クラスター・ノードに Oracle データベース・サーバーおよび Oracle Fail Safe Server がインストールされています。アプリケーション層では、各クラスター・ノードに Oracle Fail Safe Server、Web サーバーおよびマスター Reports Server がインストールされています。クライアント層は、クライアント・アプリケーションおよび Web サーバーから構成されます。

図 3-5 Oracle Reports Server の階層化構成



階層化構成によって、次のようなことも可能になります。

- 複数のクラスタおよびプラットフォームがともに動作できるような、柔軟なアーキテクチャ。たとえば、次のようになります。
 - Windows プラットフォーム上で Oracle Fail Safe によって可用性を高めた Oracle Forms Server および Oracle Reports Server が、UNIX システムで稼働しているバックエンドの Oracle Parallel Server データベースにアクセスできます。
 - アプリケーション層のサーバーが、別のプラットフォーム上で実行できます。したがって、UNIX マシンで稼働している既存の Oracle Reports Server に、Oracle Fail Safe を実行する Microsoft クラスタを1つ追加するだけで、マスター Oracle Reports Server の（それによって Reports 層全体の）可用性を高めることができます。

- ビジネス・ソリューションに対する高可用性の増分的な展開。旧来のバックエンド・データベース・システムを変更せずに、信頼性が低い中間層のアプリケーション・サーバーから高い可用性を追加していくことによって、これが可能になります。

3.2 クライアントとアプリケーションの統合

クライアント・アプリケーションを Oracle Fail Safe 環境で動作させる場合、特別なプログラミングや変更は必要ありません。シングル・ノードの Oracle リソースで動作するクライアント・アプリケーションであれば、再コーディングまたは再コンパイル、再リンクをすることなく Oracle Fail Safe 環境で正常に機能し続けます。これは、クライアントが仮想サーバーを利用してアプリケーションにアクセスできるためです。

第 III 部の各章には、クライアントとアプリケーションの統合方法に特化した項があります。リソースがクラスタ内の他のノードにフェイルオーバーするときに、クライアントおよびアプリケーションを透過的にフェイルオーバーさせるためには、第 III 部の該当する章を参照してください。

第 II 部

管理と総合管理

第 II 部では、Oracle Fail Safe の管理と総合管理について説明します。

第 II 部は、次の章から構成されています。

- 第 4 章「高い可用性を実現するための管理」
- 第 5 章「FSCMD コマンドライン・インタフェース」
- 第 6 章「トラブルシューティング・ツール」

高い可用性を実現するための管理

Oracle Fail Safe には、クラスタ環境内でリソースを容易に構成できるという利点があります。この章では、次の項目について説明します。

項目	参照
フェイルオーバーを構成する意味	4.1 項
ウィザードの入力項目の Oracle Fail Safe での処理	4.2 項
クラスタ・セキュリティの管理	4.3 項
スタンドアロン・リソースの検出	4.4 項
リソース名の変更	4.5 項
複数の Oracle ホーム環境での Oracle Fail Safe の使用方法	4.6 項
複数仮想アドレスを使用する構成	4.7 項
既存クラスタへのノードの追加	4.8 項

スタンドアロン・リソースをグループに構成するための段階的な手順は、このマニュアルの第 III 部と、Oracle Fail Safe Manager のチュートリアルおよびオンライン・ヘルプを参照してください。

4.1 フェイルオーバーを構成する意味

クラスタ内のフェイルオーバーの構成には多くのハードウェア・コンポーネントおよびソフトウェア・コンポーネントが関係しているため、複雑なプロセスになる場合があります。Oracle Fail Safe Manager のウィザードを使用すると、ネットワーク管理者は、最小限の作業で簡単かつ自動的にフェイルオーバーを構成できます。Oracle Fail Safe Manager を使用すると、クラスタ内の 1 つのノードに障害が起きた場合でも別のクラスタ・ノードが障害ノード

のグループ内のリソースを即時に引き継ぐように、リソースをグループごとに構成できます。

ウィザードを使用すると、実装時における構成の問題を最小限に抑えることができます。また、高度な専門知識がなくても、可用性が高まるようにリソースを構成できます。ウィザードで設定したポリシーの大部分は、Oracle Fail Safe Manager を使用して後から変更できます。

次のリストに、リソースにフェイルオーバーを実装するための、基本的な作業を要約します。最初の作業以外はすべて、Oracle Fail Safe Manager を使用して実行します。

1. Oracle Fail Safe とともに構成する製品が適切にインストールされているかどうかの確認（これは『Oracle Fail Safe インストレーション・ガイド』で説明しています。）
2. Oracle Fail Safe Manager の起動
3. クラスタの検証
4. グループの作成
5. グループに 1 つ以上の仮想アドレスを追加
6. スタンドアロンの Oracle データベース・サーバーを追加する場合は、「スタンドアロン・データベースの検証」ツールを使用してデータベースを検証
7. グループへのリソースの追加
8. グループの検証
9. クライアント・システムに Net8（または SQL*Net）ファイル（たとえば TNSNAMES.ORA ファイル）がある場合は、これらのファイルを更新

このリストには、基本作業のみが要約されています。構成するリソースのタイプによっては、これ以外に手順や考慮事項を追加する場合もあります。

Oracle Fail Safe Manager のウィザードを使用するための段階的なガイダンスは、Oracle Fail Safe Manager のチュートリアルおよびオンライン・ヘルプを参照してください。

4.2 ウィザードの入力項目の Oracle Fail Safe での処理

ウィザードで必要な情報がすべて収集されると、Oracle Fail Safe Manager が Oracle Fail Safe Server と対話し（次にそれが MSCS と対話して）、可用性の高い環境が実現されます。

Oracle Fail Safe では、ユーザーがウィザードで入力した情報に基づいて、環境の構成に必要な追加情報がすべて導出されます。

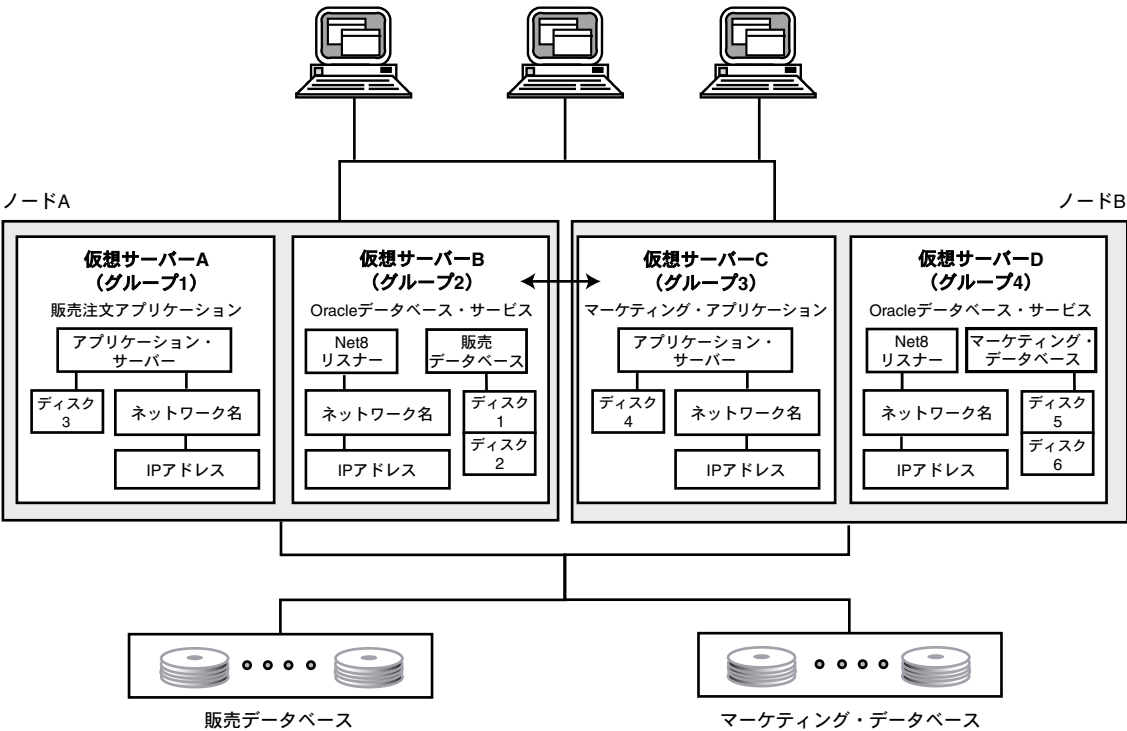
同様の手順により、多くのリソースが Oracle Fail Safe によって構成されます。次のリストで、可用性の高い Oracle データベースを構成するために Oracle Fail Safe により実行される特定の手順を説明します。

1. 仮想アドレスを使用したデータベースへのアクセスを構成します。
 - a. Net8（または SQL*Net）を、データベースの「可能所有者ノード」リストにあるすべてのノードで仮想アドレスを使用できるように構成します。（2つのノードから成るクラスタでは、両方のクラスタ・ノードが可能所有者ノードになります。3つ以上のノードから成るクラスタでは、「リソースをグループに追加」ウィザードのステップとして、リソースの可能所有者ノードを指定するように求められます。）
 - b. ネットワーク構成情報を、「可能所有者ノード」リストのすべてのノードに複製します。
2. 次のようにデータベースを構成します。
 - a. データベース・リソースが使用するすべてのデータ・ファイルがクラスタ・ディスクに置かれ、他のグループのアプリケーションによって現在使用されていないことを確認します。クラスタ・ディスクが別のグループ内に存在していても、そのグループ内のアプリケーションによって使用されていない場合、Oracle Fail Safe がそのディスクをデータベース・リソースと同じグループに移動します。
 - b. ウィザードで選択した内容に基づいて、データベース・リソースのフェイルバック・ポリシーを作成します。
 - c. 次のリソースをグループに移入します。
 - * クラスタ・グループに使用される各ディスク・リソース
 - * Oracle データベース・サーバー
 - * Net8（または SQL*Net）リスナー
3. データベースを追加するグループの可能所有者ノードのそれぞれについて次の処理を、一度に1つずつ行います。
 - a. 同じ名前の Oracle インスタンスをノード上に作成します。
 - b. データベースをノードにフェイルオーバーして、ノードがデータベースをオンライン化またはオフライン化できることを検証し、フェイルオーバー・ポリシーが機能することを確認します。
4. 「可能所有者ノード」リストのすべてのノードについてフェイルオーバーをテストした後、Oracle データベースを停止し、優先ノードで再度オンライン化します。

これらの手順を実行することで、Oracle Fail Safe では、リソースが正しく構成され、またリソース追加先のグループのすべての可能所有者ノードにフェイルオーバーおよびフェイルバックできるようになります。

図 4-1 に、各ノードがアプリケーション・サーバーを伴うグループとデータベース・サーバーを伴うグループのホストになる、2 ノードのアクティブ / アクティブ・クラスタ構成を示します。

図 4-1 Oracle Fail Safe 環境内の仮想サーバーとアドレッシング



仮想サーバー（A、B、C および D）とネットワーク・アドレスは、すべてのクライアントおよびクラスタ・ノードには既知のものです。各クラスタ・ノードの LISTENER.ORA ファイルと各クライアント・ワークステーションの TNSNAMES.ORA ファイルには、各仮想サーバーの名前とアドレス情報があります。

フェイルオーバーが適切に動作するためには、各 TNSNAMES.ORA ファイルおよび LISTENER.ORA ファイル内のホスト名（仮想アドレス）、データベース・インスタンス、SID 項目およびプロトコル情報が、グループ内のリソースの可能所有者となる各サーバー・ノードと、クライアント・マシンとで一致する必要があります。

たとえば、通常の動作中には、ノード A で仮想サーバー B がアクティブになります。ノード B は仮想サーバー B のフェイルオーバー・ノードです。クラスタ・ディスクは両方のノードと接続しているため、リソースはクラスタのどちらのノードでも稼働しますが、各グループのリソースに対するサービスは、一度に 1 つのクラスタ・ノードによって提供されます。

ノード A でシステム障害が発生すると、グループ 1 および 2 はノード A 上にあったときと同じ仮想アドレスおよびポート番号を使用してノード B でアクティブになります。ノード B はクライアントに対し透過的に、ノード A から作業負荷を引き継ぎ、クライアントは仮想

サーバー A を使用してグループ 1 に、仮想サーバー B を使用してグループ 2 にアクセスを試み続けます。クライアントは、グループを処理している物理ノードとは無関係に、同じ仮想サーバー名およびアドレスを使用してそのグループ内のリソースにアクセスを続けます。

4.3 クラスタ・セキュリティの管理

Oracle Fail Safe に関連した管理作業を実行するには、Oracle リソースおよびアプリケーションを管理し、Oracle Fail Safe Manager による操作を実行する特別な権限が必要です。

表 4-1 に、Oracle Fail Safe 環境で使用するサービスに必要な権限の一覧を示します。詳細は、右端の列に示した項を参照してください。

表 4-1 許可と権限

サービス	必要な権限	参照
Oracle Fail Safe Server	すべてのクラスタ・ノードで管理者権限を持つドメイン・ユーザー・アカウント	4.3.1 項
Oracle Fail Safe Manager	すべてのクラスタ・ノードで管理者権限を持つドメイン・ユーザー・アカウント	4.3.2 項
Oracle データベース	SYSDBA 権限を伴うデータベース管理者アカウント	7.5 項
Oracle Forms Server、 Oracle Forms Load Balancer Server および Oracle Reports Server	権限は不要（Oracle Reports Server には、プリンタにアクセスできる Windows ドメイン・ユーザー・アカウントが必要）	9.4 項 、 10.4 項 および 11.3.3.6 項
Oracle Application Server	Oracle Application Server は、Oracle Fail Safe Server と同じ Windows システム・アカウントでの実行が必要	12.6 項
Oracle Applications Concurrent Manager	Oracle Applications Concurrent Manager は、管理者アカウントでの実行が必要	13.4 項
Oracle HTTP Server	なし	適用なし
Oracle WebDB リスナー	なし	適用なし

表 4-1 許可と権限 (続き)

サービス	必要な権限	参照
Oracle MTS Service	<p>Oracle MTS Service がシステムにログオンするため使用するアカウントは、システムまたはドメイン・ユーザー・アカウントである必要があります。</p> <p>Oracle MTS Service がデータベースに接続するため使用するアカウントには、次のデータベース・ロール、権限および権利が必要です。</p> <ul style="list-style-type: none">■ CONNECT、RESOURCE および SELECT_CATALOG_ROLE の各データベース・ロール■ FORCE ANY TRANSACTION、CREATE PUBLIC SYNONYM および DROP PUBLIC SYNONYM の各データベース権限■ Oracle データベース内の Oracle MTS Service 表に対する SELECT、INSERT、UPDATE および DELETE の各権限	14.4 項

4.3.1 Oracle Fail Safe Server

Oracle Fail Safe では、適切な権限を持ったユーザー以外がクラスタ内のリソースを管理できないように、セキュリティ・コンポーネントが実装されます。

Oracle Fail Safe Server は、(システム・アカウントではなく) すべてのクラスタ・ノードに対する管理者権限を持ったドメイン・ユーザー・アカウントのもと以外では稼働しない Windows サービスとして稼働します。この Oracle Fail Safe Server のユーザー・アカウントは、Oracle Fail Safe をインストールするときに指定します。(インストール時の詳細は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。)

Oracle Fail Safe には、それ自身のセキュリティ・コンポーネントも備わっています。したがって、Oracle Fail Safe Server で使用している Windows ユーザー・アカウント (ユーザー名、パスワードまたはドメイン) を変更する場合は、Windows サービスと Oracle Fail Safe の両方のセキュリティ設定を更新する必要があります。Oracle Fail Safe には、このセキュリティ情報を更新するための Security Setup ツールがあります。

4.3.1.1 Oracle Fail Safe Security Setup ツールを使用したアカウント更新

Oracle Fail Safe には Security Setup ツールがあり、これを使用して Oracle Fail Safe が稼働するアカウントの情報を更新できます。Oracle Fail Safe Server のインストール時に、Oracle Fail Safe Security Setup ツールもインストールされます。

Windows のタスクバーから Oracle Fail Safe Security Setup ツールにアクセスするには、クラスタ・ノードから次のように選択します。

「スタート」→「プログラム」→「<Oracle_Home>」→「Oracle Fail Safe Security Setup」

注意： すべてのクラスタ・ノードのセキュリティ情報の更新に、必ず Oracle Fail Safe Security Setup ツールを使用し、すべてのクラスタ・ノードで同じアカウントを使用してください。

図 4-2 に、ドメイン FCLUSTER にユーザー・アカウント fsuser を設定する例を示します。

図 4-2 Oracle Fail Safe Server の Windows ユーザー・アカウント設定



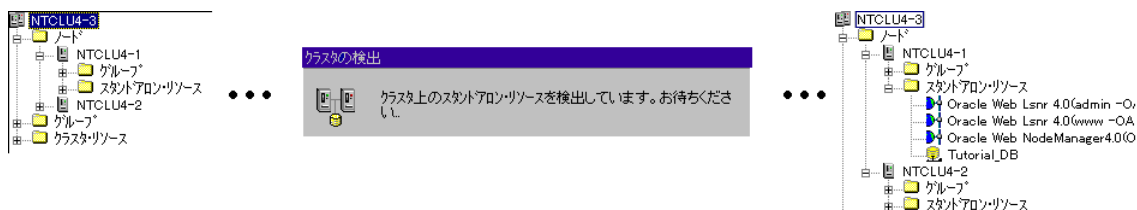
4.3.2 Oracle Fail Safe Manager

Oracle Fail Safe Manager へのログインに使用するアカウントは、（ローカル・アカウントではなく）すべてのクラスタ・ノードで管理者権限を持ったドメイン・ユーザー・アカウントである必要があります。

4.4 スタンドアロン・リソースの検出

Oracle Fail Safe Server では、Oracle Fail Safe Manager ツリー・ビュー内のスタンドアロン・リソースが自動的に検出（特定）および表示されます。図 4-3 に、Oracle Fail Safe Manager 内での検出例を示します。第 III 部の各章には、Oracle Fail Safe で可用性が高まるように構成することが可能な各タイプのコンポーネントの検出方法についての情報が記載されています。

図 4-3 スタンドアロン・リソースの検出



4.5 リソース名の変更

リソースがグループに追加された後は、リソース名を変更しないでください。リソース名を変更する必要がある場合は、Oracle Fail Safe Manager を使用してリソースをグループから削除します。その後で、新しい名前を使用してグループに再度追加します。

4.6 複数の Oracle ホーム環境での Oracle Fail Safe の使用方法

Oracle Fail Safe では、複数 Oracle ホームの機能をサポートします（複数 Oracle ホームは、リリース 8.0.4 の Oracle8 から使用可能です）。次のリストに、複数 Oracle ホーム環境で Oracle Fail Safe を使用するための要件を示します。

- Oracle Fail Safe Server は、すべてのクラスタ・ノード上のいずれか 1 つの Oracle ホームにインストールできます。ノードにインストールして実行できるのは、Oracle Fail Safe Server の 1 つのバージョンに限られます。
- Oracle Fail Safe Manager は、1 つのシステムに複数のバージョンをインストールできます。複数バージョンの Oracle Fail Safe Manager を 1 つのシステムにインストールする方法を選択すると、1 つのシステムから、（異なるバージョンの Oracle Fail Safe Server が稼働している）複数のクラスタを管理できます。

注意： 複数バージョンの Oracle Fail Safe Manager をインストールする場合、各バージョンは別々の Oracle ホームにインストールする必要があり、リリース 3.1 の Oracle Fail Safe Manager は最後にインストールしてください。

- 可用性が高まるように構成する各リソースは、可能所有者であるすべてのクラスター・ノード上の同じ Oracle ホームにインストールします。「クラスターの検証」操作を行うと、この対称性が検証されます。「クラスターの検証」操作の詳細は、[6.1.1 項](#)を参照してください。
- グループのすべてのデータベースは、同じ Oracle ホームに属している必要があります。
データベースをグループに追加する際に、Net8（または SQL*Net）のリスナー・リソースもグループに追加します。（オプションで、Oracle Intelligent Agent リソースをグループに追加できます。詳細は、[7.7.1 項](#)を参照してください。）
リスナーは、データベースが常駐しているのと同じ Oracle ホームに作成されます。Oracle Intelligent Agent は、データベースが常駐しているのと同じ Oracle ホームにある必要はありません。

4.7 複数仮想アドレスを使用する構成

Oracle Fail Safe Manager を使用してグループにリソース（汎用サービス、Oracle Forms Server 以外）を追加するには、あらかじめそのグループに 1 つ以上の仮想アドレスを追加しておく必要があります。クライアント・アプリケーションは、グループの仮想アドレスの 1 つによって、そのグループ内のリソースに接続します。

「リソースをグループに追加」ウィザードを起動して、グループには（リソース追加の前に）最大で 32 個までの仮想サーバーを追加できます。（「リソース」→「グループに追加」を選択します。）

次の制限事項に注意してください。

- グループに少なくとも 1 つの仮想アドレスを追加していないと、そのグループに別のリソースは追加できません。仮想アドレスが含まれていないグループには、汎用サービスと Oracle Forms Server 以外は追加できません。
- グループに 1 つ以上の Oracle データベースが含まれることになる場合、次のように処理します。
 - グループにデータベースを追加する前に、そのグループの 1 つ以上のデータベースとともに構成するすべての仮想アドレスをグループに追加します。
 - グループ内のすべてのデータベースで、グループに追加する最初のデータベースに対して指定する一連の仮想アドレスを使用します。（一連の仮想アドレスには、最低 1 つのアドレスを含めることができます。）

(Oracle データベースと複数仮想アドレスの構成に関する詳細は、[7.2.3.2 項](#)を参照してください。)

グループに仮想アドレスを追加すると、どのクラスタ・ノードがそのクラスタのホストになっているかに関係なく、同じネットワーク・アドレスを持つクライアントはそのグループにアクセスできます。

グループ内の複数仮想アドレスによって、柔軟な構成オプションが得られます。たとえば、プライベート・ネットワークを介してデータベースのバックアップ処理を実行している間に、ユーザーがパブリック・ネットワークを介してデータベースにアクセスすることが可能になります。あるいは、異なるネットワーク・セグメント上の異なる仮想アドレスをセキュリティ管理に割り当て、管理者があるセグメント上のデータベースにアクセスする一方で、ユーザーが別のセグメント上のデータベースにアクセスすることも可能です。

グループに複数の仮想アドレスを追加する場合、Oracle Fail Safe Manager により、クライアントがどのアドレスを使用してそのグループのリソースにアクセスできるかを指定するように要求されます。グループに複数のリソース（たとえば、データベースと Oracle Reports Server など）を追加すると、ユーザーからデータベースに直接アクセスする場合はデータベースへのアクセス専用の仮想アドレスを使用し、ユーザーから Oracle Reports Server にアクセスする場合は別の仮想アドレスを使用することができます。また、データベース・ユーザーが多数の場合、一部のユーザーにはある仮想アドレスを使用してデータベースにアクセスさせ、その他のユーザーにはもう 1 つ別の仮想アドレスを使用させることで、ネットワークのトラフィックを均衡化することが可能になります。

グループへの仮想アドレスの追加の詳細は、Oracle Fail Safe Manager オンライン・ヘルプを参照してください。

4.8 既存クラスタへのノードの追加

既存のクラスタに新規のノードを追加するソフトウェアのインストールについては、『Oracle Fail Safe インストレーション・ガイド』で説明しています。インストール作業の完了後に、最終的な手順があります。新規のノードが可能所有者となるクラスタの各グループに対して、「グループの検証」コマンドを実行する必要があります。

クラスタに新規ノードを追加し、Oracle Fail Safe と、そのノードで実行を予定しているリソースの DLL を、そのノードにインストールするとします。新規のノードは、これらのリソースの可能所有者になります。これらのリソースが新規ノード上で実行するように構成されていない場合、そのリソースを含むグループ（複数も可）が新規のノードにフェイルオーバーしても、そのノードではこれらのリソースは再起動しません。

しかし「グループの検証」コマンドを実行すると、検証されたグループ内のリソースが、そのグループの可能所有者となっている各ノードで実行するよう構成されているかどうか、Oracle Fail Safe によってチェックされます。グループ内のリソースが実行するよう構成されていない可能所有者ノードが検出されると、Oracle Fail Safe がそれを構成します。

このような理由からオラクル社では、新規のノードが可能所有者としてリストされた各グループについては、「グループの検証」コマンドを実行することを強くお勧めします。「グループの検証」コマンドは、[6.1.2 項](#)で説明します。グループの検証には、[第 5 章](#)で説明しているように、FSCMD コマンドも使用できます。

FSCMD コマンドライン・インタフェース

Oracle Fail Safe では、クラスタ内のリソースを管理する方法として、Oracle Fail Safe Manager のかわりにコマンドライン・インタフェースが提供されます。たとえば、「コマンド プロンプト」で FSCMD コマンドを使用して、Oracle リソースをオフライン化、またはオンライン化できます。FSCMD コマンドは、バッチ・プログラムやスクリプトから Oracle リソースを管理する場合に便利です。

FSCMD

Oracle Fail Safe Manager を使用して実行する操作のうちの多くは、FSCMD コマンドによって実行できます。FSCMD コマンドを実行するには、次のように、Oracle Fail Safe Manager がインストールされている場所から始まるフルパスを指定してください。

```
<Oracle_Home>%fs%fsmgr%bin%fscmd.exe
```

このパスを使用しない場合、Windows オペレーティング・システムでは FSCMD コマンドを検出できません。

注意： システムにインストールされている他の Oracle 製品と競合する可能性があるため、システム Path 環境変数にディレクトリ `<Oracle_Home>%fs%fsmgr%bin` を含めないでください。

書式

FSCMD コマンドを使用するときは、「コマンド プロンプト」ウィンドウを開いて、次の構文を使用した FSCMD コマンドラインを入力します。

```
FSCMD <action> <name> /CLUSTER=cluster_name [<qualifier>]
```

注意： わかりやすくするために、この章では構文および例から FSCMD へのフルパス表記を省略します。

説明

Oracle Fail Safe には、Oracle Fail Safe Manager で実行可能な機能の多くをスクリプトで実行できる FSCMD コマンドがあります。たとえば、FSCMD コマンドを使用して、夜間バックアップを実行する前にクラスタ・リソースをオフライン化し、バックアップの完了後にリソースを再びオンライン化できます。

FSCMD コマンドは、ローカル・ノード、リモート・ノードのどちらでも実行できます。たとえば、次の場所で FSCMD コマンドを実行します。

- クラスタ・ノード
- クラスタ・ノード以外のシステム
- Windows 95、Windows 98、Windows NT または Windows 2000 のクライアント

FSCMD コマンドを実行する際、Oracle Fail Safe Manager の「イベント状態の概要」ウィンドウで、長時間実行する操作の結果を表示することが可能です。また、長時間実行する操作の結果を取得するログ・ファイルの名前を指定できます。

コマンド・パラメータ

action

グループ、リソースまたはクラスタに適用するアクションを指定します。次の表で説明するアクションのいずれかを使用します。

アクション	説明
DUMPCLUSTER	クラスタ構成情報の出力。この操作による出力は、/LOGFILE 修飾子によって指定したファイル名のファイルに書き込まれます。
MOVEGROUP	リソースのグループを /NODE コマンド修飾子によって指定したノードに移動。この操作による出力は、/LOGFILE 修飾子によって指定したファイル名のファイルに書き込まれます。
ONLINEGROUP	グループのオンライン化。
ONLINERESOURCE	リソースのオンライン化。
OFFLINEGROUP	グループのオフライン化。
OFFLINERESOURCE	リソースのオフライン化。Oracle データベース・リソースでは、この操作には /OFFLINE コマンド修飾子が必要です。
VERIFYGROUP	Oracle Fail Safe によって構成された単一グループの検証。この操作による出力は、/LOGFILE 修飾子によって指定したファイル名のファイルに書き込まれます。
VERIFYALLGROUPS	Oracle Fail Safe によって構成されたすべてのグループの検証。この操作による出力は、/LOGFILE 修飾子によって指定したファイル名のファイルに書き込まれます。
VERIFYCLUSTER	クラスタ構成の検証。この操作による出力は、/LOGFILE 修飾子によって指定したファイル名のファイルに書き込まれます。

action パラメータは、FSCMD コマンドに対する第 1 引数としてください。

name

FSCMD コマンドで処理するリソースまたはグループの名前です。たとえば、PERSONNEL.world は、データベース・リソースとして有効な名前です。

このパラメータは、action パラメータの次に指定します。name パラメータは、DUMPCLUSTER、VERIFYALLGROUPS および VERIFYCLUSTER を除くすべての FSCMD コマンド・アクションに必要です。

コマンド修飾子

/CLUSTER=cluster-name

FSCMD コマンドを実行するクラスタの名前を指定します。

この修飾子は必須です。

/LOGFILE=file-name

DUMPCLUSTER、MOVEGROUP、VERIFYCLUSTER、VERIFYGROUP または VERIFYALLGROUPS アクションの実行時に、長時間実行する操作によって作成されるログ・ファイルの場所を指定します。LOGFILE 修飾子を指定しない場合、ログ・ファイルは現行の出力デバイス（一般的にはシステム・コンソール）に書き込まれます。

この修飾子はオプションです。

/NODE=node-name

この修飾子は、MOVEGRUOP アクションでのグループの移動先ノード名を指定するときに限り使用します。

修飾子は、MOVEGROUP アクションを指定する場合に必要となります。

/OFFLINE=offline-option

この修飾子は、OFFLINERESOURCE アクションを指定して、Oracle リソースをオフラインにする場合に限り使用します。Oracle データベースでは、次の表にあげたオフライン・オプション・モードの1つを指定しないと、そのリソースは immediate モード（デフォルト）でオフライン化されます。

モード	説明
abort	<p>データベース・インスタンスを異常終了させることにより、データベースを即時に停止します。abort モードによる停止は、次の場合を除いてできるだけ使用を避けてください。</p> <ul style="list-style-type: none">■ データベースまたはアプリケーションの機能のうちの1つが不正であり、かつ immediate、normal のいずれの停止モードも作動しない場合■ データベースをただちに停止する必要がある（たとえば、電源遮断が1分以内に発生することがわかっている）場合 <p>データベースの再起動時に、データベース・リカバリ・プロシージャが実行されます。</p>

モード	説明
immediate	<p>処理中の SQL 文を終了し、コミットされていないトランザクションをロールバックし、ユーザーとのデータベース接続を切断して、データベースを即時に停止します。immediate モードは、リソースをオフライン化するときのデフォルトのモードです。immediate モードの選択は、次の場合を除いてできるだけ避けてください。</p> <ul style="list-style-type: none"> ■ 電源遮断がまもなく発生する場合 ■ データベースまたはアプリケーションの 1 つが不正に機能している場合 ■ データベースのバックアップ処理を実行する場合
normal	<p>次のようにしてデータベースを停止します。</p> <ul style="list-style-type: none"> ■ 「データベースの停止」コマンド発行後に新規に接続できないようにします。 ■ データベースを実際に停止する前に、接続中のすべてのユーザーが切断するまで待機します。
transactional	<p>現行のトランザクションがすべて完了した後で、データベースを停止します。</p>

/DOMAIN=domain-name

(/USER 修飾子で指定した) ユーザー・アカウントが存在するドメインを指定します。

この修飾子は Windows 2000 クライアントを除くすべてのクライアントでオプションですが、ドメイン名を入力する場合は、ユーザー名およびパスワードも入力する必要があります。

/PWD=password

/USER 修飾子で指定したアカウントのパスワードを指定します。

この修飾子は Windows 2000 クライアントを除くすべてのクライアントでオプションですが、パスワードを入力する場合は、ユーザー名およびドメイン名も入力する必要があります。

/USER=username

すべてのクラスタ・ノードで管理者権限を持つドメイン・アカウントのユーザー名を指定します。

この修飾子は Windows 2000 クライアントを除くすべてのクライアントでオプションですが、ユーザー名を入力する場合は、パスワードおよびドメイン名も入力する必要があります。

使用上の注意

- action パラメータは、FSCMD コマンドに対する第 1 引数としてください。
- ドメイン名、ユーザー名およびパスワードの修飾子は、Oracle Fail Safe Server に接続するためのアカウント情報となります。これらの修飾子のうち 1 つでも入力されない修飾子があると、構文エラーが発生します。アカウント情報が提供されない場合、サーバーでは、FSCMD コマンドの起動時にログインしたユーザーを認証します。現在のアカウントが Oracle Fail Safe 環境で操作を実行するのに必要な権限を持たない場合、エラー・メッセージが生成されます。
- パーサーでは、コマンド修飾子間のデリミタとしてスラッシュ (/) またはハイフン (-) が認められます。
- コマンド・パラメータおよび修飾子では大文字と小文字が区別されないため、大文字か小文字のいずれか、または両方を組み合わせて使用できます。
- グループ名またはリソース名に 1 つ以上の空白が含まれる場合は、引用符で囲み、たとえば、"Sales Group" のようにします。
- コマンドラインでは、修飾子の間に空白を入力する必要があります。次に、コマンドラインにおける空白の正しい使用例と誤った使用例を示します。

正しい使用例：

```
FSCMD onlineresource salesdb.world /CLUSTER=ntclu-160 /USER=smith /PWD=smithpwd /DOMAIN=newengland
```

誤用例：

```
FSCMD onlineresource  
salesdb.world/CLUSTER=ntclu-160/USER=smith/PWD=smithpwd/DOMAIN=newengland  
Invalid number of parameters.
```

- Windows イベント ビューアを使用して、FSCMD コマンドの処理中に報告されていたイベントを表示します。

コマンドの例

例 1

次のコマンドでは、salesdb.world という Oracle データベースをオンライン化します。

```
FSCMD onlineresource salesdb.world /CLUSTER=ntclu-160 /USER=smith /PWD=smithpwd
/DOMAIN=newengland
```

例 2

次のコマンドでは、クラスタ NTCLU-160 上のすべてのグループが検証され、検証操作の出力がログ・ファイル C:\temp\fsverify.log に書き込まれます。

```
FSCMD verifyallgroups /LOGFILE=c:\temp\fsverify.log /CLUSTER=ntclu-160 /USER=smith
/PWD=smithpwd /DOMAIN=ORANT
```

例 3

次のコマンドでは、Oracle データベースを即時にオフライン化します。

```
FSCMD offlineresource salesdb.world /CLUSTER=NTCLU-160 /USER=smith /PWD=smithpwd
/DOMAIN=ORANT /OFFLINE=immediate
```

例 4

次のコマンドでは、Disk Group 1 というグループをオフライン化します。

```
FSCMD offlinegroup "Disk Group 1" /CLUSTER=ntclu-160
```

スクリプトの例

次のスクリプトでは、db.world データベースをバックアップします。この例では、そのデータベースが FS Group1 というグループに含まれ、ノード NTCLU-141 上の NTCLU-140 というクラスタで稼働しているものとします。

```
REM   This script shows an example of performing a backup operation on an Oracle
REM   Fail Safe database.
REM
REM   1. Move the group FS Group1 that contains the database to the node on
REM   which the backup operation will run. Alternatively, you can create file
REM   share resources for each cluster disk to allow the backup software to
REM   access the drives through a virtual server address regardless of which
REM   cluster node currently owns them.

fscmd movegroup "FS Group1" /node=NTCLU-141 /cluster=NTCLU-140

REM   2. Take the group offline.

fscmd offlineresource db.world /offline=abort /cluster=NTCLU-140
```

```
REM 3. Begin the backup operation here. As an example, the following lines
REM copy files using the copy function on the operating system.

copy e:\ofbdb\ofsl\data\*.ora e:\backup\data
copy e:\ofbdb\ofsl\log\*.ora e:\backup\log
copy e:\ofbdb\ofsl\param\*.ora e:\backup\param

REM 4. Place the resource back online.

fscmd onlineresource db.world /cluster=ntclu-140

REM The backup operation is complete.

REM The following section is necessary between steps 1 and 2 if an Oracle
REM Intelligent Agent is being used and if the agent must be notified about
REM the database shutdown.

net start OracleServiceOFS1

echo connect internal/i@db.world > svrmgr.txt
echo startup pfile=e:\ofbdb\ofsl\param\initofsl.ora >> svrmgr.txt
echo shutdown immediate >> svrmgr.txt
echo exit >> svrmgr.txt

svrmgr30 < svrmgr.txt
```


トラブルシューティング・ツール

この章では、Oracle Fail Safe Manager に搭載されたトラブルシューティング・ツールに関する一般情報を記載します。次の表に、この章で説明される情報を示します。

項目	参照
検証操作	6.1 項
クラスタのダンプ	6.2 項
その他のトラブルシューティング情報の検索	6.3 項

Oracle Fail Safe には一元化されたメッセージ機能が備わっています。ある操作を実行した結果、エラーが発生すると、システムはエラーに対応するメッセージを検索し、そのメッセージを表示します。これらのメッセージの詳細は、『Oracle Fail Safe エラー・メッセージ』に記載されています。

6.1 検証操作

Oracle Fail Safe には、ノード、グループおよびリソース状況の有効性を検証して、クラスタのコンポーネントおよび環境の有効性を検査する便利なツール・ファミリがあります。不一致や問題が検出されると、検証操作により適切な処置がとられ、実際に発生した問題や潜在的な問題をすべて修正します。

図 6-1 に、「トラブルシューティング」メニューの検証コマンドを示します。

図 6-1 「トラブルシューティング」メニューの検証コマンド

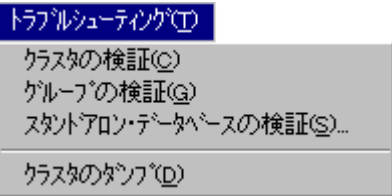


表 6-1 に、それぞれの検証コマンドとその詳細の参照先を示します。

表 6-1 トラブルシューティングのための検証コマンド

ツール	説明	参照
クラスタの検証	Oracle Fail Safe インストール、Oracle 製品インストール (Oracle ホームおよび製品バージョン番号を含む)、クラスタ・ネットワーク構成およびクラスタ・リソース DLL 登録を検証する。	6.1.1 項
グループの検証	グループ・リソースおよびその依存性が正しく構成されているかを検証する。	6.1.2 項
スタンドアロン・データベースの検証	スタンドアロン・データベース・インスタンスの有効性を検査し、別のノードに残っている旧構成情報を削除する。	6.1.3 項

検証コマンドを使用して、いつでも、クラスタ、グループまたはスタンドアロン・データベースの有効性を検査できます。検証中に問題が検出された場合、Oracle Fail Safe によって、問題を修正するようにプロンプトが出されるか、または問題を詳しく説明したエラー・メッセージが返されます。Oracle Fail Safe で変更を行う場合、その前に必ずユーザーの応答が求められます。

6.1.1 クラスタの検証

「クラスタの検証」操作により、クラスタのインストールおよびネットワーク構成の有効性が検査されます。次のように選択することで、いつでもクラスタを検証できます。

「トラブルシューティング」→「クラスタの検証」

Oracle Fail Safe ソフトウェアのインストールまたはアップグレード後、初めてクラスタに接続する際に、「クラスタの検証」操作を実行するように求められます。「クラスタの検証」はいつでも実行できますが、クラスタ構成を変更したときは必ず実行する必要があります。「クラスタの検証」操作により、次のことが検証されます。

- クラスタの少なくとも 2 つのノードで Oracle ホームが対称的であること
Oracle ホームが対称的であるということは、クラスタの少なくとも 2 つのノードでホーム名が同じであることを意味します。ただし、Oracle ホームの実際の場所は同じである必要はありません。
- Oracle Fail Safe Server のリリースがすべてのノード上で同一であること
- リソース・プロバイダ（コンポーネント）が各リソースの少なくとも 2 つの可能所有者であるノードでまったく同じに構成されていること
- ホスト名と IP アドレスのマッピングがクラスタ内のすべてのノード間で矛盾なく解決していること

マッピングの不一致が存在する場合、「クラスタの検証」コマンドはネットワーク・アダプタの順序が正しくないことを示すエラーを返します。詳細は、[付録 B](#) を参照してください。

また、「クラスタの検証」を行うと、Oracle リソース DLL が Microsoft Cluster Server (MSCS) に登録されます。

[図 6-2](#) に、標準的な「クラスタの検証」操作の出力を示します。

図 6-2 「クラスタの検証」のクラスタワイド操作ウィンドウ



「クラスタの検証」操作を実行しても正常に終了しない場合、次に示すいずれか 1 つまたは複数の問題があると考えられます。

- ハードウェア、ネットワークまたは MSCS ソフトウェアの構成の問題
- Oracle ホームおよびバージョンの対称性の問題
- Oracle Fail Safe のインストール（たとえば、リソース・プロバイダの対称性）の問題

操作が正常に終了していながら Oracle Fail Safe で問題が発生している場合、Oracle Fail Safe の構成に問題があります。

6.1.2 グループの検証

「グループの検証」操作によって、グループが正常に動作することを確認するために次のことが行われます。

- グループ内のすべてのリソースをチェックし、グループの可能所有者であるすべてのノードで正しく構成されていることを確認します。
- グループ内のリソース間の依存性を更新します。
- プロンプトを表示した後で、構成の誤りのあるグループを修正します。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタにノードを追加する場合

グループを検証するには、Oracle Fail Safe Manager のツリー・ビューからグループを選び、次のように選択します。

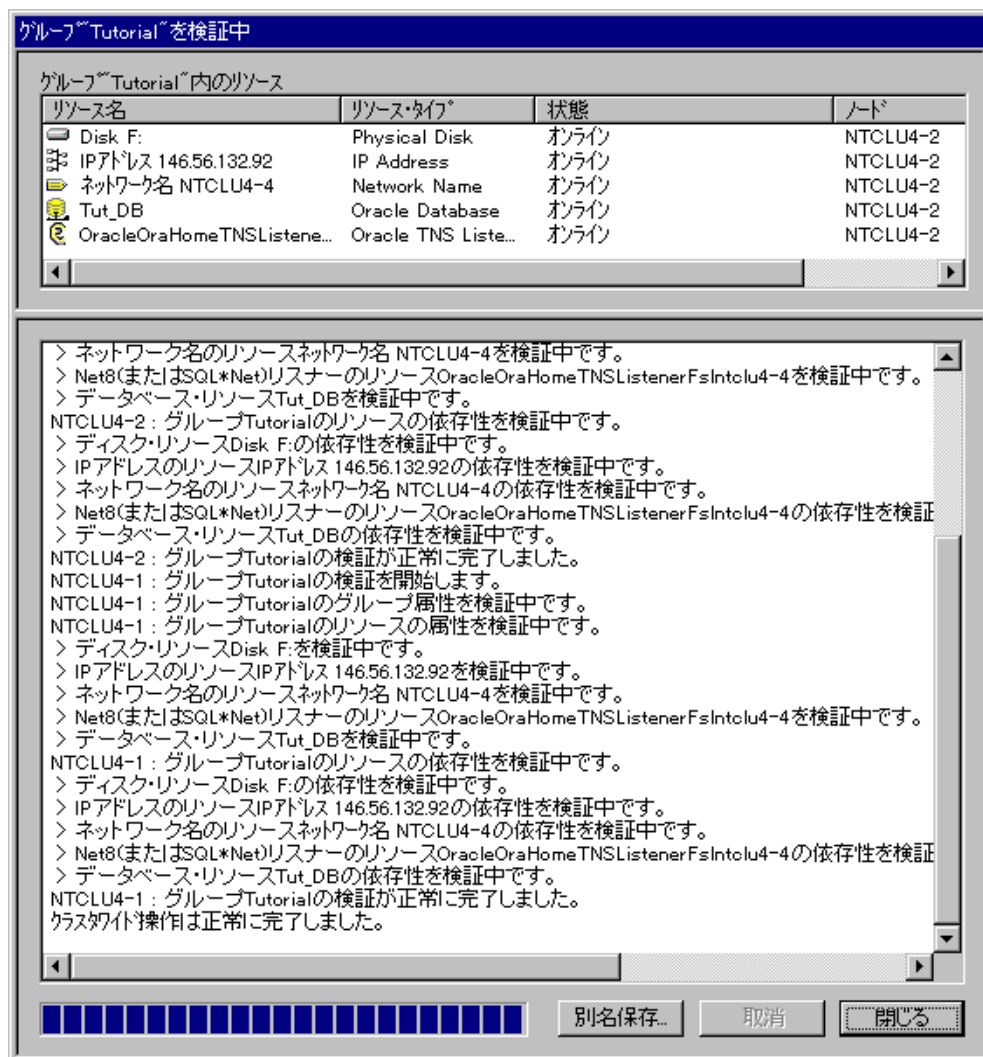
「トラブルシューティング」→「グループの検証」

または、FSCMD コマンドの VERIFYGROUP を使用して、「グループの検証」操作を実行することも可能です（第 5 章を参照）。FSCMD コマンドには、特定のクラスタ上にあり、Oracle Fail Safe によって構成されているグループをすべて検証する VERIFYALLGROUPS コマンドもあります。VERIFYGROUP コマンドと VERIFYALLGROUPS コマンドは、スクリプト内でバッチ・ジョブとして実行できます。

Oracle Fail Safe のグループ検証の際、「グループの検証」操作の進捗状況、グループ内の各リソースの状態を表示できます。

図 6-3 に、「グループの検証」操作の出力を示します。

図 6-3 「グループの検証」のクラスタワイド操作ウィンドウ



6.1.3 スタンドアロン・データベースの検証

スタンドアロン・データベースは、「スタンドアロン・データベースの検証」操作を使用していつでも検証できます。「スタンドアロン・データベースの検証」コマンドを発行するには、Oracle Fail Safe Manager のツリー・ビューからデータベースを選び、次のように選択します。

「トラブルシューティング」→「スタンドアロン・データベースの検証」

「スタンドアロン・データベースの検証」操作では、有効性検査を実行し、スタンドアロン・データベースが常駐するノード上に正しく構成されているかどうかを確認して、もう一方のクラスタ・ノードに存在するデータベースへの参照を削除します。（データベースが、一度グループに追加された後で削除されたものである場合、別のクラスタ・ノードにデータベースへの参照が残っている可能性があります。）これによって、Oracle Fail Safe を使用した可用性の高いデータベースが実現されます。

グループにスタンドアロン・データベースを追加する前に、そのデータベースに対して「スタンドアロン・データベースの検証」コマンドを実行してください。また、スタンドアロン・データベースへのアクセスに問題が生じた場合に、いつでもコマンドを使用できます。ただし、検証操作中、データベースは Oracle Fail Safe により停止および再起動されます。

たとえば、次のような場合に検証を実行します。

- グループにデータベースを追加しようとしたときに障害が発生し、その応答をする場合。
- Oracle Fail Safe Manager 以外の管理者ツールを使用してデータベースを処理し、そのデータベースにアクセスできなくなってしまった場合。
- Oracle Fail Safe Server ソフトウェアを削除しないまま、（たとえば、ソフトウェアのアップグレード中に）クラスタ・ノードから MSCS ソフトウェアを削除した場合。詳細は、『Oracle Fail Safe インストレーション・ガイド』で説明されています。

図 6-4 に、「スタンドアロン・データベースの検証」ダイアログ・ボックスを示します。このダイアログ・ボックスで、ユーザーはスタンドアロン・データベースの有効なデータベース情報およびアカウント情報を入力します。

図 6-4 「スタンドアロン・データベースの検証」ダイアログ・ボックス

「スタンドアロン・データベースの検証」ダイアログ・ボックスを使用する際、次の項目を指定します。

- 「サービス名」フィールドに、スタンドアロン・データベースのサービス名
- 「インスタンス名」フィールドに、スタンドアロン・データベースのインスタンス名
- 「データベース名」フィールドに、スタンドアロン・データベースのデータベース名
- 「パラメータ・ファイル」フィールドに、スタンドアロン・データベースの Internal アカウントのパラメータ・ファイル・ディスクとパス名
- 「Internal アカウント」領域に、スタンドアロン・データベースの Internal アカウントのセキュリティ情報

この情報を Oracle Fail Safe では次の場合に使用します。

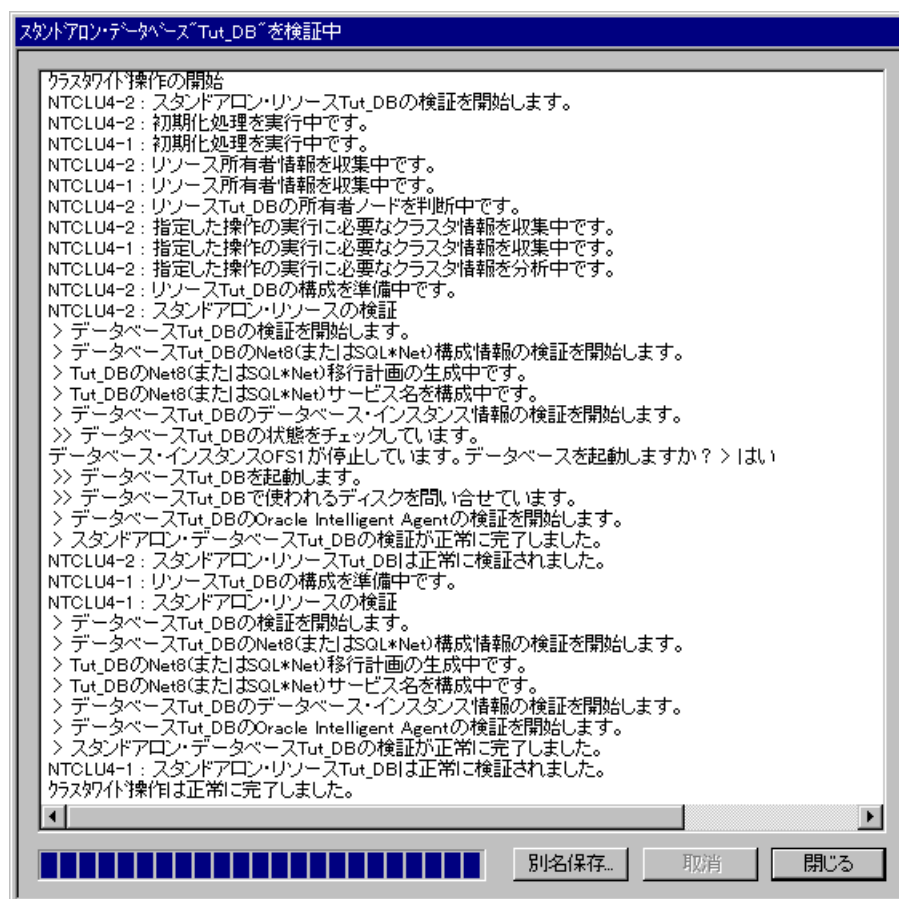
- Net8（または SQL*Net）によるクラスタワイドの問題の修正
- Oracle Fail Safe でデータベースにアクセス可能かどうかの確認
- スタンドアロン・データベースがクラスタ・ディスク上に存在するかどうかの確認

スタンドアロン・データベースがオープンされていると、「スタンドアロン・データベースの検証」操作を実行してもデータベースは再起動されません。

スタンドアロン・データベースがオープンされていないか、データベースが停止されている場合、Oracle Fail Safe では、データベース・インスタンスの停止および再起動を行ってよいかどうかユーザーに応答が求められます。次に、アクセスするためにデータベースがオープンされます。

図 6-5 に、クラスタワイド操作ウィンドウに表示される一般的な「スタンドアロン・データベースの検証」操作の出力を示します。

図 6-5 「スタンドアロン・データベースの検証」のクラスタワイド操作ウィンドウ



検証で問題が検出された場合、「スタンドアロン・データベースの検証」操作では問題を修正する前に、ユーザーの応答を求めます。たとえば、Net8（または SQL*Net）の問題が原因で、グループへのデータベースの追加に失敗した場合を考えてみましょう。「スタンドアロン・データベースの検証」コマンドを実行してネットワーク問題を修正した後、データベースをグループに追加できます。

6.2 クラスタのダンプ

Oracle Fail Safe には、Oracle Fail Safe Manager クラスタ・データをウィンドウ内に表示する「クラスタのダンプ」コマンドがあります。このコマンドは、一定期間内にクラスタに施された変更の記録を保持するために定期的に行（同時に出力を保存）したり、オラクル社カスタマ・サポート・センターの要請に従って、クラスタ環境のスナップショットを得るために実行する場合があります。

「クラスタのダンプ」コマンドを実行した際に提示されるデータには、次のものがあります。

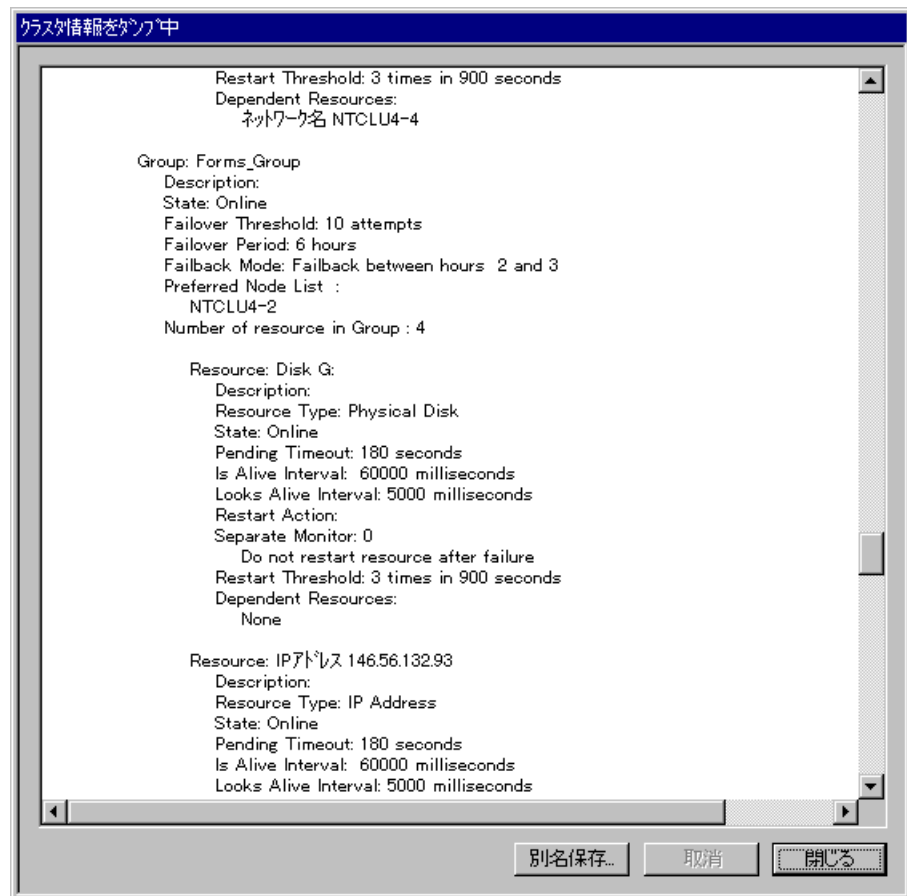
- オペレーティング・システムに関連する情報（クオラム・ディスクの場所など）
- パブリック・ネットワーク情報およびプライベート・ネットワーク情報
- クラスタとともに登録されたリソース
- グループのフェイルオーバー・ポリシーおよびフェイルバック・ポリシー

オプションで、「クラスタのダンプ」データをファイルに保存できます。「**別名保存**」をクリックしてください。

「クラスタのダンプ」コマンドを発行するには、Oracle Fail Safe Manager で「**トラブルシューティング**」→「**クラスタのダンプ**」を選択します。

図 6-6 に、「クラスタのダンプ」出力のうち、Forms_Group の情報とそのリソース情報の一部を示します。

図 6-6 グループを検証するクラスタワイド操作ウィンドウ



6.3 その他のトラブルシューティング情報の検索

この章では、Oracle Fail Safe Manager ファミリのトラブルシューティング・ツールについて説明しました。その他の情報は、次の方法で入手できます。

- 特定のコンポーネントのトラブルシューティングに関連する情報は、可用性の高いコンポーネントの構成について説明する、第 III 部「Oracle Fail Safe ソリューションの展開」の各章に記載してあります。
- ネットワーク構成問題に関連するトラブルシューティングについては、[付録 B](#) で説明してあります。
- Oracle Fail Safe の階層は Microsoft クラスタ・サーバ上に形成されているため、クラスタ・サービス、インターコネクトおよびハードウェア構成に関する問題を解決するには、MSCS のマニュアルを参照する必要がある場合があります。
- Oracle Fail Safe を起動できないときは、Microsoft Windows イベント ビューアを起動してアプリケーション・ログを表示します。通常、Oracle Fail Safe Server により、問題を識別するイベントが記録されます。

第III部

Oracle Fail Safe ソリューションの展開

第III部では、Oracle Fail Safe ソリューションの多様な展開方法について具体的に説明します。

第III部は、次の章から構成されています。

- 第7章「可用性を高めるためのデータベースの構成」
- 第8章「可用性を高めるための Oracle WebDB リスナーの構成」
- 第9章「可用性を高めるための Oracle Forms Load Balancer Server の構成」
- 第10章「可用性を高めるための Oracle Forms Server の構成」
- 第11章「可用性を高めるための Oracle Reports Server の構成」
- 第12章「可用性を高めるための Web サーバーの構成」
- 第13章「可用性を高めるための Oracle Applications Concurrent Manager の構成」
- 第14章「可用性を高めるための Oracle MTS Service の構成」
- 第15章「可用性を高めるための汎用サービスの構成」

可用性を高めるためのデータベースの構成

Oracle Fail Safe により、Microsoft クラスタで実行されている単一インスタンスの Oracle データベース（Workgroup Server および Enterprise Edition の両方）の可用性が向上します。

Oracle データベースの可用性を高めると、1つのクラスタ・ノードが停止または故障した場合でも、別のクラスタ・ノードでデータベースが再起動され、そのデータベースにアクセスするアプリケーションとデータベースとの接続が失われるのはほんの一瞬にすぎません。クラスタ対応のアプリケーションでは、このようなフェイルオーバー・イベントの発生後、自動的にデータベースに再接続できるため、結果的にフェイルオーバーがユーザーに認識されることはありません。

この章では、次の項目について説明します。

項目	参照
スタンドアロン・データベースの検出	7.1 項
Oracle データベースのグループへの追加	7.2 項
Oracle Net8 リスナー・リソースの作成および構成	7.3 項
可用性の高いデータベースへのクライアント接続	7.4 項
データベースのセキュリティ要件	7.5 項
可用性を高めるための ctxsrv サーバーの構成	7.6 項
Oracle Enterprise Manager との統合	7.7 項
データベース・リカバリの最適化	7.8 項
フェイルセーフ・データベースに対する管理作業の実行	7.9 項
クライアントおよびクラスタ対応のアプリケーション	7.10 項

項目	参照
可用性の高い Oracle データベースに関連するエラー処理と問題のトラブルシューティング	7.11 項

7.1 スタンドアロン・データベースの検出

Oracle Fail Safe Server では、スタンドアロン・データベース（グループの一部ではないデータベース）が検出されて Oracle Fail Safe Manager のツリー・ビュー内に表示されます。「サンプル・データベースの作成」コマンドを使用して作成したスタンドアロン・データベース（サンプル・データベース）と、他の方法で作成したスタンドアロン・データベースでは、次のように検出の方法が異なります。

- サンプルのスタンドアロン・データベース
サンプルのスタンドアロン・データベースは、Windows レジストリ内の次のキーを検索することにより検出されます。

HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥FailSafe¥SampleDB
- その他のスタンドアロン・データベース
その他のスタンドアロン・データベース（「サンプル・データベースの作成」コマンドで作成されていないスタンドアロン・データベース）は、各クラスタ・ノードで TNSNAMES.ORA ファイルを解析し、有効なネット・サービス名エントリを検索することにより検出されます。スタンドアロン・データベースは、次の条件が満たされると検出されます。
 - クラスタ・ノード上にスタンドアロン・データベースのインスタンスが存在
 - TNSNAMES.ORA ファイル内にスタンドアロン・データベースの次のような有効エントリが存在
 - * ノード上のデータベース・インスタンスに一致する SID、またはデータベース・パラメータ・ファイル中のサービス名に一致する SERVICE_NAME
 - * ホスト名または IP アドレス（Oracle Fail Safe では、ネットワーク別名の使用をサポートしていません。）

Oracle Fail Safe Manager のツリー・ビューにスタンドアロン・データベース（およびその他のスタンドアロン・リソース）を表示するには、各ノード・フォルダを拡張してから「スタンドアロン・リソース」フォルダを拡張します。

Oracle リリース 8.0.6 データベースを Oracle Database Configuration Assistant で作成する場合は、データベースが置かれるノードの tnsnames.ora ファイルに Net8 サービス名エントリを追加する必要があります。エントリを追加するまで、Oracle Fail Safe はデータベースを検出できません。Oracle Net8 Assistant を使用してエントリを追加できます。

7.2 Oracle データベースのグループへの追加

可用性が高まるように Oracle データベースを構成するには、データベースを少なくとも 1 つの仮想アドレスが含まれているグループに追加します。Oracle Fail Safe により、Oracle データベースに必要なすべてのリソースが追加されます。通常、このグループには次のリソースが含まれます。

- それぞれが 1 つの IP アドレスおよびネットワーク名から構成される 1 つ以上の仮想アドレス
- Oracle データベース・サーバー
- Oracle データベースで使用する全ディスク
- グループの仮想アドレス（複数も可）上でグループ内のデータベースへの接続要求をリスニングする Net8（または SQL*Net）ネットワーク・リスナー
- グループの仮想アドレスの 1 つを使用するように構成されている Oracle Intelligent Agent（データベースの管理に Oracle Enterprise Manager が使用される場合）

7.2.1 構成前に

データベースをグループに追加する前に、次のことに注意してください。

- データベース初期化パラメータ・ファイルを除き、データベースにより使用されるファイルはすべて共有クラスタ・ディスク上に置く必要があります。データベース初期化パラメータ・ファイルは、プライベート・ディスクにも共有クラスタ・ディスクにも配置できます。初期化パラメータ・ファイルの配置場所の詳細は、[7.2.3.3 項](#)を参照してください。
 - リソースが属するグループは 1 つのみです。このため、2 つのデータベースが同じディスク・ドライブを共有する場合は、この 2 つのデータベースを同一グループ内に指定する必要があります。
 - フェイルオーバーの際、一時表内のデータはフェイルオーバーされません。（ソートやハッシュ結合など）一時表や一時表領域を使用する操作では、フェイルオーバー・ノードで再起動されるときに必要な一時オブジェクトが再作成されます。ただし、一時表内の特定のデータに依存するアプリケーションが正常に機能することを確認する必要があります。
- 一時表の詳細は、『Oracle8i 概要』マニュアルの一時表に関する説明を参照してください。
- グループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

7.2.2 構成手順

表 7-1 に、可用性が高まるように Oracle データベースを構成する際に必要な作業の一覧を示します。各作業の詳細は、オンライン・ヘルプとチュートリアルを参照してください。手順ごとの指示を表示するには、Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」または「ヘルプ」→「チュートリアル」を選択します。

表 7-1 データベースを構成する手順

手順	処置	Oracle Fail Safe Manager での手順
1	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して「グループの作成」ウィザードを開きます。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。(仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。)
2	サンプル・データベースの作成 (必要な場合)	「リソース」→「サンプルの作成」→「データベース」を選択してサンプル・スタンダアロン・データベースを作成し、Oracle Fail Safe の機能を本番データベースで使用する前にテストできるようにします。サンプルは、本番作業用には使用しないでください。
3	スタンダアロン・データベースの検証	「トラブルシューティング」→「スタンダアロン・データベースの検証」を選択して、データベースとそのデータベース用の Net8 (または SQL*Net) の構成を検証します。このコマンドにより、Oracle Fail Safe Server がデータベースに接続でき、スタンダアロン・データベースがクラスタ・ディスク上に配置されていることが確認されます。
4	Oracle データベースのグループへの追加	「リソース」→「グループに追加」を選択してから「Oracle Database」を選択して「リソースをグループに追加」ウィザードを開きます。このウィザードは、可用性の高い Oracle データベース・サーバーを構成する際に役立ちます。
5	各クライアント・システム上の TNSNAMES.ORA ファイルの変更	仮想サーバーを認識するようにクライアントを構成します (Net8 Easy Config などのネットワーク構成ツールを使用して各クライアント・システム上の TNSNAMES.ORA ファイルを変更します)。詳細は、7.4 項を参照してください。

7.2.3 Oracle データベースの構成データ

Oracle Fail Safe Manager には、可用性が向上するように Oracle データベースを構成する際に役立つ「リソースをグループに追加」ウィザードがあります。ウィザードで表示されるページは、現在グループに含まれている仮想アドレスの数と、クラスタ内のノード数によって異なります。

1 つのグループに 1 つの仮想アドレスが含まれる構成が典型的ですが、複雑な構成では複数の仮想アドレスが含まれる場合もあります。「リソースをグループに追加」ウィザードを使用して典型的な構成を実行するには、次のデータが必要です。

- Oracle データベースの識別情報
- データベースの INTERNAL アカウントとパスワード

複数の仮想アドレスが含まれているグループにデータベースを追加する場合は、クライアントがデータベースへのアクセスに使用する仮想アドレス（複数も可）を指定するように要求されます。

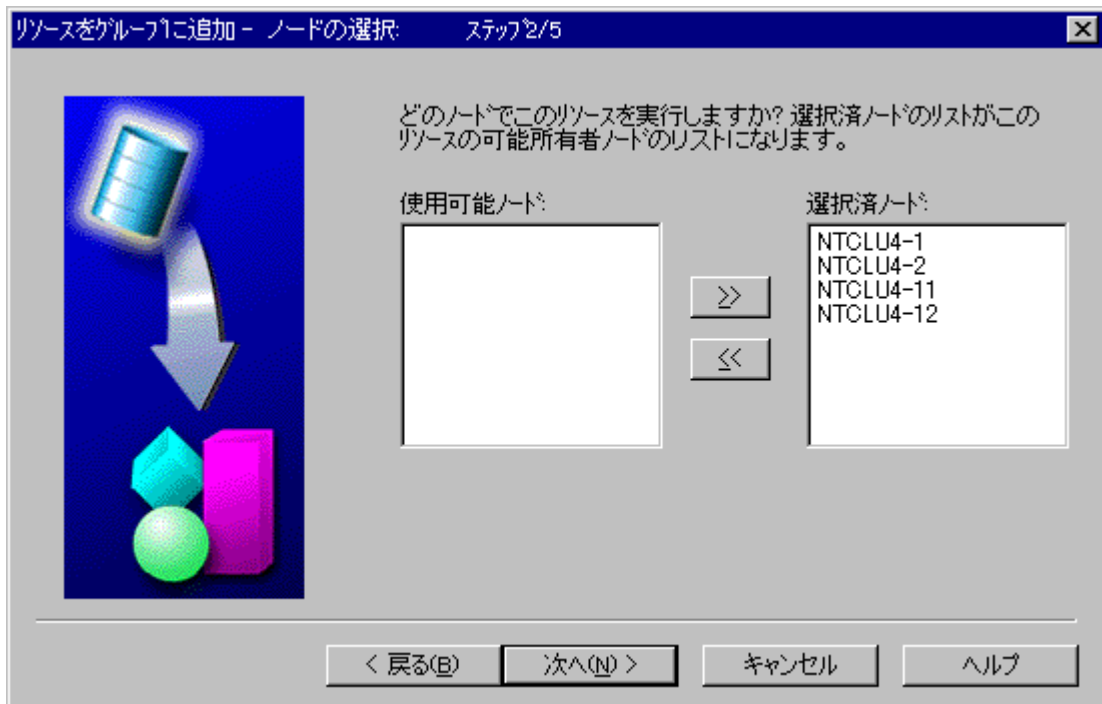
次の項では、構成の要件を詳細に説明します。

7.2.3.1 ノードの選択

データベースをグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、図 7-1 に示すように、選択済ノードのリストを指定して、データベースの可能所有者となるノードを指定するように求められます。特定のノードをデータベースの可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

2.6.6 項では、「可能所有者ノード」リストの概念について詳細に説明します。

図 7-1 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



データベースをグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードをデータベースの可能所有者とするのかを指定するように求められます。このような場合、図 7-2 に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

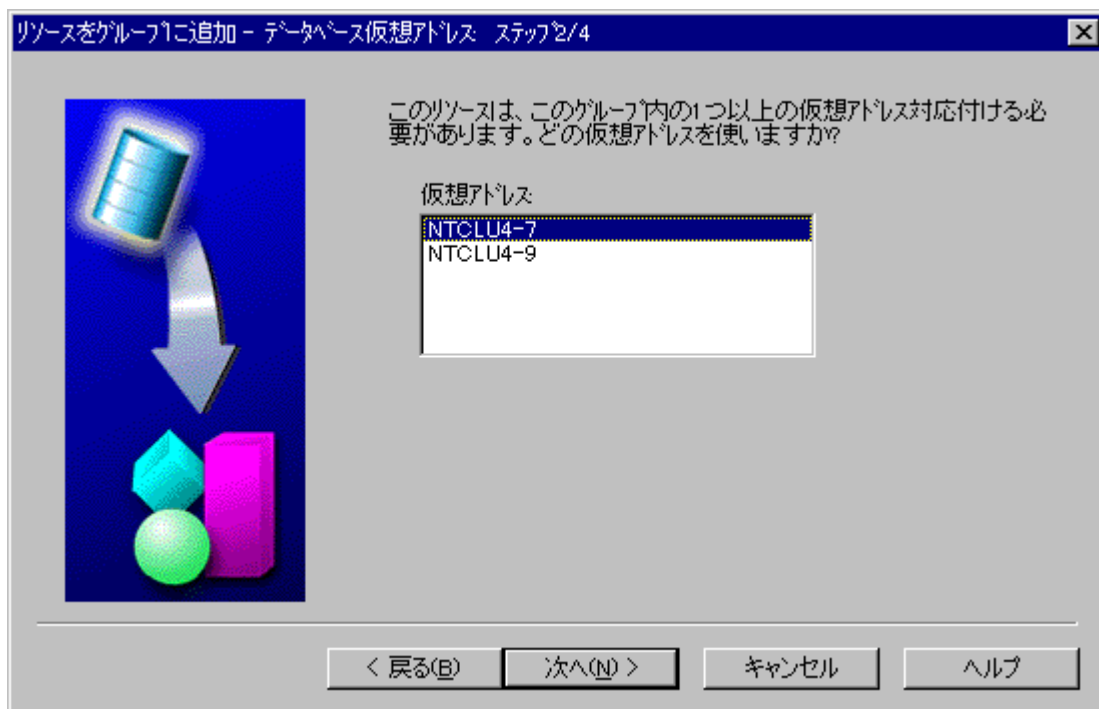
図 7-2 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



7.2.3.2 仮想アドレス

図 7-3 に示すように、データベースを追加しようとするグループに複数の仮想アドレスが含まれている場合、クライアントがグループ内のデータベースにアクセスするときに、グループ内のどの仮想アドレスを使用するかを尋ねる「リソースをグループに追加」ウィザード・ページが表示されます。仮想アドレスが1つしか含まれていないデータベースを追加する場合には、このページは表示されません。

図 7-3 「データベース仮想アドレス」ウィザード・ページ



Oracle Fail Safe では、1つのグループに複数の仮想アドレスがサポートされます。グループ内のすべてのデータベースは同じ仮想アドレスを使用する必要があり、グループにデータベースを追加する前に仮想アドレスがグループに追加される必要があります。グループの作成順序は、次のとおりです。

1. グループの作成
2. グループに1つ以上の仮想アドレスを追加
3. グループに1つ以上のデータベースを追加

たとえば、グループに2つの仮想アドレスを持つデータベースが含まれ、2番目のデータベースをそのグループに追加する場合、2番目のデータベースは、グループに構成されていた最初のデータベースと同じ仮想アドレスを使用する必要があります。Oracle Fail Safe Manager は、グループに追加したすべてのデータベースで同じ仮想アドレスが使用されていることを確認します。

複数の仮想アドレスを持つグループ内にリソースを構成する方法の詳細は、[4.7 項](#)を参照してください。

7.2.3.3 データベース識別情報

図 7-4 に示すように、「リソースをグループに追加」ウィザードでは、可用性が高まるように構成されるデータベースを一意に識別する、データベース識別情報を入力する必要があります。Oracle Fail Safe では、（たとえば、TNSNAMES.ORA を更新する場合）このデータを使用してデータベースがクラスタ内に構成されます。また、ユーザーが入力したデータが MSCS に渡されます。データは MSCS に登録されて、データベースがオンラインまたはオフラインになるとき、あるいは Is Alive ポーリングが実行されるときに使用されます。Oracle Fail Safe では、次の情報が要求されます。

- サービス名

これは**ネット・サービス名**です。（Oracle8i より前には、**サービス名**と呼ばれていました。）この名前は、Oracle Fail Safe Manager のツリー・ビューと MSCS のツリー・ビューに表示されます。クライアント・アプリケーションでは、この名前を接続要求に指定します。

Net8 サービス名にドメイン名を指定しない場合、7.4.1.1 項で説明するように Oracle Fail Safe によってドメイン名が選択され、ネット・サービス名に追加されます。

- インスタンス名

これはデータベース・インスタンスの名前であり、SID とも呼ばれます。

- データベース名

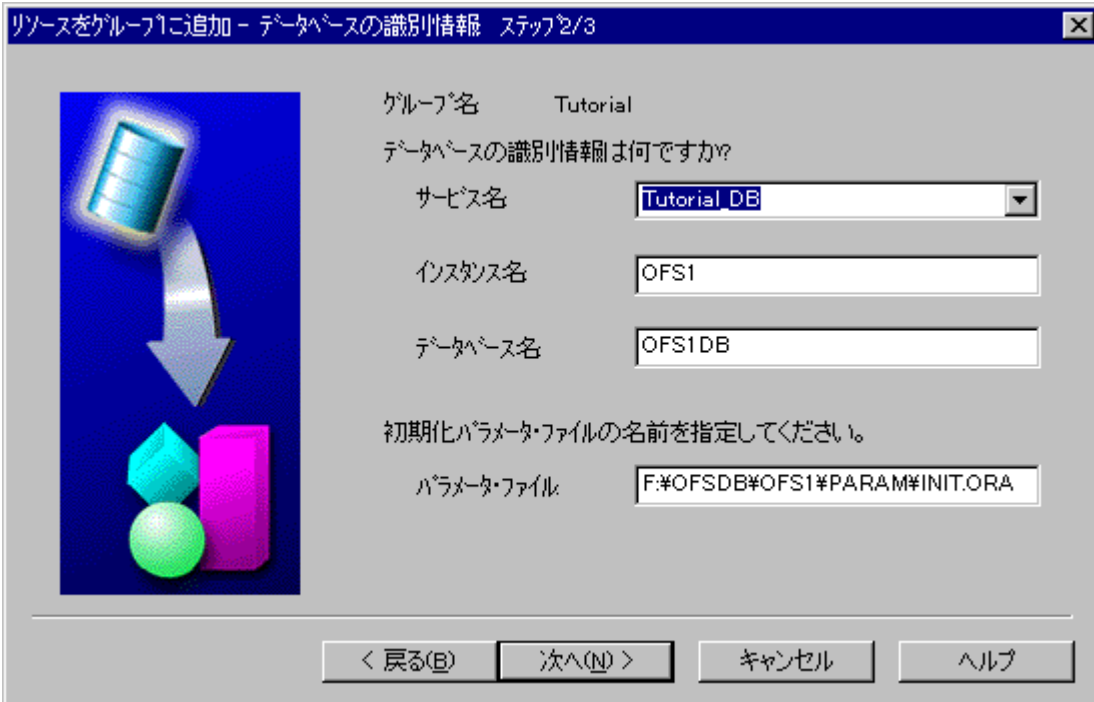
これは、初期化パラメータ・ファイル内でデータベースの識別に使用される DB_NAME パラメータです。データベース名は、データベースの作成時に（たとえば、CREATE DATABASE SQL 文内で）使用された名前です。

- 初期化パラメータ・ファイルの名前と場所

Oracle Server の起動時には、初期化パラメータ・ファイルを使用してデータベース名、メモリーの割当量、制御ファイル名、各種の制限事項、およびその他のシステム・パラメータが指定されます。

通常は、どのクラスタ・ノードがデータベースのホストであるかにかかわらずパラメータ・ファイルにアクセスできるように、パラメータ・ファイルはクラスタ・ディスク上に配置します。ただし、データベースが稼働するように構成されたすべてのクラスタ・ノードの同じ場所に初期化パラメータ・ファイルを置くことが確実な場合は、各ノードのプライベート・ディスクにファイルのコピーを配置することも可能です。また、ホストになっているノードに応じてデータベースに異なるパラメータを設定するために、各ノードのプライベート・ディスクにパラメータ・ファイルを配置することもできます。ノード間でメモリー量または処理能力に差がある場合は、このような方法が有効です。

図 7-4 「データベースの識別情報」ウィザード・ページ



リソースをグループに追加 - データベースの識別情報 ステップ2/3

グループ名 Tutorial

データベースの識別情報は何ですか?

サービス名 Tutorial_DB

インスタンス名 OFS1

データベース名 OFS1DB

初期化パラメータファイルの名前を指定してください。

パラメータファイル F:\OFSDB\OFS1\PARAM\INIT.ORA

< 戻る(B) 次へ(N) > キャンセル ヘルプ

7.2.3.4 データベース認証

図 7-5 に示すように、「リソースをグループに追加」ウィザードでは、起動、停止および Is Alive ポーリングの実行時に使用できるアカウントのユーザー名とパスワードを入力するように要求されます。Oracle Fail Safe が実行されている Windows オペレーティング・システム (OS) ・アカウントを使用することも、データベース・ユーザー・アカウントを指定することもできます。Windows アカウントを使用するには、「OS 認証を使用」を選択します。アカウントを指定するには、「このアカウントを使用」を選択します。

データベース・ユーザー・アカウントを指定した場合、オラクル社では INTERNAL アカウントの使用をお勧めしますが、DBA 権限を持つ任意のアカウントを使用できます。

OS 認証を使用する場合、Oracle データベースの起動や停止は、管理者権限を持ったアカウントのみが実行できます。データベース・ユーザー・アカウントを使用し、可用性が高まるようにデータベースを構成した後でこのアカウントのパスワードを変更する場合は、Oracle Fail Safe Manager を使用してパスワードを更新する必要があるので注意してください。可用性を高めるためのデータベース構成後にこのアカウントのパスワードを更新する方法の詳細は、7.5.2 項を参照してください。

可用性が高まるようにデータベースが構成された後、認証のタイプを切り替えるには、グループからリソースを削除し、ウィザードを再実行して目的の認証タイプを指定する必要があります。

図 7-5 「データベースの認証」ウィザード・ページ

リソースをグループに追加 - データベースの認証 ステップ 3/3

Oracle Fail Safe Serverは構成情報のためにデータベースにアクセスする必要があります。INTERNALアカウントのパスワードは何ですか?

☐ OS認証を使用

☒ このアカウントを使用

INTERNAL

パスワード: *****

パスワードの確認: *****

< 戻る(B) 完了(F) キャンセル ヘルプ

7.3 Oracle Net8 リスナー・リソースの作成および構成

データベースをグループに追加すると、Oracle Fail Safe では、Net8 リスナー・リソースおよびデータベース・リソースがグループに作成および構成されます。リリース 3.1 より前の Oracle Fail Safe では、データベース・リソースがリスナー・リソースに依存していました。(過負荷などによって) リスナーで障害が発生すると、クラスタによってデータベース・リソースがオフライン化されました。しかし、データベース・リソースはオンライン状態のリスナーを必要としません。リスナーが必要なのは、クライアントがデータベースに接続するためです。

そのためリリース 3.1 以降の Oracle Fail Safe では、データベース・リソースとリスナー・リソースの間に依存性を作成しません。リスナーに依存しているデータベースが検出された場合、Oracle Fail Safe によってその依存性は削除されます。データベースがオンラインになる

と、Oracle Fail Safe は最初に、リスナー・リソースがオンラインであることを確認します。データベース・リソースの Is Alive ポーリング中に、リスナーを介してデータベースに接続できない場合には、**Bequeath プロトコル**・アダプタを使用してデータベースに接続します。この障害をユーザーに通知するログ・イベントも記録されます。このログ・イベントが表示される場合、リスナー・リソースをチェックしてください。リスナー・リソースの障害が、データベースへの既存の接続に影響することはありません。ただし、新規のユーザーがデータベースに接続できない原因にはなります。リスナー・リソースは独自の再起動ポリシーを持つため、障害発生時にはクラスタによって自動的に再起動されます。

7.3.1 Net8 リスナーの定義

Oracle8i リリース 8.1.5 から、Oracle Net8 リスナーの動作が変更になっています。Oracle8i リリース 8.1.5（以降）のリスナーの定義でマシンのホスト名が使用されている場合、このリスナーはホスト名に関連付けられている IP アドレスのみでなく、ノード上のすべての IP アドレスでリスニングを行います。これは、Oracle Fail Safe とともに構成され仮想アドレスでリスニングするよう定義されたリスナーとの競合の原因となります。

この変更を回避するために、リスナーはホスト・エントリにホスト名ではなくノード IP アドレスを使用する必要があります。データベースをグループに追加する際、Oracle8i リリース 8.1.5（以降）のリスナーでホスト・エントリを使用していることが検出された場合、このエントリを修正して IP アドレスを使用するかどうかを確認されます。修正しないと、「リソースをグループに追加」操作は続行できません。

次に示すのは、Oracle Fail Safe 環境で無効なエントリの例です。

```
LISTENER =
....
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=NTCLU-52)
    (PORT=1521)
  )
```

次に示すのは、Oracle Fail Safe 環境で有効なエントリの例です。

```
LISTENER =
....
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=138.2.26.52)
    (PORT=1521)
  )
```

7.4 可用性の高いデータベースへのクライアント接続

(データベースを含む) ネットワーク・オブジェクトは、ネットワーク・アドレスによって識別されます。クライアントとデータベース間を接続するには、クライアントの TNSNAMES.ORA ファイル内のネットワーク・アドレスと、サーバーの LISTENER.ORA ファイル内のネットワーク・アドレスが一致している必要があります。つまり、クライアントがネットワーク・アドレスを使用してネットワーク・オブジェクトの特定の場所に接続要求を送ると、受け側は、このアドレス上で要求をリスニングし、クライアント情報と一致するアドレス情報に基づいて接続を認可します。

データベースをグループに追加すると、Oracle Fail Safe では、データベースが常駐する Oracle ホーム内にグループのリスナーを作成します。仮想アドレス情報を構成する際、Oracle Fail Safe ではデータベースの可能所有者であるクラスタ上、および Oracle Fail Safe Manager を実行しているクライアント・システム上のすべての Oracle ホーム内の TNSNAMES.ORA ファイルが更新されます。これによって、次に示す要因とは無関係に、Oracle Fail Safe で常に最新の構成でデータベースにアクセスできるようになります。

- Oracle Fail Safe をインストールした Oracle ホーム
- Oracle Fail Safe Server によりロードされる Oracle データベース DLL ファイルが置かれている Oracle ホーム
- 現在の Windows PATH 変数の設定

次の項では、LISTENER.ORA ファイルにエントリを作成し、TNSNAMES.ORA ファイルを更新して、スタンドアロン・データベースが稼働しているノードのホスト名を、データベースが追加されたグループの仮想サーバー・アドレスに置き換える方法を説明します。これにより、どのクラスタ・ノードがデータベースのホストであるかに関係なく、クライアントは確実にデータベースに接続できます。

7.4.1 データベースをグループに追加すると更新される Net8 構成

次の項で説明するように、データベースをグループに追加すると、Oracle Fail Safe により TNSNAMES.ORA ファイルおよび LISTENER.ORA ファイル内のデータベースの Net8 構成が変更されます。

7.4.1.1 Oracle Fail Safe により行われる TNSNAMES.ORA ファイルの更新内容

データベースをグループに追加すると、そのデータベースの TNSNAMES.ORA ファイル内のネット・サービス名は、グループの仮想アドレスを使用するように更新されます。ノード上に Oracle ホームが複数ある場合は、すべての TNSNAMES.ORA ファイルが更新されます。さらに、次のことが行われます。

- サンプル・データベースを作成、またはグループにデータベースを追加する際に、Net8 サービス名にドメイン名を指定しない場合、次のように Oracle Fail Safe によってドメイン名が選択され、ネット・サービス名に追加されます。

- Oracle Fail Safe が、ノード上で最高のデータベース・バージョンの Oracle ホームでデフォルトのドメイン名を検索します。これが見つかった場合、デフォルトのドメイン名がネット・サービス名に追加されます。たとえば、Oracle 8.1.6 がノード上で最高のデータベース・バージョンであると仮定します。Net8 ネット・サービス名に「MyDB」と指定し、Oracle 8.1.6 ホームでのデフォルトのドメイン名が「us.oracle.com」であれば、ネット・サービス名は「MyDB.us.oracle.com」となります。
- ノード上の最高データベース・バージョンの Oracle ホームにデフォルトのドメイン名がない場合、ネット・サービス名に「ofsdomain」が追加されます。たとえば、「MyDB」と指定した場合、ネット・サービス名は「MyDB.ofsdomain」となります。
- 「リソースをグループに追加」ウィザードで入力したものと同一ネット・サービス名を Oracle Fail Safe で使用する場合は、CONNECT_DATA パラメータの値は変更されません。たとえば、ネット・サービス名が SERVICE_NAME パラメータで構成されている場合、CONNECT_DATA パラメータでは引き続き SERVICE_NAME パラメータを使用します。ただし、Oracle Fail Safe でデータベースに新しいネット・サービス名を作成する必要がある場合は、CONNECT_DATA パラメータは SID パラメータを使用するように変更されます。
- サーバー・ノードでは、内部クラスタ通信専用に構成された仮想ホスト名を使用するアドレスが、アドレス・リストの最上位に置かれます。そのため、サーバー・ノードで実行されるアプリケーション（データベース・リソース・モニターを含む）は、データベース・リソースへの接続にパブリック・インターコネクトを利用するアドレスを使用する前に、クラスタ・ノード間のプライベート・ネットワーク・インターコネクトを利用するアドレスの使用を試みます。

クライアント・ノードでは、内部クラスタ通信専用に構成された仮想ホスト名を使用するアドレスは、アドレス・リストに含まれません。

注意： Oracle Fail Safe により、（そのデータベースの可能所有者である）すべてのクラスタ・ノード上、および Oracle Fail Safe Manager を実行しているクライアント・ノード上で TNSNAMES.ORA ファイルが更新されます。（Oracle Fail Safe を実行していない）リモート・クライアントがクラスタ・ノード経由で Oracle データベースに対して作業を処理できるようにする必要があります場合は、TNSNAMES.ORA ファイルを編集し、仮想アドレス情報でホスト名を更新する必要があります。ネットワーク構成ツールを使用して各クライアントのローカル TNSNAMES.ORA ファイルを編集してください。

7.4.1.2 Oracle Fail Safe により行われる LISTENER.ORA ファイルの更新内容

データベースをグループに追加すると、Oracle Fail Safe では LISTENER.ORA ファイルを次のように変更します。

1. スタンドアロン・データベース・リスナーから SID_DESC パラメータを削除します。
2. データベースに関連付けられた仮想アドレスで動作するように構成された Oracle Fail Safe リスナーを新規作成します。
3. SID_DESC パラメータを新規の Oracle Fail Safe リスナーに追加します。
4. スタンドアロン・データベース・リスナーを停止後、再起動して変更内容を有効にします。
5. 新規の Oracle Fail Safe リスナーを起動します。

7.4.2 Database Configuration Assistant を使用して作成した Net8 構成の作成および更新

Database Configuration Assistant (DBCA) を使用してスタンドアロン・データベースを作成すると、新規データベースの情報が次のように Net8 構成に追加されます。

1. DBCA により、データベースの SID_DESC パラメータがデフォルトの LISTENER.ORA ファイル内のデフォルト・リスナーに追加されます。この SID_DESC パラメータにはデータベース SID 名が含まれます。
2. DBCA により、データベースのネット・サービス名エントリがデータベースの Oracle ホームのみの TNSNAMES.ORA ファイルに追加されます。このエントリには SERVICE_NAME パラメータが含まれます。

DBCA により Net8 情報が構成された後、Oracle Fail Safe では、Oracle Fail Safe Manager のツリー・ビュー（データベースが作成されたクラスタ・ノードの「スタンドアロン・リソース」の下）に新規データベースが表示されます。

データベースを作成したクラスタ・ノードに複数の Oracle ホームがある場合は、新規データベースに対して Oracle Fail Safe Manager の「スタンドアロン・データベースの検証」コマンドを実行してください。Oracle Fail Safe により、すべての Oracle ホーム内の TNSNAMES.ORA ファイルがチェックされます。TNSNAMES.ORA ファイルにデータベースのネット・サービス名エントリが含まれていないことが検出されると、そのファイルを更新してよいかどうかを尋ねられます。「はい」を選択すると、Oracle Fail Safe では新規データベースのネット・サービス名エントリを追加します。

デフォルトのドメイン名の値（たとえば、SQLNET.ORA ファイルの NAMES.DEFAULT_DOMAIN パラメータの値）が Oracle ホーム間で異なる場合は、ネット・サービス名エントリはいくつかの Oracle ホームからアクセスできなくなります。この問題を解決するには、各 Oracle ホーム内の TNSNAMES.ORA ファイルを編集し、それぞれの Oracle ホームのデフォルト・ドメイン名をネット・サービス名エントリに追加します。

7.4.3 Oracle8i Database サーバー・ソフトウェアへのアップグレードに伴って必要な SID リスト・エントリ (SID_DESC)

Oracle データベース・サーバー・ソフトウェアを、Oracle7 または Oracle8 から Oracle8i にアップグレードする場合、Oracle8i リスナー (7.3.1 項を参照) を使用する必要があり、次のように Oracle7 または Oracle8 リスナーの SID リストを Oracle8i リスナーに移す必要があります。

1. Oracle8i リリースと、Oracle7 または Oracle8 リリースの両方の LISTENER.ORA ファイルを検索します。たとえば、次のようになります。
 - Oracle リリース 7.3.4 および Oracle8i の LISTENER.ORA ファイルは
<Oracle_Home>%NETWORK%ADMIN ディレクトリにあります。
 - Oracle8 リリースの LISTENER.ORA ファイルは
<Oracle_Home>%NET80%ADMIN ディレクトリにあります。
2. Oracle7 または Oracle8 リスナーの SID リストを特定します。

たとえば Oracle8 リスナーの場合、
<Oracle_Home>%NET80%ADMIN%LISTENER.ORA に次のようなリストがあります。

```
SID_LIST_LISTENER =
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=ORCL)
    )
    (SID_DESC=
      (SID_NAME=OFS2)
    )
  )
```

3. 手順 2 の SID リストにある SID_DESC エントリをコピーし、Oracle8i リスナーの SID リストにそれを追加します。たとえば、
<Oracle_Home>%NETWORK%ADMIN%LISTENER.ORA での更新後の SID リストは次のようになります。

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME= c:\%oracle)
      (PROGRAM = extproc)
    )
    (SID_DESC=
      (SID_NAME=ORCL)
    )
    (SID_DESC=
```

```
(SID_NAME=OFS2)
)
)
```

4. Oracle8i リスナーで、ホスト名を IP アドレスに変更します。
5. Oracle7 または Oracle8 データベースのリスナーが起動している場合は、それを停止します。さらに、Oracle7 または Oracle8 リスナーの起動状態を手動に変更します。（これを変更するには、Windows の「コントロール パネル」から「サービス」ウィンドウを開いてください。）
6. Oracle8i リスナーを起動します。

Oracle8i リスナーを使用しない場合、Oracle Fail Safe で次の操作を実行するときに問題が発生します。

- グループの検証
- データベース・リソースのグループへの追加
- データベース・リソースのグループからの削除

7.4.4 複数のリスナーを伴うノードでの Net8（または SQL*Net）の構成

Oracle Fail Safe では、スタンドアロン・データベースのリスナーの検索時に、すべての Oracle ホームで定義されたリスナーが検索されます。複数の Oracle Server が（1 つ以上の Oracle ホームに）インストールされている場合は、複数のリスナーが存在する可能性があります。複数のリスナーがある場合でも、Oracle Fail Safe ではスタンドアロン・データベースのリスナーを検索することが可能です。

システムに追加の Oracle ホームをインストールした後、使用するリスナーの数およびそのシステムで使用するリスナーを決定する必要があります。その後で、次のことを行います。

1. 使用しないリスナーの定義を LISTENER.ORA ファイルから削除します。これにより、Oracle Fail Safe で不要なリスナーが検出されなくなります。
2. 同じアドレスまたは SID で複数のリスナーを実行していないことを確認します。（同じアドレスまたは SID で別のアクティブ・リスナーがすでに動作中の場合は、Oracle リスナーを起動できません。）
3. 使用するリスナーを起動します。Windows の「コントロール パネル」にある「サービス」の「スタートアップ」で、これらのリスナーが自動的に起動するように設定されていることを確認します。

スタンドアロン・データベースのリスナーを検索するとき、LISTENER.ORA ファイルに定義されたリスナーの状態によって、検索結果が左右されます。Oracle Fail Safe でリスナーを検索する順序は、次のとおりです。

1. 起動しているリスナー
2. 停止しているリスナー

3. Windows サービス定義のないリスナー

たとえば、データベースの2つのリスナーが、それぞれ別の Oracle ホームにあるとします。このとき、ホーム1のリスナーは停止しており、ホーム2のリスナーは起動しているとします。Oracle Fail Safe では、起動しているリスナーを最初に検索するため、システム上の2つのホームを検索するとホーム2のリスナーが検出されます。起動しているリスナーが見つかったら、検索を中止します。起動しているリスナーが見つからない場合は、停止しているリスナーを検索し、それでも見つからない場合は、次のリスナー・グループを検索します。

注意： Oracle Fail Safe のどの操作においても、実行前にスタンドアロン・データベースのリスナーが予定どおりの状態（停止または起動）であることを確認してください。

Oracle Fail Safe では、グループのデータベースが常駐するのと同じ Oracle ホームに、そのグループのリスナーが作成されます。

7.4.5 高い可用性を提供するように構成されたデータベースでの外部プロシージャの使用

Oracle Fail Safe では、グループの Net8 リスナー定義（LISTENER.ORA ファイル）内に外部プロシージャのアドレスが構成されます。最初のデータベースがグループに追加される時点で、データベースにサービスを提供する元の Net8 リスナーに外部プロシージャが構成されているかどうか判断されます。構成されている場合、グループの LISTENER.ORA ファイル内に、IPC リスナー・アドレスおよび SID 記述子（SID_DESC）の両方が作成されます。

Oracle Fail Safe では、グループ内の最初の仮想アドレスのネットワーク名に接頭辞 EXTPROC を連結して、IPC アドレスのキーを作成します。グループ内の最初の仮想アドレスは、アルファベット順の仮想アドレス・ネットワーク名を使用して決定されます。したがって、最初の仮想アドレスが ntclu45 の場合は、キー値は EXTPROCntclu45 で、IPC アドレス・エントリは次のようになります。

```
(ADDRESS=
  (PROTOCOL=IPC)
  (KEY=EXTPROCntclu45)
)
```

Oracle Fail Safe では、データベースの元の Net8 リスナーから SID 記述子情報をグループの Net8 リスナーにコピーします。たとえば、データベースの元の Net8 リスナーに次の SID 記述子がある場合、グループの Net8 リスナーの記述子もこれと同じになります。

```
(SID_DESC=
  (SID_NAME=extproc)
  (ORACLE_HOME = C:\Oracle\Ora8i)
  (PROGRAM=extproc)
)
```

PL/SQL または SQL のアプリケーションで外部プロシージャをコールする場合、アプリケーションでは TNSNAMES.ORA ファイル内に EXTPROC_CONNECTION_DATA ネット・サービス名エントリを指定します。このエントリには、リスナーが外部プロシージャのプロセスを起動するために必要な情報が含まれています。クラスタ内のすべての TNSNAMES.ORA ファイルで、EXTPROC_CONNECTION_DATA ネット・サービス名エントリに、グループ内の Net8 リスナーの IPC アドレスが追加されます。

7.4.6 マルチスレッド・サーバーを使用するデータベースのサポート

次の項では、マルチスレッド・サーバー構成を使用する Oracle8 および Oracle8i データベースを、Oracle Fail Safe でどのようにサポートするかを説明します。

7.4.6.1 Oracle8 データベース用のマルチスレッド・サーバー

Oracle Fail Safe では、マルチスレッド・サーバー構成を使用する Oracle8 データベースをサポートします。ただし、マルチスレッド・サーバー構成が定義されているデータベース初期化ファイルは自動的に更新されません。

スタンドアロン・データベース、または現在グループ内のリソースであるデータベースを構成して、マルチスレッド・サーバーを使用するようにできます。いずれの場合も、データベース初期化ファイルは次の手順で更新する必要があります。

1. マルチスレッド・サーバー構成を使用するデータベースを含むグループのリスナー・パラメータを、次のように特定します。

- a. データベースが含まれている Oracle ホーム内の Net8（または SQL*Net）構成ディレクトリで、LISTENER.ORA ファイルを検索します。

たとえば、データベースが Oracle ホーム（C:\ORANT805）にインストールされている場合、LISTENER.ORA ファイルは C:\ORANT805\NET805\ADMIN ディレクトリにあります。

- b. LISTENER.ORA ファイル内でデータベースの SID を検索し、TCP プロトコルを使用するグループの最初のリスナー・アドレスを見つけます。

たとえば、次の LISTENER.ORA ファイルでは、太字になっているテキストがグループの最初のリスナー・アドレスを示します。

```

LISTENER =                               (Entries for default Listener)
  (ADDRESS_LIST =
    .
    .
    .
Fslvirtualnode =                         (Entries for Fail Safe Listener)
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=IPC)
      (KEY=OFS1)
    )
  )

```



```

        (ADDRESS=
          (PROTOCOL=IPC)
          (KEY=805mts.world)
        )
        (ADDRESS=
          (PROTOCOL=TCP)
          (Host=virtualnode)
          (Port=1521)
        )
        (ADDRESS=
          (PROTOCOL=TCP)
          (Host=virtualnode)
          (Port=1526)
        )
      )
    )
  )
  SID_LIST_Fslvirtualnode =
    (SID_LIST=
      (SID_DESC=
        (SID_NAME=OFS1)
      )
    )
  )
)

```

- c. 行 (PROTOCOL=TCP) を含む最初のアドレスをファイル内で検索し、アドレス・パラメータを 1 行に書式化します。たとえば、次のようになります。

```
(ADDRESS=(PROTOCOL=TCP) (Host=virtualnode) (Port=1521))
```

2. データベース初期化ファイル（たとえば、INITOFS1.ORA）を更新して、グループのリスナー・パラメータを使用します。これには次の手順を実行します。

- a. データベース初期化ファイルを開きます。

データベースの初期化パラメータ・ファイル（たとえば、`H:\OFSDB\OFS1\PARAM\INITOFS1.ORA` など）は、共有インターコネクト上のディスクに存在する場合があるため注意してください。

クラスタの各ノードのプライベート・ディスク上にこのファイルのコピーがある場合は、すべてのコピーを更新する必要があります。

- b. データベース初期化パラメータ・ファイルで、次のパラメータを含む行を検索します。

```
mts_listener_address
```

- c. MTS_LISTENER_ADDRESS パラメータの値を、手順 1c で書式化したリスナー・アドレスに置き換えます。

たとえば、元の MTS_LISTENER_ADDRESS パラメータに次の値が含まれているとします。

```
mts_listener_address = "(ADDRESS=(PROTOCOL=TCP) (HOST=node1) (PORT=1521))"
```

この行を次のように置き換えます。

```
mts_listener_address =  
"(ADDRESS=(PROTOCOL=TCP) (HOST=virtualnode) (PORT=1521))"
```

- d. データベース初期化ファイルを保存します。
3. mts_service パラメータの値がデータベース SID であるかどうかをチェックします。

Database Configuration Assistant では、mts_service パラメータの値にデータベース名が使用される場合があります。その場合、値をデータベース SID に変更してください。

4. グループ内のリソースを停止してから再起動します。

変更を有効にするには、Oracle Fail Safe Manager または FSCMD コマンドを使用して、データベースを含むグループをオフラインにしてからオンラインに戻します。これにより、グループ内のすべてのリソースが一度停止されてから再起動されます。

7.4.6.2 Oracle8i データベース用のマルチスレッド・サーバー

Oracle8i および Oracle Fail Safe の両方が含まれた環境でマルチスレッド・サーバー構成を使用する場合、次の項で説明するように、Oracle8i データベースのデータベース・パラメータ・ファイルに変更が必要になることがあります。

7.4.6.2.1 スタンドアロンの Oracle8i データベース スタンドアロンの Oracle8i データベースのマルチスレッド・サーバー構成がデフォルト・リスナーに依存している場合、データベース・パラメータ・ファイル内にリスナー・パラメータが指定されていません。（デフォルト・リスナーとは、ノードのホスト名、デフォルト・ポート番号および TCP プロトコルで動作するリスナーです。）この場合、Oracle Fail Safe でホスト名ではなく IP アドレスを使用するようにデフォルトのリスナーが変更されると、その構成が機能しなくなります。

この問題を解決するには、次の手順を実行し、LOCAL_LISTENER パラメータをデータベース・パラメータ・ファイルに追加します。

1. LOCAL_LISTENER パラメータの値を決定します。

Oracle8i ホームの LISTENER.ORA ファイルで、デフォルト・リスナーの定義を検索します。この定義内の TCP プロトコルの最初のアドレスを、LOCAL_LISTENER パラメータの値として使用します。

たとえば、デフォルト・リスナーが次のように定義されているとします。

```
LISTENER =
  (DESCRIPTION_LIST=
    (DESCRIPTION=
      (ADDRESS_LIST=
        (ADDRESS=
          (PROTOCOL=TCP)
          (HOST=124.7.56.1)
          (PORT=1521)
        )
      )
    )
  )
```

LOCAL_LISTENER の値は二重引用符を付けて次のようになります。

```
"(ADDRESS=(PROTOCOL=TCP)(HOST=124.7.56.1)(PORT=1521))"
```

2. この LOCAL_LISTENER パラメータをデータベース・パラメータ・ファイルに追加します。

Oracle8i データベースのデータベース・パラメータ・ファイルを検索します。この LOCAL_LISTENER パラメータをファイルの末尾に追加します。前述の例を使用した場合、パラメータ行全体は次のようになります。

```
LOCAL_LISTENER = "(ADDRESS=(PROTOCOL=TCP) (HOST=124.7.56.1) (PORT=1521))"
```

この変更は、データベースが再起動された時点で有効になります。

7.4.6.2.2 グループ内の Oracle8i データベース リスナー情報は、マルチスレッド・サーバー構成の LOCAL_LISTENER パラメータまたは MTS_DISPATCHER パラメータのいずれかに指定できます。

マルチスレッド・サーバー構成で、LOCAL_LISTENER パラメータを使用して完全なリスナー情報を指定する場合（完全なリスナー情報にはホストとポートの両方の値が含まれる）は、Oracle Fail Safe により、「リソースをグループに追加」操作中にマルチスレッド・サーバー構成用のデータベース・パラメータ・ファイルが自動的に更新されます。

グループに追加された Oracle8i データベースは、マルチスレッド・サーバー・モードで稼働します。データベース・パラメータ・ファイルをさらに変更する必要はありません。

次の例は、Oracle Fail Safe により自動的に更新されるマルチスレッド・サーバー構成を示します。

```
mts_dispatcher = "(PROTOCOL=TCP) (DISPATCHERS=1)"
local_listener = "(ADDRESS=(PROTOCOL=TCP) (HOST=124.7.56.1) (PORT=1521))"
```

Oracle8i データベースをグループに追加すると、Oracle Fail Safe では LOCAL_LISTENER パラメータを更新し、そのグループのリスナー情報を使用します。

ただし、マルチスレッド・サーバー構成で MTS_DISPATCHER パラメータを使用して完全なリスナー情報を指定する場合は、MTS_DISPATCHER パラメータからホストとポートの値を削除する必要があります。Oracle Fail Safe では、常に LOCAL_LISTENER パラメータをデータベース・パラメータ・ファイルに書き込みます。

Oracle Fail Safe Manager を使用してデータベースをグループから削除すると、データベース初期化ファイルから LOCAL_LISTENER パラメータが削除されます。[7.4.6.2.1 項](#)の指示に従って、このパラメータをデータベース初期化ファイルに戻す必要があります。

7.4.6.3 マルチスレッド・サーバー構成でのデータベースのオフライン化

マルチスレッド・サーバー構成では、Oracle Fail Safe はマルチスレッド・サーバー・ディスパッチャを介してデータベースに接続します。接続がマルチスレッド・サーバー・ディスパッチャ経由のため、データベースの起動および停止操作はできません。この結果、データベースをオフラインにするには abort モードを使用する必要があります。normal モード、transactional モードまたは immediate モードを使用してデータベースをオフラインにすることはできません。

7.4.6.4 マルチスレッド・サーバー構成でのグループからのデータベースの削除

データベースをグループから削除する場合は、データベース初期化ファイルを手動で更新して、MTS_LISTENER_ADDRESS パラメータのアドレスがデフォルト・リスナーのアドレスを参照するように変更します。

7.5 データベースのセキュリティ要件

Oracle データベースを管理するには、SYSDBA 権限を付与されたデータベース管理者アカウントを使用します。これにより、リモート・クライアントから Oracle データベースを管理できます。

サンプル・データベースを作成する場合や、グループにデータベースを追加する場合は、ユーザー名 INTERNAL とパスワードを指定する必要があります。指定しない場合、MSCS ではデータベースをオンラインまたはオフラインにできません。認証パスワード・ファイルを使用して、初期化パラメータ REMOTE_LOGIN_PASSWORDFILE を SHARED または EXCLUSIVE に設定します。

注意： Oracle Fail Safe では、Windows レジストリの DBA_AUTHORIZATION パラメータを BYPASS の値に設定できません。

データベース管理者の認証と REMOTE_LOGIN_PASSWORDFILE パラメータの詳細は、『Oracle8i 管理者ガイド』を参照してください。

7.5.1 クラスタ・ノード上のパスワード・ファイルの同期化

データベース・パスワード・ファイルはプライベート・ディスクに格納されます。一方のクラスタ・ノードでパスワード・ファイルに対して行った変更内容は、他方のクラスタ・ノード上の対応するファイルに自動的に適用されません。

このため、1つのクラスタ・ノード上のパスワード・ファイルにアカウントを追加した場合は、データベースを実行するよう構成されている他のクラスタ・ノード上のパスワード・ファイルにもそのアカウントを追加する必要があります。パスワード・ファイルに SYS および INTERNAL 以外のアカウントが格納されている場合は、フェイルセーフ・データベース用に、他のクラスタで追加アカウントの SYSOPER および SYSDBA 権限を付与する必要があります。

「リソースをグループに追加」ウィザードを使用してデータベースを追加すると、Oracle Fail Safe Server では、データベースが稼働するよう構成された他のノード上に、そのデータベースのインスタンスが作成され、パスワード・ファイルの最大ユーザー数にはデフォルト値が使用されます。インスタンスが作成されたノード上のパスワード・ファイルには、「リソースをグループに追加」ウィザードで指定した INTERNAL アカウントのパスワードのみが含まれます。

データベースが稼働するよう構成された他のノードで、次の手順を実行して、他のクラスタ・ノード上のパスワード・ファイルを同期化します。

1. パスワード・ファイル内のアカウント数がデフォルトの最大値を超える場合は、新しいパスワード・ファイルを作成します。それ以外の場合は、手順 2 に進みます。

新しいパスワード・ファイルを作成するには、Oracle7、Oracle8 または Oracle8i データベースの管理者ガイドまたは Windows 用マニュアルで、パスワード・ファイルの作成方法の指示を参照してください。

2. データベースを含むグループを、そのデータベースが稼働するよう構成された他のノードに移します。
3. データベースを移した先のノードで、SYS および INTERNAL 以外の権限をアカウントに付与します。
4. データベースが稼働するよう構成された各クラスタ・ノードに対して、手順 2 と 3 を繰り返します。

これで、データベースが稼働するよう構成されたすべてのノードで、パスワード・ファイルのローカル・コピーが同一のものになります。

7.5.2 データベースの INTERNAL パスワードの変更

この項は、「リソースをグループに追加」ウィザードで「このアカウントを使用」を選択し、データベース認証にデータベース・ユーザー・アカウントを指定した場合にのみ適用されます。

INTERNAL アカウントは、Oracle リソース DLL が、データベースにアクセスしてデータベースがアクティブであることを確認するポーリング (Is Alive ポーリング) を実行するために使用されます。また、データベースのオンライン化およびオフライン化にも使用されます。INTERNAL アカウントは暗号化され、クラスタ・メタデータ内に格納されます。このため、データベースの INTERNAL アカウントを変更するときは、クラスタにまたがる情報も更新する必要があります。更新しない場合、Is Alive ポーリングとデータベースの起動および停止に失敗します。

データベースがグループのリソースとして構成されている場合にデータベースの INTERNAL パスワードを変更するときは、Oracle Fail Safe Manager を使用して、対応するデータベース・リソースの認証情報を変更する必要があります。また、データベースが稼働するよう構成された他のノード上の INTERNAL パスワードも変更する必要があります。グループが実行されるノード上で次の手順を実行します。

1. Oracle Fail Safe Manager を使用して、データベースをオフラインにします。
2. データベースのマニュアルに記述されている方法に従って、INTERNAL パスワードを変更します。

Oracle7、Oracle8 または Oracle8i データベースに対応する管理者ガイドを参照してください。INTERNAL パスワードの変更に関する情報を探します。
3. データベースを含むグループを、そのデータベースが稼働するよう構成された他のノードに移します。
4. データベースの可能所有者ノードであるクラスタの各ノードでデータベースを移し、(最初のノードについて手順 2 で説明したとおりに) INTERNAL パスワードを更新するまで、手順 2 および 3 を繰り返します。
5. Oracle Fail Safe Manager で、ツリー・ビューからデータベースを選択し「認証」タブをクリックします。データベースに対する変更と一致するようにパスワード・フィールドを更新します。
6. Oracle Fail Safe Manager を使用して、データベースをオンラインにします。

これで、どちらのノードがデータベースのホストであるかにかかわらず、Oracle Fail Safe で両方のノード上のデータベースにアクセスできます。

7.5.3 Oracle Data Migration Assistant を使用したフェイルセーフ・データベースの移行

この項では、Oracle8i の Data Migration Assistant を使用して異なるリリース間でフェイルセーフ・データベースを移行する方法、あるいは異なる Oracle ホーム間でデータベースを移行する方法について説明します。

移行するすべてのデータベースで、次の手順を実行します。

1. データベースをグループから削除します。
Oracle Fail Safe Manager のツリー・ビューで、移行するデータベースを選択します。その後、次のように選択します。

「リソース」→「グループから削除」

2. フェイルセーフ・データベースの移行先となる Oracle ホームから、Oracle Data Migration Assistant を実行します。
3. 移行するフェイルセーフ・データベースのデータベース・パラメータ・ファイルの場所を把握しておきます。データベースを移行する際に、データベース・パラメータ・ファイルが変換されます。データベース・パラメータ・ファイルがクラスタ・ディスク上にある場合、パラメータ・ファイルの場所は Oracle Fail Safe 用に適切です。データベース・パラメータ・ファイルがプライベート・ディスク上にある場合、Oracle Data Migration Assistant はそのローカル・コピーのみを変換します。この場合、他のクラスタ・ノード上のコピーも編集し、適切な変更を加える必要があります。
4. Oracle Data Migration Assistant では、変換済データベース・ファイルの場所も指定するように求められます。データ・ファイルを現在の場所のままにしておくか、またはローカル・ノードから現在アクセス可能なクラスタ・ディスクを指定します。後者を選択する場合、そのクラスタ・ディスクが他のグループによって使用されていないことを確認してください。
5. Oracle7 データベースを Oracle8 データベースに移行する場合、Oracle Data Migration Assistant はプライベート・ディスク上に Oracle_Home¥database¥mig<SID>.ora という新規データ・ファイル（SID はデータベースのインスタンス名）を作成します。この新規データ・ファイルを、次のようにして他のデータ・ファイルがあるクラスタ・ディスクに移します。
 - a. SVRMGR30 ユーティリティを使用してデータベースを停止します。

```
SVRMGR30> CONNECT username/password@service
SVRMGR30> SHUTDOWN
SVRMGR30> EXIT
```
 - b. ORACLE_HOME¥database¥mig<SID>.ora を、クラスタ・ディスクにコピーします（<SID> はデータベースのインスタンス名）。

- c. SVRMGR30 ユーティリティを使用して、次のコマンドを実行します。

```
SVRMGR30> CONNECT username/password@service
SVRMGR30> STARTUP PFILE=init<SID>.ora MOUNT
SVRMGR30> ALTER DATABASE RENAME FILE
'ORACLE_HOME%\database%\mig<SID>.ora' TO
'cluster_disk%\mig<SID>.ora';
SVRMGR30> SHUTDOWN
SVRMGR30> EXIT
```

6. Oracle Data Migration Assistant によってグループ内のすべてのデータベースが移行されてから、Oracle Fail Safe Manager を使用して次のようにデータベースをグループに戻し、データベースをオンライン化します。
- a. 「リソース」メニューで「グループに追加」をクリックします。
- b. 「リソースをグループに追加」ウィザードの手順に従います。

移行するグループのすべてのデータベースは、同じ Oracle ホームに属している必要があります。Oracle Data Migration Assistant を使用して、あるグループ内の 1 つのデータベースを新規の Oracle ホームに移行する場合、そのグループ内のすべてのデータベースをその新規 Oracle ホームに移行してください。

7.6 可用性を高めるための ctxsrv サーバーの構成

Oracle データベースとともに ctxsrv サーバーを使用している場合、次のようにして ctxsrv サーバーの可用性が高まるように構成できます。

- ctxsrv サーバーを起動し、パーソナリティ・マスクを指定するバッチ・ファイルを作成します。たとえば、次のようなコマンドラインを含む context.bat というファイルを作成します。

```
ctxsrv -user CTXSYS/CTXSYS -personality QDM
```
- MSCS クラスタ アドミニストレータを開き、次の手順で ctxsrv サーバーを可用性の高い汎用アプリケーションとして構成します。
 - 「ファイル」メニューで「新規作成」をクリックし、次に「リソース」をクリックします。
 - 「新しいリソース」ページで次のようにします。
 - 「名前」フィールドで、ctxsrv サーバーの名前を入力します。
 - 必要があれば、「説明」フィールドで ctxsrv サーバーの説明を入力します。
 - 「リソースの種類」フィールドで、「汎用アプリケーション」を選択します。
 - 「グループ」フィールドで、ctxsrv サーバーが関連付けられているデータベースを含むグループを選択します。

- c. 「**実行可能な所有者**」 ページで、ctxsrv サーバーをオンライン化できるクラスタ内のノードを指定します。これらのノードは、ctxsrv サーバーが関連付けられているデータベースの可能所有者と同じである必要があります。
- d. 「**依存関係**」 ページで、ctxsrv サーバーが関連付けられているデータベースをリソースの依存性として指定します。ctxsrv サーバーに（データベースに必要なディスク以外の）ディスク依存性がある場合、そのディスク（複数も可）もリソースの依存性として指定します。
- e. 「**汎用アプリケーション パラメータ**」 ページで次のようにします。
 - 「**コマンドライン**」 フィールドに、手順 1 で作成したバッチ・ファイルのファイル指定を入力します（context.bat など）。
 - 「**現行のディレクトリ**」 フィールドに、ctxsrv サーバーのインストール先ディレクトリを入力します（D:\Orant\bin など）。
- f. 「**レジストリの複製**」 ページで、レジストリ・キーの入力は不要です。「完了」をクリックしてください。

構成を完了すると、クラスタ・サービスによって .bat ファイルが実行され、ctxsrv サーバーのログイン情報を示すコマンド・ウィンドウが開かれます。コマンド・ウィンドウが閉じられた場合、ただちに別のウィンドウが開き、ctxsrv サーバーは通常どおり検索を続行します。ctxsrv サーバーを含むグループがフェイルオーバーしても、操作および検索機能は続行します。

7.7 Oracle Enterprise Manager との統合

Oracle Fail Safe 環境でデータベースを管理および監視するには、Oracle Enterprise Manager を使用します。たとえば、次の目的で Oracle Enterprise Manager を使用します。

- グループ（仮想サーバー）を検出します。

注意： Oracle Enterprise Manager で各グループに構成されたリソースを検出するためには、各グループに対して検出を実行する必要があります。検出された各グループはノードとして Oracle Enterprise Manager のノード・リストに表示されるため、ユーザーはスタンドアロン・リソースを管理するようにグループ内のリソースを管理できます。

- スタンドアロン・データベースと同様にして、グループ内のデータベースにジョブとイベントを作成および登録します。
 - 物理ノードと同様にして、グループにジョブとイベントを作成および登録します。
- 各グループは、Oracle Enterprise Manager では物理ノードとみなされます。このため、グループが検出されると、Oracle Enterprise Manager コンソールのノード・リストに追加されます。

動的検出やジョブ・スケジューリングなど、Oracle Enterprise Manager で利用できるサービスの多くは Oracle Intelligent Agent に依存しています。次の各項では、Oracle Enterprise Manager を使用して Oracle データベースを管理する場合における Oracle Intelligent Agent のグループへの追加、ジョブおよびイベントのスケジューリング、適切な権限の設定方法の詳細を示します。

関連項目：

- Oracle Enterprise Manager との統合およびデフォルトの Intelligent Agent の起動に関する詳細は、Oracle Fail Safe のオンライン・ヘルプを参照
- Oracle Enterprise Manager との統合の問題に関するトラブルシューティングの詳細は [7.11.8 項](#)を参照

7.7.1 Oracle Intelligent Agent のグループへの追加

グループにデータベースを追加した後で、「リソースをグループに追加」ウィザードを再実行し、そのグループに Oracle Intelligent Agent を追加します。ウィザードには、[図 7-6](#) のようなダイアログ・ボックスが表示されます。

図 7-6 「リソースをグループに追加 - リソース」ウィザード・ページ



グループ内のデータベースの数にかかわらず、グループに追加する Oracle Intelligent Agent は 1 つです。ただし、Oracle Intelligent Agent を追加するには、グループに少なくとも 1 つのデータベースが含まれている必要があります。同様に、グループから最後のデータベースを削除するには、まず Oracle Intelligent Agent を削除する必要があります。

グループに Oracle Intelligent Agent を追加するように指定すると、次の操作が実行されます。

- Oracle Fail Safe により、新規の Intelligent Agent が作成されます。
- 新規の Intelligent Agent では、ジョブおよびイベント情報の格納に、(Oracle Intelligent Agent を追加するために「リソースをグループに追加」ウィザードを実行した際に指定した) クラスタ・ディスクが使用されます。
- Oracle Fail Safe Server により、新規の Intelligent Agent がグループの一部として構成されます。
 - Oracle Intelligent Agent が、データベースに関連付けられた仮想アドレスをリスニングするように構成されます。

- Oracle Intelligent Agent はグループとともにフェイルオーバーします。

クラスタ内に構成されている Oracle データベースに対してジョブをスケジュールする方法と、Oracle Enterprise Manager を使用して（フェイルオーバーなどの）イベントを監視する方法の詳細は、Oracle Fail Safe のヘルプを参照してください。

7.8 データベース・リカバリの最適化

可用性が高まるように Oracle Fail Safe で構成された Oracle データベースでは、予定外に発生したシステム・ダウンおよび（ソフトウェア・アップグレードや定期メンテナンスなどの）計画的システム・ダウンの際に、高速フェイルオーバーと高速リカバリが実現できます。Oracle8i のファスト・スタートおよび障害リカバリ機能を利用して、データベース・リカバリに費やされる時間を制御し、Oracle Fail Safe により可用性が高くなるように構成されたデータベースを連続的に監視することができます。

Oracle Fail Safe と Oracle8 および Oracle8i テクノロジーにより、計画的 / 計画外いずれのフェイルオーバーの場合も、一方のノードでデータベースを停止して別のノードでインスタンスを完全にリカバリするまでの所要時間が最適化されます。Oracle データベースのチェックポイント・アルゴリズムにより、計画的および計画外フェイルオーバーでのインスタンスのリカバリ時間が最適化されます。

計画的フェイルオーバーの実行に Oracle Fail Safe Manager（または FSCMD）を使用すると、Oracle Fail Safe Server では、Oracle データベース・サーバーを停止する前にそのチェックポイントを取得します。インスタンス・リカバリを即時に完了し、データベース・クライアントでデータベースをすぐに利用できるように、別のノードのデータベースは制限付きモードで起動されます。（計画的フェイルオーバーの実行に MSCS を使用した場合、データベースが停止される前のチェックポイントの取得は行われません。）

注意： データベースをグループに追加した後、Oracle データベースをオンラインおよびオフラインにするとときは、Oracle Fail Safe Manager または FSCMD コマンドのみを使用してください。それ以外を使用した場合、データベースのチェックポイントの取得は事前に行われません。また、Oracle Fail Safe Manager、FSCMD または MSCS 以外のツールを使用してデータベースをオフライン化すると、Oracle Fail Safe ではそれを障害の発生したりソースとしてみなし、再びオンライン化を試行します。

計画外フェイルオーバーの場合、インスタンス・リカバリの時間はデータベース・リカバリ処理によって制御されます。ファスト・スタート・リカバリ操作の詳細は、Oracle8 および Oracle8i のマニュアルを参照してください。

7.9 フェイルセーフ・データベースに対する管理作業の実行

可用性が向上するように構成されたデータベースに対する管理作業も、他のデータベースと同様にして実行します。ただし、データベースへのアクセスを制限する操作、またはフェイルオーバー機能を一時的に使用禁止にする操作の途中でデータベースをオフラインにする（およびクラスタでのデータベースの監視を停止する）場合は例外で、Oracle Fail Safe Manager または FSCMS コマンドライン・インタフェース（[第 5 章](#)を参照）を使用する必要があります。これには、コールド・バックアップ操作だけでなく、ユーザーがデータベースにアクセス中に実行する必要がある管理操作や、MSCS によるデータベースの定期的な Is Alive ポーリング中の応答時間に影響する操作が含まれます。

Oracle Fail Safe Manager を使用してグループ内に構成されているデータベースに対して管理作業を実行するには、次の手順に従います。

1. Oracle Fail Safe Manager または FSCMD コマンドを使用して、データベースをオフライン化して停止し、クラスタによるデータベースの監視を中止します。データベースに接続されているユーザーはすべて切断されます。
2. Server Manager などのツールを使用し、データベースを起動して管理作業を実行します。
3. データベースが起動されている間、ユーザーはデータベースにアクセスできます。
4. 管理作業の終了後、Server Manager などのツールを使用してデータベースを停止します。
5. Oracle Fail Safe Manager または FSCMD コマンドを使用し、データベースを再度オンラインにします。クラスタによるデータベースの監視が再開されます。

[第 5 章](#)には、FSCMD コマンドを使用してバックアップ操作を実行するスクリプトの例が示されています。

管理作業の途中で、（新しい表領域や関連データ・ファイルの追加など）データベースの構成を変更する操作を行った場合は、「グループの検証」操作を実行します。新規データ・ファイルを追加すると、グループが新しいディスクに依存する可能性があります。「グループの検証」操作を実行すると、ディスクがクラスタ・ディスクであり、別のグループには属していないことが確認されます。そうである場合、ディスクはデータベースと同じグループに追加され、新規ディスクが正常にデータベースとともにフェイルオーバーするよう、クラスタ・レジストリ内の情報が更新されます。

7.10 クライアントおよびクラスタ対応のアプリケーション

フェイルオーバーが発生すると、クライアントおよびアプリケーションとデータベースとの接続がすべて失われ、データベースへの再接続が必要になります。フェイルオーバーの発生時に、正常に機能しているクラスタ・ノードと自動的に再接続するようにアプリケーションを設計することにより、フェイルオーバーはアプリケーションから透過的になります。このクラスタ対応のアプリケーションにより、仮想アドレスとの再接続や、損害発生時に損失した可能性のあるトランザクションの再実行に関する詳細事項がすべて処理されます。使用する Oracle Fail Safe 環境で可用性の高い完全なソリューションを作成するには、アプリケーションの自動再接続を実装して、アプリケーションをクラスタ対応にする必要があります。

この項では、ODBC および OCI アプリケーションをクラスタ対応にして、ユーザーに対するデータベース・フェイルオーバーの影響を最小限にする方法を説明します。この項で説明する項目は次の表のとおりです。

項目	記載されている項
アプリケーションに対して透過的なフェイルオーバーの実現	7.10.1 項
Oracle7 クライアント・アプリケーション	7.10.2 項
Oracle8 および Oracle8i クライアント・アプリケーション	7.10.3 項

7.10.1 アプリケーションに対して透過的なフェイルオーバーの実現

Oracle データベースは、フェイルオーバー後に別のクラスタ・ノード上で自動的に再起動されます。最大限の可用性を実現するには、アプリケーションがデータベースに自動的に再接続できる必要があります。フェイルオーバーしたデータベースに自動的に再接続するように設計されたアプリケーションを、**クラスタ対応のアプリケーション**と呼びます。

7.10.1.1 クラスタ対応のアプリケーション

Oracle8 と Oracle8i の OCI および ODBC アプリケーションがクラスタ対応の場合、アプリケーションでは自動的にエラーをトラップして障害からリカバリし、次のことを実行します。

- サーバーの障害を検出すると、自動的に再接続します。
- コミットされていないトランザクションを再実行します。
- 割り込まれた SQL SELECT 文を再実行して、障害の前にフェッチされた行と同数の行を取り出します。
- クライアントでコミットされた後、サーバーから応答を受け取っていないトランザクションの状態をチェックします。

再接続されたアプリケーションは、フェイルオーバーが発生しなかったかのように動作を継続します。ユーザーはほとんどの場合、サーバー・ノードでの障害発生やデータベースのフェイルオーバーに気が付きません。ただし、フェイルオーバーの発生時にアクティビティが短時間休止して応答時間が遅れることがあります。

7.10.1.2 クラスタ対応ではないアプリケーション

クラスタ対応ではないアプリケーションでは、自動的にリカバリすることはできません。したがって、フェイルオーバーの発生時に、ユーザーは次のことを実行する必要があります。

- 仮想サーバー・アドレスで再接続
- ロールバックされたすべてのトランザクションを再発行

7.10.1.3 クラスタ対応の ODBC および OCI アプリケーション

アプリケーションの再接続は、デフォルトでは発生しません。アプリケーションを変更してエラーを自動的かつ透過的に処理するようにし、クライアント・セッションを存続させたまま障害を回復できるようにします。

Oracle8 と Oracle8i の ODBC および OCI アプリケーションはクライアント状態を保持するように設計されているので、これらのアプリケーションの場合は再接続が比較的簡単です。つまり、ODBC および OCI アプリケーションでは、データベース・リカバリ時の再接続を迅速に処理するために、事前に準備された一連のカーソルを保持し、処理中の SQL SELECT 文を記憶します。

次の項では、Oracle7 および Oracle8 および Oracle8i データベースに接続する ODBC および OCI アプリケーションで透過的なフェイルオーバーを実現する方法について説明します。

7.10.2 Oracle7 クライアント・アプリケーション

Oracle7 を実行するサーバーの場合は、アプリケーションをデータベースに再接続する必要があります。ただし、エラー処理ルーチンをアプリケーションに追加して、フェイルオーバー中に次のエラー・メッセージをトラップし、自動的かつ透過的にクライアントに再接続することができます。

- ORA-01012: ログオンされていません。
- ORA-01033: Oracle の初期化または停止中です。
- ORA-01034: Oracle は使用不能です。
- ORA-01089: 停止処理中 (shutdown immediate) なので操作は許可されません。
- ORA-03113: 通信チャンネルでファイルの終わりが検出されました。
- ORA-03114: Oracle に接続されていません。
- ORA-12203: TNS: 接続先に接続できません。
- ORA-12500: TNS: リスナーは専用サーバー・プロセスの起動に失敗しました。

- ORA-12537:TNS: 接続がクローズされました。
- ORA-12571:TNS: パケット・ライター障害が発生しました。

注意： Oracle エラー・メッセージの詳細（各エラーの原因と推奨ユーザー・アクションを含む）は、『Oracle8i エラー・メッセージ』を参照してください。

前述のエラーはいずれも、フェイルオーバー処理中に、アプリケーションからのデータベース操作の実行要求が失敗したことを示すために返される可能性があります。フェイルオーバー中の場合、データベースがオンラインになってからアプリケーションをサーバーに再接続し、Oracle データベースに対するトランザクション処理を再開することによって、これらのエラーは解決できます。クライアント・アプリケーションで前述のリストにあるエラーのいずれかが検出された場合、次の処理を実行する必要があります。

1. 現在の状態を保存します。
2. データベースに再接続します。
3. 未完了のアプリケーション処理（ロールバックされ、コミットされていないトランザクションなど）を再実行するか、または割り込まれた SELECT 文を再実行します。

7.10.3 Oracle8 および Oracle8i クライアント・アプリケーション

Oracle8 または Oracle8i の OCI および ODBC を使用して記述されたアプリケーションの場合、データベース接続が失われたことを特に意識することなく、自動的に再接続できます。この機能は、Oracle8 OCI の透過的アプリケーション・フェイルオーバー機能により提供されます。Oracle8 および Oracle8i では、Oracle Fail Safe を使用して構成されたデータベースに対する透過的アプリケーション・フェイルオーバー機能が拡張され、Oracle Objects for OLE、JDBC (OCI) および Oracle OCI ライブラリ上に階層を形成するすべてのクライアントがフェイルオーバーの対象になります（いずれの場合もアプリケーション・コードを変更する必要はありません）。

注意： スタンドアロン・データベースの場合とフェイルセーフ・データベースの場合では、透過的アプリケーション・フェイルオーバー機能の実際の処理は異なります。スタンドアロン・データベースの場合、透過的アプリケーション・フェイルオーバーが発生すると、Net8 は別のリスナーに接続して障害の発生した接続を再確立します。フェイルセーフ・データベースの場合、フェイルオーバーが発生すると、Net8 は他のクラスタ・ノードにフェイルオーバーしたものと同じリスナーに再接続します。

スタンドアロン・データベースの場合、透過的アプリケーション・フェイルオーバーのフェイルオーバーという語は、Net8 によってあるリスナーから別のリスナーに接続がフェイルオーバーされることを指します。フェイルセーフ・データベースの場合、透過的アプリケーション・フェイルオーバーのフェイルオーバーという語の使い方に多少誤りがあります。この場合、アプリケーションではなく、アプリケーションに接続されているリスナーのフェイルオーバーによって接続が再確立されます。

実装ではこのような違いがあっても、透過的アプリケーション・フェイルオーバーを管理する方法は変わりません。フェイルオーバーという語が2つの異なる文脈で使用されるため、混乱しないようにしてください。

Oracle8 ODBC ドライバは、Oracle8 OCI インタフェースと Net8 上に階層を形成します。したがって、ODBC クライアントは Oracle8 の機能を自動的に利用できます。

Oracle Fail Safe を使用して構成されているデータベースに接続された状態で透過的アプリケーション・フェイルオーバーを利用するには、Oracle8 クライアント・アプリケーションで次のことを実行します。

- Oracle8 OCI リリース 8.0.4（またはそれ以降）または Oracle8 ODBC リリース 8.0.5（またはそれ以降）を使用します。

Oracle8 リリース 8.0.5 以降では、Oracle8 ODBC ドライバで Oracle8 OCI 透過的アプリケーション・フェイルオーバー機能が使用可能になっています。

- Net8 を介して Oracle8 データベースに接続します。

Oracle7 データベースに接続するアプリケーションでは、[7.10.2 項](#)に記載されている方法を使用する必要があります。

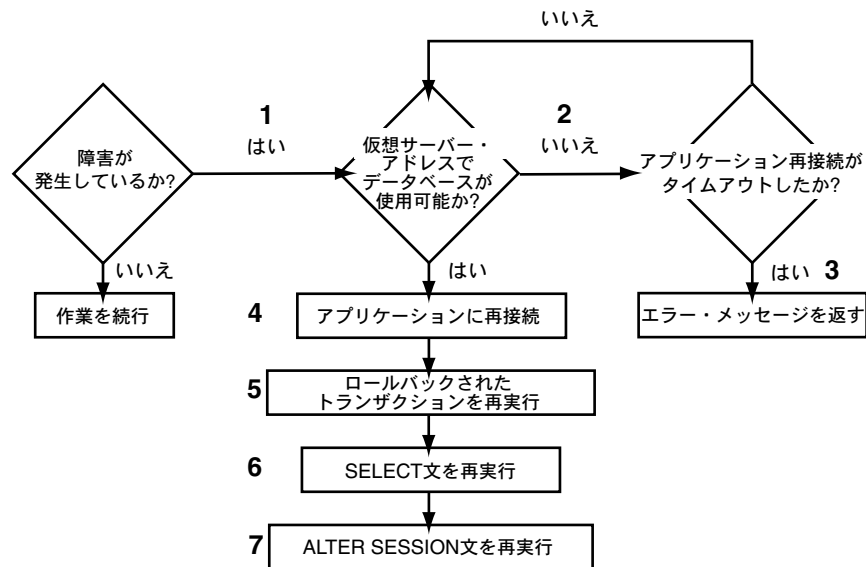
透過的アプリケーション・フェイルオーバーでは、クラスタ・ノードでの障害発生後、クライアントで明示的に再接続する必要はありません。OCI 接続がクライアント・アプリケーションの再接続と状態のリカバリを自動的に処理します。実際、障害の発生時にデータベースを頻繁に更新していた場合を除き、アプリケーション側ではフェイルオーバーの発生に気付かない場合があります。

7.10.3.1 透過的アプリケーション・フェイルオーバーのためのアプリケーション再接続

透過的アプリケーション・フェイルオーバーでは、アプリケーションにサービスを提供できるクラスタ・ノードが存在する限りは、ユーザーは接続が失われたことに気が付きません。

図 7-7 に、OCI フェイルオーバー・プロセスに関連した処理を示します。

図 7-7 フェイルオーバーのフロー・チャート



次に、図 7-7 に示したコールアウトについて説明します。

1. 障害が発生してデータベース接続が失われるまでは、通常どおりに処理が進められます。（アプリケーションでは次回 Oracle データベースによる処理を要求する時点で、接続が失われたことが検出されます。）
2. Oracle8 OCI ライブラリ・ルーチンでは、接続がリストアされるか、アプリケーションの再接続タイムアウトを超えるまで、データベースへの再接続が試行され続けます。
3. 指定時間内にデータベースをリストアできなかった場合、アプリケーションでは状態メッセージを返してユーザーに通知します。
4. Oracle8 OCI ライブラリ・ルーチンにより、仮想アドレスにあるデータベースにアプリケーションが再接続されます。
5. クライアントおよびユーザー・セッションを正常に機能しているクラスタ・ノードに再接続した後、トランザクションを手動で再実行する必要があります。

6. データベース接続を再確立する際、OCI ライブラリ・ルーチンでは割り込まれた SQL SELECT 文をフェイルオーバー後に再実行できます。OCI ルーチンは、フェイルオーバーの前にフェッチされていた数と同じ数の行をフェッチし、フェッチされていた行を破棄します。これは、フェイルオーバー後のフェッチが、元の間合せが割り込まれた箇所から続行されるようにするためです。間合せが再発行されるときは元の間合せスナップショット時刻が使用され、一連の同じ行が確実に返されます。この機能をアクティブにするには、TNSNAMES.ORA ファイルの CONNECT_DATA 文字列で FAILOVER_MODE パラメータに TYPE=SELECT サブパラメータを指定します。
- アプリケーションでは同じ間合せスナップショット時刻を使用するため、次のことが発生します。
- ほとんどの場合、SELECT 文が再実行されるときに行が取り出される順序は決まっています。このため、再実行の可能性のある間合せには ORDER BY 句を指定する必要があります。ただし、ORDER BY 句が指定されていない場合でも、間合せの再実行により返される行はほとんど常に元の順序で返されます。例外としては、ハッシュ結合またはパラレル間合せ機能を使用して実行する間合せなどがあります。Oracle OCI ソフトウェアでは、破棄される一連の行が以前に取り出された行に一致することを透過的に検証し、アプリケーションで不適切な結果が生成されないようにします。
 - TYPE=SELECT サブパラメータを使用した場合、フェイルオーバー後のリカバリ時間が大幅に長くなることがあります。たとえば 100,000 行を取り出す間合せで、99,990 行がフェッチされた後にフェイルオーバーにより割り込まれた場合、クライアント・アプリケーションでは、99,990 行を再フェッチして以前の 99,990 行を破棄し、間合せの最後の 10 行を取り出した後でなければ新規の処理に移ることはできません。
7. フェイルオーバー発生時に複数のセッションが 1 つの接続を使用している場合、OCI ライブラリではすべてのセッションを再接続できます。ただし、いずれかのセッションで ALTER SESSION 文が実行されていた場合は、OCI ライブラリではその文をフェイルオーバー・ノードで再実行することはできません。ALTER SESSION 文はユーザーが再実行する必要があります。

図 7-7 で説明したフェイルオーバーは、アプリケーションが [7.10.3.2 項](#)、[7.10.3.3 項](#) および [7.10.3.4 項](#) で説明されているツールや手法を使用してコーディングされている場合に限り有効です。

7.10.3.2 TNSNAMES.ORA ファイルでの再接続の構成

クライアント・アプリケーションに対して自動再接続を使用可能にするには、Net8 TNSNAMES.ORA ファイルを更新する必要があります。Oracle8（およびそれ以降）の ODBC クライアントと Oracle8（およびそれ以降）の OCI クライアントに対して自動再接続を使用可能にする場合、次の項で説明する変更が必要になります。

新しい接続を可能な限り元どおりに作成するようにします。クライアント（およびユーザー）が変更まったく気付かずに済むのが理想的です。

再接続の構成は、TNSNAMES.ORA ファイルまたは Oracle Names で使用されるネーム・サーバーのいずれかの DESCRIPTION 定義の CONNECT_DATA 文字列で行います。次の項を参照してください。

- **7.10.3.2.1 項**には、グループに構成されているデータベースに対してクライアント接続を使用可能にする方法が説明されています。
- **7.10.3.2.2 項**には、Oracle8 または Oracle8i の透過的アプリケーション・フェイルオーバー機能を構成する方法が説明されています。

Oracle8 ODBC、Oracle8i ODBC、Oracle8 OCI または Oracle8i OCI のいずれのクライアント・アプリケーションでも、必要な更新内容は同じです。

接続操作に関する詳細情報とその構文は、『Net8 管理者ガイド』を参照してください。

7.10.3.2.1 クライアント接続を使用可能にするには グループ内のリソースであるデータベースにアクセスするには、まず、データベースに関連付けられている仮想サーバー・アドレスを使用するようにクライアント・アプリケーションを構成する必要があります。クライアント・システムで TNSNAMES.ORA ファイルを更新し、物理クラスタ・ノードの名前ではなく適切な仮想サーバーの名前を HOST パラメータに指定します。

次に、HOST パラメータで仮想サーバー Virtual-2 を指定した TNSNAMES.ORA ファイルの例を示します。

```
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS= (COMMUNITY=A.WORLD)
      (PROTOCOL=TCP)
      (HOST=Virtual-2) <--- Oracle Fail Safe virtual server name
      (PORT=1521))
    .
    .
    .
  )
)
```

アプリケーションで透過的アプリケーション・フェイルオーバーを使用するかどうかにかかわらず、TNSNAMES.ORA ファイルには正しい仮想サーバーを指定する必要があります。

7.10.3.2.2 フェイルオーバーの構成 透過的アプリケーション・フェイルオーバーのタイプと方法を構成するには、TNSNAMES.ORA ファイルの CONNECT_DATA 文字列に FAILOVER_MODE パラメータを使用します。たとえば、次のようになります。

```
(DESCRIPTION =
  (ADDRESS_LIST =
    .
    .
    .
  )
  (CONNECT_DATA=
    (SID=NH_SALES)

    (FAILOVER_MODE=(TYPE=SELECT)
                  (METHOD=BASIC))
    (RETRIES=20)
    (DELAY=15)
```

SID パラメータは、データベース・インスタンスの一意の名前を示します。

TYPE サブパラメータは、ユーザー・セッションを再確立するかどうか、およびセッションの再確立時にどのアプリケーション操作を自動的にリカバリするかを示します。

表 7-2 に、TYPE サブパラメータのオプションを説明します。

表 7-2 FAILOVER_MODE パラメータの TYPE サブパラメータ・オプション

オプション	再接続？	アプリケーション問合せ	説明
NONE	ユーザー・セッションを再確立しない。	アプリケーション問合せを再実行しない。	透過的アプリケーション・フェイルオーバーを試行せずにユーザーにエラー・メッセージを返す。デフォルトの値。
SELECT	新しいクライアント接続にユーザー・セッションを再確立する。	障害発生前に実行を開始した SELECT 文は、新しい接続で自動的に再実行される。アクティブ・トランザクションはロールバックされ、新しい接続では再実行されない。	障害が発生しなくても、すべての SELECT 文についてクライアント側でオーバーヘッドが発生する。フェイルオーバーが発生すると、フェイルオーバー・ノードで SELECT 文が再実行されます。ユーザーは SELECT 文の再実行に気付きません。
SESSION	新しいクライアント接続にユーザー・セッションを再確立する。	障害発生前に実行を開始した SELECT 文は、新しい接続では再実行されない。	TYPE=SELECT オプションを選択した場合に生じるオーバーヘッドを回避する。

FAILOVER_MODE パラメータには、フェイルオーバー・パフォーマンスの最適化に使用できる METHOD サブパラメータがあります。METHOD サブパラメータには、フェイルオーバー・ノードとの接続の確立方法を定義する、BASIC および PRECONNECT という 2 つのオプションがあります。BASIC サブパラメータはフェイルオーバー時に接続を確立するもので、Oracle Fail Safe では、この接続方法のみがサポートされています。METHOD=BASIC

はデフォルト値であるため、CONNECT_DATA 文字列に METHOD サブパラメータを挿入するかどうかは任意です。

RETRIES サブパラメータは、接続を試行する回数を示します。DELAY サブパラメータを指定した場合、RETRIES サブパラメータのデフォルト値は 5 になります。

DELAY サブパラメータは、接続を試行する間隔の秒数を示します。RETRIES サブパラメータを指定した場合、DELAY サブパラメータのデフォルト値は 1 秒になります。

(RETRIES および DELAYS サブパラメータは、Oracle リリース 8.0.6 および 8.1.6 で採用されました。)

7.10.3.3 Oracle8 および Oracle8i の ODBC クライアント・アプリケーションの再接続

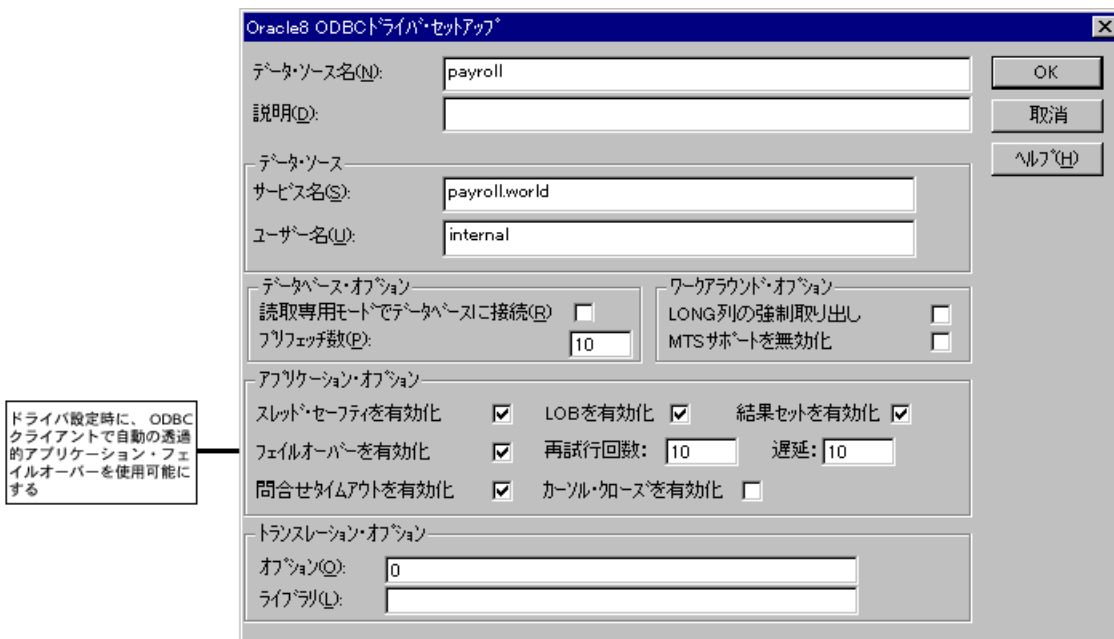
Oracle8 リリース 8.0.5 からは、ODBC ドライバにより、すべての Oracle8 ODBC クライアントで、データベース自動再接続の即時利用とフェイルオーバー後の SQL SELECT 文の再実行ができるようになりました。追加のコーディングは必要ありません。Oracle8 リリース 8.0.5 ODBC ドライバでは、Oracle8 データベースのリリース 8.0.3 以降および Oracle8i データベースに Net8 を使用して接続する ODBC クライアント・アプリケーションに対し、透過的なフェイルオーバーをサポートしています。

Oracle8 ODBC クライアントでデータベースに自動的に再接続するには、次のことを実行します。

1. Net8 TNSNAMES.ORA ファイルを更新し (7.10.3.2 項で説明)、適切な透過的アプリケーション・フェイルオーバー機能を構成します。
2. ODBC データソース アドミニストレータを使用して、データ・ソースの ODBC ドライバ設定中に自動フェイルオーバーを使用可能にします。ODBC データソース アドミニストレータにアクセスするには、次のようにします。
 - a. Windows デスクトップで、「コントロール パネル」を開きます。
 - b. 「コントロール パネル」で「ODBC」をダブルクリックします。
 - c. 「ユーザー DSN」タブで、「追加」をクリックします。
 - d. 「データ ソースの新規作成」ウィザードで、「Oracle ODBC Driver」(バージョン 8.00.05 またはそれ以降) を選択します。
 - e. 「完了」をクリックして「Oracle8 ODBC ドライバセットアップ」を開きます。
 - f. 「Oracle8 ODBC ドライバセットアップ」ダイアログ・ボックスで、各フィールドにデータベース情報を入力し、「フェイルオーバーを可能にする」オプションを選択して自動フェイルオーバーを使用可能にします。

図 7-8 は、Oracle8 ODBC クライアントの自動フェイルオーバーを使用可能にするドライバ・セットアップのダイアログ・ボックスです。

図 7-8 ODBC クライアントの自動フェイルオーバーを使用可能にする



7.10.3.4 Oracle8 OCI クライアント・アプリケーションの再接続

OCI クライアント・アプリケーションの可能なソリューションとしては、(7.10.2 項に説明されているように) OCI エラーをトラップして処理するという方法もありますが、OCI 透過的アプリケーション・フェイルオーバー機能を使用することによって、再接続とフェイルオーバー作業のほとんどを自動的に処理することができるうえ、コーディング作業も削減できます。この新機能を使用できるのは、OCI で作成され、Net8 で Oracle8 データベースに接続されているクライアント・アプリケーションに限られます。Oracle7 データベースに接続するアプリケーションや、データベース・アクセスに Net8 を使用していないアプリケーションでは、この方法は使用できません。

次の項では、透過的アプリケーション・フェイルオーバーを使用可能にするために使用できる OCI プログラミング手法およびコールバック関数について説明します。ある程度の OCI プログラミング知識を有する読者を対象にして説明します。詳細は次の資料を参照してください。

- 『Oracle8i コール・インタフェース・プログラマーズ・ガイド』 および 『Oracle コール・インタフェース for Windows スタート・ガイド』。
- ファイル <Oracle_Home>\%fs%mgr¥sample¥Fsfailover.c 内のオンライン・サンプル C プログラム。

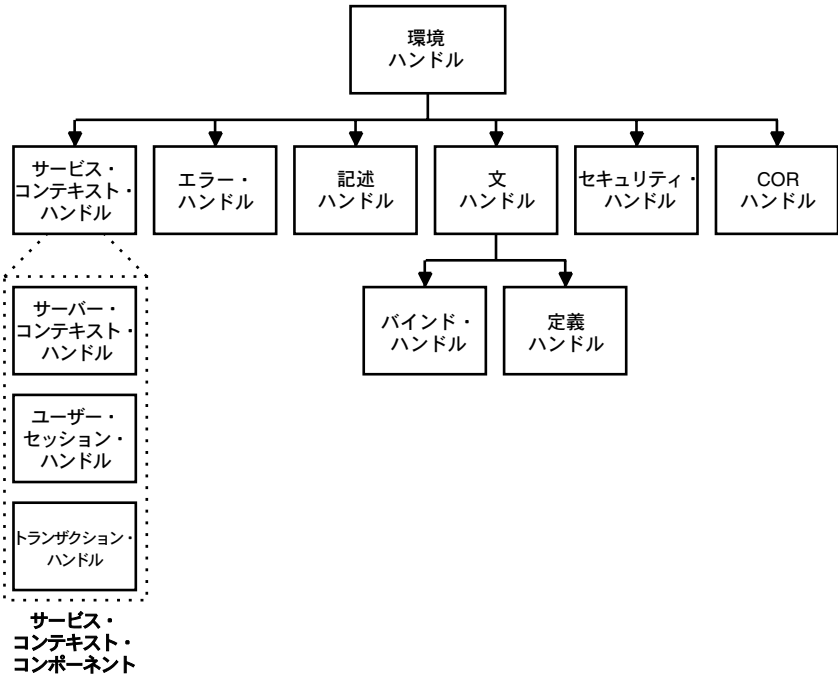
このファイルには、クラスタ対応の OCI クライアント・アプリケーションからの詳細なコード・サンプルが含まれています。

7.10.3.4.1 OCI プログラムの構造 次に、OCI アプリケーション操作の一般的なフローを示します。

- 1. OCI プログラミング環境およびプロセスを初期化します。
- 2. 必要なハンドルを割り当てます。
- 3. サーバー接続およびユーザー・セッションを確立します。
- 4. サーバーに対して SQL 文を発行し、アプリケーション・データを処理します。
- 5. 各文および各ハンドルを解放し、ユーザー・セッションおよびサーバー接続を終了します。

バインド・ハンドルおよび定義ハンドルを除くすべてのハンドルは、特定の環境ハンドルと関連させて割り当てます。図 7-9 に、ハンドルの階層を示します。

図 7-9 OCI ハンドルの階層



7.10.3.4.2 透過的アプリケーション・フェイルオーバーのコールバック関数 透過的アプリケーション・フェイルオーバーのコールバック関数を使用して、フェイルオーバーの発生をユーザーに通知し、現在のアプリケーションの状態を保存するか、あるいはその他のフェイルオーバー関連タスクを実行します。たとえば、フェイルオーバーにより遅延が生じてシステムの応答が突然途絶えたりすると、ユーザーはシステム・ダウンと考え、アプリケーションを取り消す可能性があります。これを回避するために、コールバック関数を使用してフェイルオーバー中であることをユーザーに通知し、しばらく待機するように要求します。

次のコードは、フェイルオーバー・コールバック関数の基本構造を示しています。

```
sb4 callback_fn (dvoid *Asvchp,      <— server context handle
                dvoid *envhp,       <— environment handle
                dvoid *fo_ctx       <— pointer in memory stores state/context
                ub4   fo_type,      <— type of failover: SESSION or SELECT
                ub4   fo_event);    <— indicates why the function was called
```

コールバック関数の使用例は、[7.10.3.5 項](#)に記載されたプログラムを参照してください。

コールバック関数のプログラミングには、次の 2 つの手順が必要です。

1. コールバック関数の作成 ([7.10.3.4.3 項](#)を参照)
2. コールバック関数のサービス・コンテキスト・ハンドルへの登録 ([7.10.3.4.4 項](#)を参照)

7.10.3.4.3 フェイルオーバー・コールバック関数の作成 フェイルオーバー中および新規接続でユーザー・セッションを再確立している間、OCI ライブラリではコールバック関数を何回もコールします。

1. 元の接続が失われてフェイルオーバーが開始された時点で、コールバック関数への最初のコールが実行されます。このコールにより、システム応答が遅れる可能性があることをユーザーに知らせるメッセージを表示できます。
2. フェイルオーバーが完了すると、再び OCI ライブラリからコールバック関数がコールされます。コールバック関数への 2 番目のコールは、次のように、フェイルオーバーが成功した場合と失敗した場合とで異なります。
 - フェイルオーバーが成功した場合、新規接続が再確立されて使用可能になると、コールバック関数がコールされます。以前に発行された ALTER SESSION コマンドでフェイルオーバー時に割り込まれたものがあれば、コールバック関数によりこれらのコマンドが再実行されます。フェイルオーバーが完了し、作業を再開できることをユーザーに知らせるために、コールバック関数をさらにコールする場合があります。
 - フェイルオーバーが失敗した場合、フェイルオーバーが行われなことをアプリケーションに通知するためにコールバック関数がコールされます。

フェイルオーバー・コールバック関数の作成と使用の詳細は、『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を参照してください。

また、[7.10.3.5 項](#)に記載されているプログラムの例も参照してください。

7.10.3.4.4 フェイルオーバー・コールバック関数の登録 フェイルオーバー・コールバック関数を作成した後、その関数を使用するには、サーバー・コンテキスト・ハンドルに登録する必要があります。登録するには、コールバック定義構造を作成し、サーバー・ハンドルの OCI_ATTR_FOBCK 属性をこの構造に設定します。

例 7-1 に、コールバック関数の登録に使用する OCIFobckStruct 構造タイプを示します。

例 7-1 コールバックを登録する OCI の構造

```
typedef struct
{
    OCICallbackFailover callback_function;
    dvoid *fo_ctx;
}
OCIFobckStruct;
```

例 7-1 で、callback_function にはコールする関数のアドレスが含まれ、fo_ctx にはクライアント・コンテキストのアドレスが含まれます。

<Oracle_Home>%fs%smgr¥sample¥Fsfailover.c 内にあるオンラインのサンプル・プログラムに、コールバック関数とコールバック登録の例が示されています。

コールバック関数の登録の詳細は、『Oracle8i コール・インタフェース・プログラマーズ・ガイド』を参照してください。

7.10.3.5 透過的アプリケーション・フェイルオーバーのデモ・アプリケーション

Oracle Fail Safe では、Oracle Fail Safe 構成で ODBC または OCI を使用するクライアント用の透過的アプリケーション・フェイルオーバーを示す実行可能デモが提供されています。このプログラムとそのドキュメント (readme.txt) は、<Oracle_Home>%fs%smgr¥sample ディレクトリにあり、オンラインで使用できます。

ODBC クライアント用の透過的アプリケーション・フェイルオーバーのデモには、最低限 Oracle8 リリース 8.0.5 の ODBC ドライバが必要です。

7.11 可用性の高い Oracle データベースに関連するエラー処理と問題のトラブルシューティング

次の各項では、Oracle Fail Safe が可用性の高いデータベースのオンライン化を試行した際にエラーが発生した場合に、そのエラーを処理するスクリプトの指定方法と、可用性が高まるよう構成された Oracle データベースで発生する特定の問題のトラブルシューティングの方法を説明します。Oracle データベースのトラブルシューティングに関する一般的な情報は、Oracle データベース・サーバーのマニュアルを参照してください。

7.11.1 データベースをオンラインにする際に発生するエラーの処理

Oracle Fail Safe がデータベースのオンライン化を試行した際に発生するエラーを処理するスクリプトを指定できます。

エラー処理スクリプトを指定するには、次のようにします。

1. エラーを処理するスクリプトを作成します。
2. スクリプトに FsDbError.bat という名前を付けます。
3. スクリプトが成功した場合は 0 を返し、失敗した場合は 0 以外の整数を返すことを確認します。
4. データベース・リソースの可能所有者である各クラスタ・ノードの次のディレクトリにスクリプトを置き、ファイル所有者がそのクラスタ・ノードに対するローカル管理者権限を持っていることを確認します。

```
<Oracle_Fail_Safe_Home>%fs%fssvr%scripts
```

Oracle Fail Safe は、データベースをオンラインにできない場合にスクリプトを実行します。クラスタ上のすべての Fail Safe データベースに同じスクリプトを使用します。データベースをオンラインにできない場合、Oracle Fail Safe は次のように、エラー・コード、データベース・サービス名、データベース SID およびデータベース・パラメータ・ファイル指定などをスクリプトに渡します。

```
FsDbError.bat ORA-01113 0817DB.WORLD 0817DB D:%Ora%admin%0817DB%pfile%init.ora
```

Oracle Fail Safe は、データベースを再起動できない場合に、スクリプトを実行するプロセスを生成します。このプロセスは、データベース・リソースの保留タイムアウト値で指定された期間、スクリプトの終了を待機します。スクリプトが保留タイムアウト期間内に終了しない場合、スクリプトは終了させられます。

Oracle Fail Safe は、イベントのログを Windows イベント ログに記録し、スクリプトが成功したか、失敗したか、Oracle Fail Safe によって終了させられたかを示します。スクリプトが失敗した場合は、エラー・コードもイベント ログに書き込まれます。

スクリプトが失敗した場合、Oracle Fail Safe は、データベースの再起動ポリシーおよびフェイルオーバー・ポリシーの定義に従ってデータベースのオンライン化の試行を継続します。

7.11.2 問題のトラブルシューティング

多くの場合、問題のトラブルシューティングでは、まず「クラスタの検証」、「グループの検証」、あるいは「スタンドアロン・データベースの検証」コマンドを発行します。第 6 章に、これらのツールの一般的な説明を記載しています。

データベースを含むグループに対して「グループの検証」コマンドを発行すると、Oracle Fail Safe では次のことを行います。

- グループ内の各データベースに問い合せて、使用されているディスクを特定します。その後、そのディスクがクラスタ・ディスクであるか、すでにグループに追加されているかをチェックします。（たとえば、可用性が高まるように構成された後にデータベースにディスクが追加されたために）ディスクの有効性検査が失敗した場合、「グループの検証」操作では、問題を修正する前に確認を求められます。
- ディスク・ドライブの変更を検出し、必要に応じてリソースの依存性を更新します。
- ネットワーク名が正しい IP アドレスに ping するかどうかを検証します。
- Net8 または SQL*Net 構成が正しいかどうかを確認します。
- グループ内のリソースの構成に誤りがあれば修正します。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- グループに構成されているデータベースにディスクを追加する場合
- クラスタに新しいノードが追加された場合

たとえば、Oracle Fail Safe Manager を使用してクラスタ構成を更新せずに、データベースに新しいディスクを追加した場合を考えてみましょう。その後サーバー・ノードが停止しても、フェイルオーバーは適切に実行されません。これは、クラスタ・ソフトウェアに構成の変更を通知していないことによるものです。この問題を回避するため、データベースの構造を変更したときは、必ずそのデータベースを含むグループを検証してください。グループを検証すると、Oracle Fail Safe では自動的に変更を検出し、クラスタ構成を更新します。前述の例では、Oracle Fail Safe は、ユーザーに代わって新しいディスクをグループに追加します。

「グループの検証」で問題が検出された場合、Oracle Fail Safe は問題を修正するための応答を求めるか、または問題を詳しく説明したエラー・メッセージを返します。

7.11.3 データベースのグループへの追加の問題

グループへのスタンドアロン・データベースの追加時に問題が発生した場合は、次の操作を実行します。

- 「スタンドアロン・データベースの検証」操作を実行します（[6.1.3 項](#)を参照）。

「スタンドアロン・データベースの検証」操作を実行することにより、データベースが有効に機能するスタンドアロン・データベースであることが検証されます。

たとえば、スタンドアロン・データベースをグループに追加しようとしたが、ネットワーク（SQL*Net または Net8）を構成する際にエラーが発生したとします。この場合は、Oracle Fail Safe によってクラスタワイド操作がロールバックされ、データベースはスタンドアロン・データベースのままになります。この問題を解決するには、次の手順を実行します。

1. 「クラスタの検証」操作を実行して、クラスタ・ネットワーク構成が適切かどうかを確認します。
2. 「スタンドアロン・データベースの検証」操作を実行して、ネットワーク（Net8 または SQL*Net）が機能していることを確認します。
3. スタンドアロン・データベースをグループに追加してみます。
4. データベースをグループに追加する操作に失敗した場合は、[7.11.7.4 項](#)に記述されている方法に従って、SQL*Net または Net8 のロールバック・ファイルを調べてください。

- 次のことを確認します。

- データベース・ファイルが共有クラスタ・ディスク上にあること

Oracle Fail Safe では、構成内の各ディスクについて、ディスクが共有ディスク・インターコネクト上にあるかどうかを判断します。データベース・ファイルがクラスタ化されていないディスク上にある場合は、そのデータベースを移動して共有クラスタ・ディスク上に置く必要があります。

- 「リソースをグループに追加」ウィザードで次の情報を正しく指定してあること
 - * データベースのアクセスに使用するユーザー名とパスワード
 - * データベースのパラメータ・ファイル
 - * （ネット）サービス名
 - * データベース名
 - * インスタンス名

7.11.4 グループのオンライン化の問題

データベースが含まれるグループのオンライン化の際に問題が発生した場合は、次のことを行ってください。

- グループを検証します。

(Oracle Fail Safe Manager の「トラブルシューティング」メニューから)「グループの検証」コマンドを使用すると、Oracle Fail Safe ではグループの構成をチェックし、問題が見つかったら修正を行います。「グループの検証」コマンドで問題を修正できない場合は、その問題を手動で解決するヒントとなるエラー・メッセージが返されます。

「グループの検証」コマンドで問題が見つかった場合には、その後の手順についてユーザーの応答が求められます。

- SQL*Net または Net8 のリスナー・ログを確認します。

SQL*Net または Net8 では、エラーの検出や、リスナーを介したデータベースへのアクセスのたびに、リスナー・ログ・ファイルにエントリを作成します。このログ・ファイルの中に、問題の識別に役立つエラーがないかどうかを調べます。

- Oracle7 システムでは、ログ・ファイルは次の場所にあります。

`<Oracle_Home>\Network\Log`

- Oracle8 または Oracle8i システムでは、ログ・ファイルは次の場所にあります。

`<Oracle_Home>\Net80\Log`

- データベースのネット・サービス名を調査します。

Oracle Fail Safe データベース・リソース DLL では、Is Alive ポーリング間隔でグループ内の各データベースにアクセスします。(Is Alive ポーリング間隔は、Oracle Fail Safe Manager 内のデータベースの「フェイルオーバー」プロパティのページに表示されます。) DLL では、データベース接続情報を使用してデータベースにアクセスします。データベース・アクセス情報が変更されている場合、データベースへのアクセスに失敗します。したがって、MSCS ではデータベース・リソースを alive ではないと見なします。

- SQL*Net または Net8 の構成データを確認します。

サーバー・システム上の LISTENER.ORA ファイル、およびサーバーとクライアントの両方のシステム上の TNSNAMES.ORA ファイルには、クラスタ内のグループに対する有効な仮想サーバー・アドレスが含まれていなければなりません。

- グループ内の各リソースを個別にオンライン化します。

グループ内に複数のデータベースが存在する場合、この方法で問題の原因となっているデータベースを識別できます。

- データベースの保留タイムアウト値が十分であることを確認します。

データベースを含むグループがオンライン化に失敗する、または頻繁にフェイルオーバーする場合は、保留タイムアウト値が正しく設定されているかチェックしてください。

い。データベースの保留タイムアウト値の設定が低すぎると、オンライン化の失敗、頻繁なフェイルオーバーが発生します。

保留タイムアウト値には、クラスタ・ソフトウェアでデータベースがオンライン（またはオフライン）になってから操作が失敗したと判断されるまでの時間を指定します。応答時間の長さをクラスタ・システムにより使用不能状態と判断される可能性を避け、かつ障害発生時のフェイルオーバー応答時間を最小限に抑えられるよう、適切な値を設定してください。

保留タイムアウト値を設定するには、次のようにデータベース・プロパティを変更します。

1. Oracle Fail Safe Manager のツリー・ビューで、データベース名を選択します。
 2. 「ポリシー」タブをクリックします。
 3. 「保留タイムアウト」ボックスで保留タイムアウト値を変更します。
- データベース認証アカウントが、パスワードを伴う INTERNAL アカウントであることを確認します。さらに、データベースのパスワード・ファイル内の初期化パラメータ REMOTE_LOGIN_PASSWORDFILE が、SHARED または EXCLUSIVE に設定されていることを確認します。
 - Oracle Fail Safe でデータベースへのアクセスに使用するアカウントのパスワードが変更されている場合は、Oracle Fail Safe Manager でその変更を更新します。

Oracle Fail Safe でデータベースへのアクセスに使用するアカウントのパスワードを変更後、Oracle Fail Safe Manager でその情報を更新しない場合、データベースに対するポーリングは失敗します。Oracle Fail Safe のデータベース・パスワードの変更を更新する方法の詳細は、[7.5.2 項](#)を参照してください。

7.11.5 データベース管理者の認証（Internal としての接続）

Oracle Fail Safe でデータベースをオンラインまたはオフラインの状態にしようとして問題が発生することがあります。この問題の原因として、データベース管理者の認証の設定方法に問題があることが考えられます。この問題を解決するには、次の方法を実行してください。

- 「リソースをグループに追加」ウィザードで「このアカウントを使用」を選択し、データベース・ユーザー・アカウントを指定した場合は、データベース初期化パラメータ・ファイルで REMOTE_LOGIN_PASSWORDFILE 初期化パラメータが SHARED または EXCLUSIVE に設定されていることを確認します。

「リソースをグループに追加」ウィザードで「OS 認証を使用」を選択した場合は、データベース初期化パラメータ・ファイルで REMOTE_LOGIN_PASSWORDFILE 初期化パラメータが NONE に設定されていることを確認します。

[7.5 項](#)に、データベース管理者認証に対してこのパラメータを適切に設定する方法を説明します。

- Oracle Fail Safe でグループ内のデータベースにアクセスできることを確認します。

グループの検証や、データベースがオンライン状態であることを確認するためのポーリングなど、Oracle Fail Safe Server により実行される一部の操作では、Oracle Fail Safe でグループ内にあるデータベースにアクセスする必要があります。データベースをグループに追加したときにデータベース・アカウント・パスワードを正しく指定しなかった場合、あるいはそのパスワードが変更された場合は、それを更新する必要があります。この更新を行わないと、Oracle Fail Safe ではそのデータベースをオフラインにします。

7.5.2 項には、データベース・パスワードの正しい更新方法が説明されています。

Windows システムでのデータベース管理者アカウントのパスワードを変更する場合は、Oracle Fail Safe Manager で使用されるパスワード情報も更新する必要があります。変更しないと、データベースのクラスタ・ポーリングが失敗し、そのデータベースはオフラインになります。

7.11.6 サンプル・データベースの問題

サンプル・データベースの作成時または削除時にエラーが発生することがあります。この場合には、次の点を確認してください。

- 「サンプル・データベースの作成」コマンドで「サンプル・データベースの作成」スクリプト・ファイルを開くことができない場合は、Oracle Fail Safe ソフトウェアを使用してそのサンプル・データベース・ファイルを再インストールするか、またはファイルを CD-ROM から直接アクセスします。その後、再度「サンプル・データベースの作成」操作を実行します。サンプル・データベース・ファイルのインストール方法の詳細は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
- Oracle8 または Oracle8i 用の Oracle Fail Safe サンプル・データベース・ファイルがクラスタにインストールされていないまま、Oracle8 または Oracle8i のサンプル・データベース作成を試みたり、Oracle Fail Safe のインストール後に Oracle データベース・ソフトウェアをインストールすると、次のエラー・メッセージが表示されます。

FS-10270: Oracle Fail Safe サンプル・データベースのファイルがインストール・ディレクトリまたは CD-ROM に存在しません。

Oracle Fail Safe のインストール・プログラムを実行して Oracle データベースのサンプル・ファイルをインストールするか、または Oracle Fail Safe CD-ROM をロードしてからでないと、サンプル・データベースは作成できません。Oracle Fail Safe のインストール時には、サンプル・データベースの作成先のクラスタ・ノードに現在インストールされているデータベース・リリースに対応するサンプル・データベース・ファイルのみがインストールされます。

Oracle Fail Safe をインストールした後で Oracle データベース・サーバーのバージョンをインストールする場合は、インストールした Oracle データベース・サーバーのバージョンに対応するサンプル・データベース・ファイルのみをインストールします。

- 「サンプル・データベースの削除」コマンドが失敗した場合は、サンプル・データベースではないデータベースを選択している可能性があります。選択したデータベースがサンプル・データベースであることを確認してから、削除操作をもう一度実行してみてください。

Oracle Fail Safe では、サンプル・データベースの情報は、Windows レジストリ内に HKEY_LOCAL_MACHINE¥SOFTWARE¥ORACLE¥FailSafe¥SampleDB キーで格納されます。

削除したサンプル・データベースが Oracle Fail Safe Manager のツリー・ビューに表示される場合、Windows レジストリからそのデータベースのエントリが消去されているかどうか確認してください。

7.11.7 仮想サーバーの構成で検出される問題

スタンドアロン・データベースまたはグループに構成されているデータベースに接続しようとして問題が検出された場合は、データベースの Net8（または SQL*Net）構成を調べる必要があります。

Oracle Fail Safe では、Net8（または SQL*Net）構成の検証と修復を支援する「グループの検証」および「スタンドアロン・データベースの検証」操作が提供されています。詳細は、[6.1.2 項](#)および [6.1.3 項](#)を参照してください。

7.11.7.1 仮想アドレスの構成で発生する問題

Oracle Fail Safe は仮想アドレス情報を構成するとき、Net8 または SQL*Net の LISTENER.ORA ファイルおよび TNSNAMES.ORA ファイルを変更し、リスナーを停止してから再起動します。発生する可能性のある問題と問題を解決するための処置を、次のリストに示します。

- FS-10070: Net8（または SQL*Net）: %s

このメッセージ・コードは、Net8 の LISTENER.ORA および TNSNAMES.ORA ファイルの解析（読み込みまたは更新）に問題があることを報告するものです。

- 不適切な更新やファイルの破損によりファイルが無効となった場合、Oracle Fail Safe ではこれらのファイルを使用して仮想サーバー情報を構成できません。有効なバージョンのファイルを取り出すか、Oracle Net8 Assistant を使用してファイルを作成し直します。
- これらのファイルが有効な場合は、操作で使用されているグループのネット・サービス名、データベース SID およびネットワーク名が正しいかどうかをチェックします。情報が不適切であると、仮想サーバーの構成でエラーが生じることがあります。同じデータベース SID が、2 つ以上のリスナーに含まれていないことを確認してください。複数の Oracle ホームを伴ったシステムでは、すべての LISTENER.ORA ファイルをチェックします。

- FS-10066: Net8（または SQL*Net）リスナーの NT サービス <name> の起動に失敗しました

リスナーの定義変更や新規リスナーの定義作成後、Oracle Fail Safe ではリスナーが起動されます。

多くの場合、このエラーは、別のリスナーがすでにデータベースをリスニングしていることが原因で発生します。システム上で 1 つのリスナーに限り、特定のアドレスまたはデータベース SID のリスニングが可能です。たとえば、LISTENER_A が次のような定義の場合、システム上のその他のリスナーでは、IPC プロトコルを使用したキー ORCL のリスニング、TCP プロトコルを使用したホスト server_A でのポート 1521 のリスニング、ORCL SID 名のリスニングができません。

```
LISTENER =
  (ADDRESS_LIST=
    (ADDRESS=
      (PROTOCOL=IPC)
      (KEY=ORCL)
    )
    (ADDRESS=
      (PROTOCOL=TCP)
      (Host=server_A)
      (Port=1521)
    )
  )

SID_LIST_LISTENER =
  (SID_LIST=
    (SID_DESC=
      (SID_NAME=ORCL)
    )
  )
```

他のリスナーで LISTENER_A と同じアドレスまたはデータベース SID を使用しようとすると、起動に失敗します。

この問題が発生すると、Oracle Fail Safe では、更新されたリスナーおよびネット・サービス名の定義が含まれた LISTENER.ORA ファイルおよび TNSNAMES.ORA ファイルをロールバック・ファイルとして保存します。ロールバック・ファイルのファイル名は、<filename>_RLB.ORA です。

ロールバック・ファイルを読み込んでリスナー定義を探し、システム上で稼働している他のすべてのリスナー定義と照合する必要があります。リスナー間でアドレスまたはデータベース SID 名が重複しないようにしてください。ここでも、複数の Oracle ホームを使用するシステムでは、すべての LISTENER.ORA ファイルをチェックしてください。

- リスナーを起動できない一般的な原因のもう 1 つは、仮想アドレスです。リスナーが使用する仮想アドレスは、Oracle Fail Safe がリスナーを起動しようとしているノード上でアクティブでなければなりません。

ネットワーク構成問題のトラブルシューティングに関する詳細は、Net8 のマニュアル（ログ・ディレクトリの情報を含む）を参照してください。

7.11.7.2 リスナー作成時の問題

Oracle Fail Safe Manager では、Listener 制御ユーティリティ（LSNRCTL）を使用して新規リスナーを作成し、Oracle ホームにあるファイルに出力します。

たとえば、Oracle ホームとネットワーク・ディレクトリのパスが C:\ORANT\NET80\ADMIN の場合、リスナー出力ファイルは、次のディレクトリおよびファイルに書き込まれます。

```
C:\ORANT\NET80\LOG\fslnode.out
```

各リスナーには、リスナー名に .out 拡張子が付いた名前の出力ファイルがあります。（この例では、リスナー名は fslnode です。）新規リスナーの作成が困難な場合は、出力ファイルを使用して問題を診断できます。

7.11.7.3 アーカイブされた LISTENER.ORA または TNSNAMES.ORA ファイル

Oracle Fail Safe が LISTENER.ORA または TNSNAMES.ORA を変更すると、そのファイルの元のバージョンが毎回アーカイブされます。Oracle Fail Safe によって変更される前の SQL*Net または Net8 のネット・サービス名定義やリスナー定義を参照する必要がある場合は、構成ファイルのアーカイブ・バージョンを調べます。

Oracle Fail Safe は、アーカイブされたバージョンの構成ファイルを 2 つ前の分まで保持します。アーカイブされたバージョンのファイル名の形式は、<filename>_000.ORA および <filename>_001.ORA となります。<filename>_000.ORA が最新のファイルです。

7.11.7.4 ロールバック・ファイル

SQL*Net または Net8 構成ファイルの変更後、Oracle Fail Safe である操作の実行中にエラーを検出すると、そのファイルの更新済バージョンが <filename>_RLB.ORA として保存されます。その後、ファイルの元のバージョンがリストアされます。

このロールバック・バージョンのファイルは、問題の診断に役立つことがあります。

7.11.8 セキュリティ・アクセスおよび認証の問題

アクセスおよび認証の問題は、Oracle Enterprise Manager を介して操作を実行する場合に最も多く発生します。

次に、典型的な認証の問題をいくつか示します。

- Oracle Fail Safe への接続時に、Oracle Enterprise Manager から次のようなエラーが発生
FS-10101: クラスタ上のユーザー <ユーザー名> の認証に失敗しました。

Oracle Enterprise Manager コンソールで、クラスタのユーザー設定項目がすべてのクラスタ・ノードに対する Windows 管理者の設定になっており、ユーザ名とドメインが正しく指定されていることを確認してください。(Oracle Enterprise Manager のリリース 2.0 または 2.1 では、クラスタの優先接続情報を指定できないため、適用されません。)

- Oracle Enterprise Manager から Oracle Fail Safe に送ったジョブが、「ユーザーの認証に失敗しました」というエラーで失敗
 - クラスタ内のそれぞれのノード上で、「バッチ ジョブでログオン」のアクセス権限を使用して設定された Windows アカウントを持っていることを確認します。
 - Oracle Enterprise Manager で、クラスタ内の各ノードに対するユーザー設定項目が、クラスタ内の各ノードのローカル・アカウントのユーザー名とパスワードと一致していることを確認します。
- グループ内に構成されているデータベースに対して操作やアクセスを実行すると、「ORA-01031: 権限が不十分です」というエラーが発生
 - サンプル・データベースの作成、グループへのデータベースの追加を実行する場合は、データベースの認可情報にパスワードと INTERNAL アカウントが使用されていることを確認します。
 - Oracle Enterprise Manager からデータベースにアクセスする場合、各データベースのユーザー設定項目がデータベースの INTERNAL アカウントと一致していることを確認します。

7.11.9 クライアントがデータベースにアクセスできない

ユーザーおよびクライアント・アプリケーションがグループ内に構成されているデータベースにアクセスできない場合は、次の手順を実行して問題を修正します。

1. グループに対して仮想サーバーを使用するように、TNSNAMES.ORA ファイルを更新します。
2. 「グループの検証」コマンドを実行して、ネットワーク (Net8 または SQL*Net) 構成の有効性を検査します。

可用性を高めるための Oracle WebDB リスナーの構成

Oracle WebDB は、Web データベース・アプリケーションおよびコンテンツ・ドリブン Web サイトの構築、配置、監視に有効です。Oracle WebDB を使用すると、迅速かつ容易に Web から Oracle データベースを使用できるようになります。

Oracle Fail Safe は、可用性の高い Oracle WebDB リスナーを構成する際に役立ちます。あるクラスタ・ノードが停止または障害を起こした場合、WebDB リスナーを含むグループは別のクラスタ・ノードに自動的にフェイルオーバーし、クライアントは、通常、数秒以内に WebDB リスナーおよび Web サイトへのアクセスを再開できます。Web クライアントは、ブラウザの「再読み込み」または「更新」ボタンをクリックするか、または Oracle WebDB リスナーの（仮想アドレスを指す）URL を再入力して、Oracle WebDB リスナーに再接続します。

この章では、次の項目について説明します。

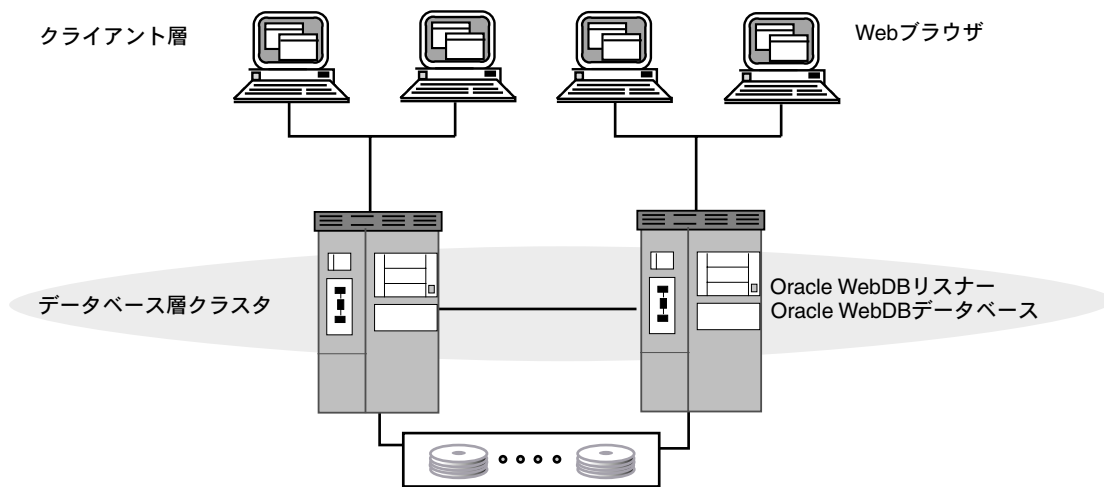
項目	参照
概要	8.1 項
スタンドアロン Oracle WebDB リスナーの検出	8.2 項
Oracle WebDB リスナーのグループへの追加	8.3 項
Oracle WebDB リスナーへのクライアント接続	8.4 項
Oracle WebDB リスナーまたは PL/SQL 設定の変更	8.5 項
Oracle WebDB リスナーのセキュリティ要件	8.6 項
Oracle WebDB のインストールに関するヒント	8.7 項
可用性の高い Oracle WebDB リスナーに関連する問題のトラブルシューティング	8.8 項

この章では、Oracle Fail Safe を使用して Oracle WebDB リスナーの可用性を高める方法について説明していますが、インターネット・ソリューション全体の可用性を高めるには、Oracle WebDB をインストールしたデータベースの可用性およびアプリケーション（Oracle Forms、Oracle Reports、Oracle Applications など）の可用性も向上させる必要があります。Oracle WebDB をインストールしたデータベースは、他の Oracle データベース・サーバー同様に構成します。可用性が向上するように Oracle データベース・サーバーを構成する方法の詳細は、[第 7 章](#)を参照してください。

8.1 概要

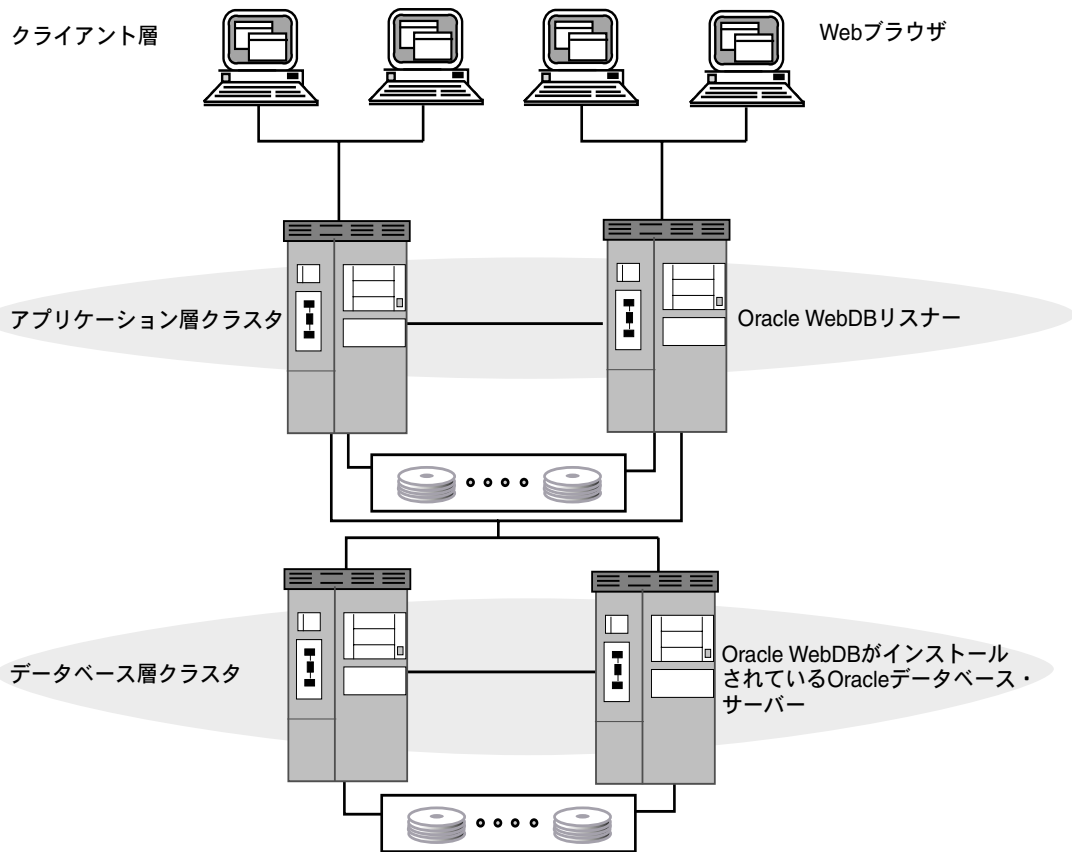
[図 8-1](#) に、データベース層に可用性が高まるように WebDB リスナーおよび WebDB データベースを構成した、2つのノードから成るクラスタを示します。これは、典型的な Oracle WebDB 構成です。

図 8-1 可用性の高い Oracle WebDB 構成のための 2 層アーキテクチャ



[図 8-2](#) に、アプリケーション層に可用性が高まるように Oracle WebDB リスナーを構成し、データベース層に Oracle WebDB データベースを構成し、クライアント層に Web ブラウザをインストールして 3 層アーキテクチャを作成した、2つのノードから成るクラスタを示します。この構成は、Oracle Applications のインストール時など、他のアプリケーションの可用性も高める場合に使用できます。

図 8-2 可用性の高い Oracle WebDB 構成のための 3 層アーキテクチャ



8.2 スタンドアロン Oracle WebDB リスナーの検出

Oracle Fail Safe Manager では、スタンドアロン Oracle WebDB リスナーを検出できません。可用性が高まるように Oracle WebDB リスナーを作成して構成するには、Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用します。

8.3 Oracle WebDB リスナーのグループへの追加

可用性が高まるように Oracle WebDB リスナーを構成するには、Oracle WebDB リスナーを少なくとも 1 つの仮想アドレスが含まれているグループに追加します。Oracle Fail Safe により、Oracle WebDB リスナーが依存する他のすべてのリソースが追加されます。通常、このグループには次のリソースが含まれます。

- 仮想アドレス（IP アドレスとネットワーク名）。
- Oracle WebDB リスナー。
- Oracle WebDB 構成ファイルが常駐する全クラスタ・ディスク。
- また、Oracle WebDB リスナーによりアクセスされる Oracle WebDB データベース、およびデータベースの依存リソースが含まれる場合があります。

8.3.1 構成前に

Oracle WebDB リスナーをグループに追加する前に、次のことに注意してください。

- Oracle WebDB リスナーをグループに作成して追加する場合、仮想アドレスおよびポートの組合せは、別コンポーネント（別の Web サーバーなど）で使用されていないものにします。Oracle Application Server をインストールした場合、作成したデフォルト Web リスナーで使用されている組合せも使用できません。詳細は、[12.7.1 項](#)を参照してください。
- Oracle WebDB リスナーは、Oracle WebDB リスナーの可能所有者となる全クラスタ・ノードにインストールします。詳細は、[8.7 項](#)を参照してください。
- グループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

8.3.2 Oracle WebDB リスナーの構成手順

Oracle Fail Safe Manager により、Oracle WebDB リスナーの可用性が高まるように必要な手順が自動化されます。Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードで、リソース・タイプに Oracle WebDB リスナーを指定してください。

[表 8-1](#) に、可用性が高まるように Oracle WebDB リスナーを構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプとチュートリアルを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択するか、または「ヘルプ」→「チュートリアル」を選択します。

表 8-1 Oracle WebDB リスナーの構成手順

手順	処置	説明
1	Oracle WebDB リスナーの可能所有者となる各クラスタ・ノードのプライベート・ディスク上に、Oracle WebDB リスナーがインストールされていることを確認	インストールの詳細情報は、Oracle WebDB のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に Oracle WebDB リスナー・コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle WebDB リスナー・コンポーネントを選択します。これで DLL ファイルをインストールします。詳しい指示は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
3	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」
4	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
5	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）
6	グループへの Oracle WebDB リスナーの追加	「リソース」→「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い Oracle WebDB リスナーを作成および構成する際に役立ちます。
7	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
8	その他のアプリケーション・コンポーネントをフェイルオーバー用に構成	その他のアプリケーション・サーバーおよびデータベース・サーバーを Oracle Fail Safe を使用して構成し、Web ベースのアプリケーション全体の可用性が高くなるようにします。

8.3.3 Oracle WebDB リスナーの構成データ

Oracle Fail Safe Manager には、可用性が向上するように Oracle WebDB リスナーを構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- Oracle WebDB リスナーを実行する仮想アドレス
- Oracle WebDB リスナーのクラスタ内で一意の名前
- 仮想アドレスと組み合わせて、クラスタ内の一意のアドレスを指定するポート番号
- Oracle Fail Safe で Oracle WebDB リスナー構成ファイルを作成するクラスタ・ディスク
- Oracle WebDB リスナーでアクセスするデータベース・サーバー名、データベースにアクセスするためのアカウント名およびパスワード（オプション）

Oracle WebDB リスナーをグループに追加するクラスタワイド操作は、次の場合に失敗します。

- 同じ名前を持つ別のクラスタ・リソースがすでに存在する場合
- 別のリスナー（Oracle Application Server Web リスナー、Oracle HTTP Web Server、Oracle TNS リスナーなど）で同一の IP アドレスとポートをリスニングしている場合

次の各項では、Oracle WebDB リスナーをグループに追加するためのデータ要件を詳細に説明します。

8.3.3.1 ノードの選択

Oracle WebDB リスナーをグループに追加する際に、クラスタが2つのノードで構成されており、そのうち1つのノードが使用できない場合、どのノードを Oracle WebDB リスナーの可能所有者とするのかを指定するように求められます。このような場合、[図 8-3](#) に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 8-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ

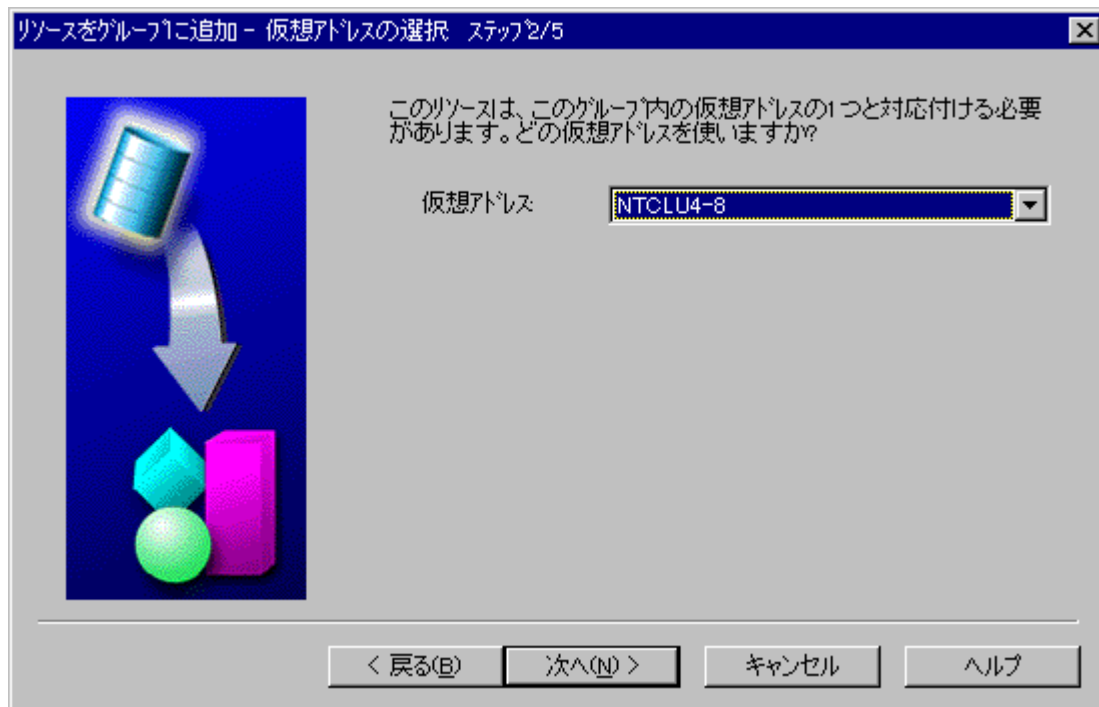


8.3.3.2 Oracle WebDB リスナーの仮想アドレス

Oracle WebDB リスナーをグループに追加するには、そのグループに少なくとも 1 つの仮想アドレスが含まれている必要があります。図 8-4 に示すように、Oracle WebDB リスナーを追加するグループ内に複数の仮想アドレスが含まれている場合、Oracle WebDB リスナーと関連付ける仮想アドレスを選択するように要求されます。

この仮想アドレス（および次のウィザード・ページで指定するポート番号）は、8.4 項で説明するように、アプリケーションで Oracle WebDB サイトへの接続のために指定するホストとなります。

図 8-4 Oracle WebDB リスナーの「仮想アドレスの選択」ウィザード・ページ



8.3.3.3 Oracle WebDB リスナーの識別情報

図 8-5 に示すように、「リソースをグループに追加」ウィザードでは、Oracle WebDB リスナーの識別情報に関する次の情報が要求されます。

- Oracle WebDB リスナー名
クラスタ上の Oracle WebDB リスナーを一意に識別する名前です。
- Oracle ホーム
Oracle Fail Safe で Oracle WebDB リスナーを作成する Oracle ホームです。
- Oracle WebDB リスナーを実行するポート
このポート番号は、Oracle WebDB リスナーと関連付けられたグループ IP アドレスと組み合わせると、Oracle WebDB リスナーでリスニングするアドレスとなります。IP アドレスとポート番号の組合せは、クラスタ内で一意になるようにします。

- Oracle Fail Safe で構成ファイルを作成する共有クラスタ・ディスク

Oracle WebDB リスナーをグループに追加すると、Oracle Fail Safe により構成ファイルが作成されます。これらのファイルが作成されるディスクは、フェイルオーバー後に Oracle WebDB リスナーを実行しているどのノードからもアクセスできるように、共有クラスタ・ディスクとする必要があります。

Oracle WebDB リスナー構成ファイルにディスクを指定すると、Oracle Fail Safe では、Oracle WebDB リスナーを追加するグループと同じグループにこのディスクを追加するので注意してください。他のグループのリソースによる依存がないディスクを選択してください。

Oracle Fail Safe では、次のファイル指定で作成するディレクトリ内に、Oracle WebDB リスナー構成ファイルを作成します。

```
<cluster-disk>:%webdb_<virtual_address_name>%<listener_name>
```

ここで、<cluster-disk> はこのウィザード・ページで指定するディスク、<virtual_address_name> は Oracle WebDB リスナーと関連付けられた仮想アドレス、<listener_name> はこのウィザード・ページで指定した Oracle WebDB リスナー名です。

Oracle WebDB 構成ファイルは、次のとおりです。

- wdbsvr.app

Oracle Fail Safe により、このファイルが作成されます。このファイルでは、このリスナーを介してアクセス可能な Oracle WebDB データベースが定義されます。次のウィザード・ページでデータベースを指定する場合、Oracle Fail Safe ではそのデータベースをファイルに指定します。リスナーの起動時に、wdbsvr.app 構成ファイルが読み込まれます。

このリスナーを介して複数のデータベースにアクセスするように構成できます。また、リスナーで接続するデフォルト・データベースを指定できます。wdbsvr.app ファイルを変更して、複数のデータベースあるいはデフォルト・データベースを指定するには、Oracle WebDB マニュアルを参照してください。

- wdbsvr.cfg

Oracle Fail Safe により、このファイルが作成されます。このファイルには、スレッドの最大数および仮想ディレクトリのマップ表など、リスナーのパラメータに関連する情報が含まれます。

- HTML ファイル

これらのファイルは、実行可能な Oracle WebDB リスナーのインストール時に作成されます。Oracle Fail Safe では、プライベート・ディスクのデフォルトの場所から、指定する共有クラスタ・ディスクへファイルをコピーします。HTML ファイルでは、リスナーのプロパティを変更する使いやすい Web インタフェースが提供されます。

図 8-5 「Oracle WebDB リスナーの識別情報」ウィザード・ページ

リソースをグループに追加 - Oracle WebDBリスナーの識別情報 ステップ2/4

Oracle Fail Safeで作成しグループに追加するOracle WebDBリスナーの名前は何か?

WebDBリスナー名

どのOracleホームからOracle Fail SafeでOracle WebDBリスナーを作成しますか?

Oracleホーム:

Oracle WebDBリスナー・プロセスが実行するポートはどれですか?

ポート番号:

Oracle Fail SafeでOracle WebDBリスナーの構成ファイルを書き込むディスクはどれですか?構成ファイルはクラスタ・ディスクにある必要があります。

クラスタ・ディスク:

< 戻る(B) 次へ(N) > キャンセル ヘルプ

8.3.3.4 Oracle WebDB リスナーのデータベース

図 8-6 に示すように、「リソースをグループに追加」ウィザードでは、Oracle WebDB リスナーがアクセスするデータベースに関する情報が要求され、選択したデータベースの TNSNAMES.ORA の更新に Oracle Fail Safe で使用する情報を表示します。このページでのデータ指定は、オプションです。

- Oracle WebDB リスナーが接続するデータベースはどれですか？

Oracle WebDB がまだインストールされていないデータベースを選択します。ただし、Oracle WebDB が当該データベースにインストールされる前に、ユーザーが Oracle WebDB リスナーを使用してデータベースへのアクセスを試行した場合は、Oracle WebDB リスナーによりエラーがレポートされます。

「なし」を選択した場合、WebDB リスナーの可用性を高めた後、データベース名を wdbsvr.app ファイル ([default_DAD] セクションの下) に追加する必要があります。この追加作業には、Oracle WebDB リスナーの「パラメータ」タブ・ページを使用します。

- データベース詳細

「データベース詳細」ボックスには、選択したデータベースの TNSNAMES.ORA ファイルに示されるエントリが表示されます。このデータは、(Oracle Fail Safe での Oracle WebDB リスナー作成時に) Oracle WebDB リスナーの可能所有者である他のクラスタ・ノード上の TNSNAMES.ORA ファイルにコピーされます。

Oracle Fail Safe では、実行されているノードにかかわらず Oracle WebDB リスナーで接続情報を使用できるように、TNSNAMES.ORA ファイルを Oracle WebDB リスナーの可能所有者であるすべてのクラスタ・ノードのデータベース情報で更新します。

この Oracle WebDB リスナーを使用して他のデータベースにアクセスする場合、「リソースをグループに追加」ウィザードの完了後、wdbsvr.app ファイルに他のデータベース・エントリを追加できます。詳細は、Oracle WebDB のマニュアルを参照してください。

図 8-6 「Oracle WebDB リスナーのデータベース」ウィザード・ページ



8.3.3.5 Oracle WebDB リスナーのアカウント

「Oracle WebDB リスナーのデータベース」ウィザード・ページでデータベースを選択した場合、次のウィザード・ページで、[図 8-7](#) に示すようにデータベース・ユーザー名、および Oracle WebDB リスナーがデータベースへのアクセスに使用するアカウントのパスワードを指定するように求められます。ユーザー名とパスワードは、両方またはどちらか一方を指定するか、あるいはいずれの値も指定しなくても構いません。

ユーザー名を指定しない場合、WebDB リスナーの可用性を高めた後、ユーザー名を wdbsrv.app ファイル ([default_DAD] セクションの下) に追加する必要があります。この追加作業には、Oracle WebDB リスナーの「パラメータ」タブ・ページを使用します。

パスワードを指定しない場合、Oracle WebDB リスナーの使用時にパスワードの指定を求められます。ユーザー名またはパスワードを指定した場合、暗号化されないまま wdbsvr.app ファイルに保存されます。

図 8-7 「Oracle WebDB リスナーのアカウント」ウィザード・ページ

リソースをグループに追加 - Oracle WebDBリスナーのアカウント: ステップ 4/4

Oracle WebDBアカウント情報を入力できます(オプション)。WebDBリスナーはこの情報を利用してWebDBデータベースに接続します。

このアカウント情報により、WebDBリスナーはフェイルオーバー後WebDBデータベースに透過的に再接続できます。

ユーザー名:

パスワード:

パスワードの確認:

< 戻る(B) 完了(F) キャンセル ヘルプ

8.4 Oracle WebDB リスナーへのクライアント接続

クライアントは、次のように、Web ブラウザでリスナーの仮想アドレスおよびポート番号を URL として指定し、Oracle WebDB リスナー（およびリスナーでリスニングするデータベース）に接続します。

```
http://<virtual-address>:<port>
```

<virtual-address> は Oracle WebDB リスナーの仮想アドレス、<port> は Oracle WebDB リスナーのポート番号です。

たとえば、可用性の高い Oracle WebDB リスナーの仮想アドレスが ntclu4-6、ポート番号が 4846 の場合、Oracle WebDB リスナーに接続するには Web ブラウザで次のように指定します。

```
http://ntclu4-6:4846
```

8.5 Oracle WebDB リスナーまたは PL/SQL 設定の変更

可用性が高まるように構成されたリスナーの Oracle WebDB リスナーまたは PL/SQL 設定を変更するには、次の URL を使用します。

- Oracle WebDB リスナー設定の変更

```
http://<virtual-address>:<port>/admin_/listener.htm?schema=webdb
```

- PL/SQL ゲートウェイ設定の変更

```
http://<virtual-address>:<port>/admin_/gateway.htm?schema=webdb
```

いずれの場合も、<virtual-address> は Oracle WebDB リスナーの仮想アドレス、<port> は Oracle WebDB リスナーのポート番号です。

8.6 Oracle WebDB リスナーのセキュリティ要件

Oracle WebDB リスナーをクラスタ環境で実行する場合、特別な権限は不要です。Oracle のリソースおよびアプリケーションを管理し、Oracle Fail Safe Manager を介して操作を実行するための適切な権限については、[4.3 項](#)を参照してください。

8.7 Oracle WebDB のインストールに関するヒント

可用性が高まるように Oracle WebDB リスナーを構成するには、リスナーの可能所有者となるクラスタ・ノードすべてにリスナーをインストールする必要があります。（可能所有者ノードの説明は、[2.6.6 項](#)を参照してください。）Oracle WebDB リスナーで要求を受け取るデータベースは、フェイルセーフ Oracle WebDB データベース、あるいはスタンドアロン Oracle WebDB データベースのいずれでも構いません。次に、可能なインストール・タイプを説明します。

- Oracle WebDB データベースの常駐していないクラスタに Oracle WebDB リスナーをインストールするには、次のようにします。

Oracle WebDB リスナーの可能所有者となる各クラスタ・ノードのプライベート・ディスクで、それぞれ Oracle WebDB リスナーのカスタム・インストールを実行します。

- フェイルセーフ Oracle WebDB データベースの常駐するクラスタに Oracle WebDB リスナーをインストールするには、次のようにします。
 1. Oracle WebDB リスナーの可能所有者となる各クラスタ・ノードのプライベート・ディスクで、それぞれ Oracle WebDB リスナーのカスタム・インストールを実行します。
 2. Oracle WebDB データベースを含むグループに対して「グループの検証」コマンドを実行します。ウィザードで TNSNAMES.ORA ファイルを更新するかどうかを問われたら、「はい」を選択します。
 3. ノードの 1 つで、Oracle WebDB のカスタム・インストールを実行し、インストール手順においてデータベース情報を求められたら、フェイルセーフ・データベースを指定します。これにより、Oracle WebDB パッケージがフェイルセーフ・データベースヘインストールされます。このタイプの構成は、[図 8-1](#) に示されています。
- スタンドアロン Oracle WebDB データベースの常駐するクラスタに Oracle WebDB リスナーをインストールするには、次のようにします。
 1. Oracle WebDB リスナーの可能所有者となる各クラスタ・ノードのプライベート・ディスクで、それぞれ Oracle WebDB リスナーのカスタム・インストールを実行します。
 2. リスナーの Oracle WebDB ホームにある TNSNAMES.ORA を、スタンドアロン・データベース情報で手動により更新します。
 3. Oracle WebDB データベースの常駐するノードで、もう 1 つの Oracle WebDB のカスタム・インストールを実行し、製品リストのページで「Oracle WebDB」を選択して、Oracle WebDB パッケージをスタンドアロン・データベースにインストールします。インストール手順においてデータベース情報を求められたら、TNSNAMES.ORA の更新（手順 2）に使用した情報を指定します。

必要な場合、可用性が高まるようにスタンドアロン Oracle WebDB データベースを構成できます。詳細は、[第 7 章](#)を参照してください。

Oracle WebDB はいずれの Oracle ホームにもインストールでき、またシステムに複数のバージョンをインストールできます。Oracle WebDB リスナーをグループに追加する際、Oracle Fail Safe で使用する Oracle WebDB のインストールを選択します。

8.8 可用性の高い Oracle WebDB リスナーに関連する問題のトラブルシューティング

トラブルシューティング・ツール（「クラスタの検証」および「グループの検証」）に関する一般情報は、[第 6 章](#)に記載しています。Oracle WebDB リスナーのトラブルシューティングに関する一般的な情報は、Oracle WebDB のマニュアルを参照してください。

Oracle WebDB リスナーを含むグループに「グループの検証」コマンドを発行すると、Oracle Fail Safe では、クラスタ・レジストリにある情報を構成ファイル内に記述されている情報と比較します。情報が一致しない場合、Oracle Fail Safe により問題を修正するかどうかを尋ねられます。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合

可用性を高めるための Oracle Forms Load Balancer Server の構成

Oracle Fail Safe には、Oracle Forms Load Balancer Server をグループに追加するウィザードがあり、Oracle Forms Load Balancer Server を活用した Web ベースの Oracle Forms アプリケーションの可用性を向上させます。

Oracle Fail Safe には、可用性が高まるように Oracle Forms Server を構成するウィザードもあります。Oracle Forms Load Balancer Server を使用する場合、Oracle Forms Server よりも Oracle Forms Load Balancer Server の可用性が高まるように構成することをお勧めします。(第 10 章では、Oracle Forms Server を可用性が向上するように構成する方法を説明します。)

この章では、次の項目について説明します。

項目	参照
概要	9.1 項
スタンドアロン Oracle Forms Load Balancer Server の検出	9.2 項
Oracle Forms Load Balancer Server のグループへの追加	9.3 項
Oracle Forms Load Balancer Server のセキュリティ要件	9.4 項
起動パラメータの変更	9.5 項
可用性の高い Oracle Forms Load Balancer Server に関連する問題のトラブルシューティング	9.6 項

9.1 概要

Oracle Forms Load Balancer Server を使用すると、一連の中間層のコンピュータ（サーバー・ファームと呼ばれることもある）のメンテナンス、これらのシステム間のサーバー・トラフィックのロード・バランスを実行できます。ロード・バランスは、CGI サポートのある Web サーバーで実行できる CGI-bin 実行可能ファイルを使用して実装されます。

Oracle Forms Load Balancer Server では、様々なロード・バランス・セット内のすべての Oracle Forms Server の記録をとります。指定セットのサーバーの状態を記録し、その負荷を示す統計を収集します。各 Form 実行要求を、指定セット内で最も負荷の小さいコンピュータに送る役割を果たします。

各 Oracle Forms Load Balancer Client では、現在システムで実行中の Forms プロセス数などの負荷情報を Oracle Forms Load Balancer Server へ送信します。Oracle Forms Load Balancer Client は、各システム上で Oracle Forms Server とともに実行されます。

次の手順では、CGI-bin 実行可能ファイルでロード・バランスを使用する際に発生するイベントについて説明します。

1. Oracle Forms Load Balancer Client では、Oracle Forms Load Balancer Server へ定期的に負荷情報を送信します。この負荷情報には、各 Oracle Forms Load Balancer Client で実行中の合計プロセス数が含まれます。
2. ユーザーは、Forms CGI-bin 実行可能ファイルを指す URL にアクセスします。
3. Forms CGI-bin 実行可能ファイルでは、使用可能であり、負荷が最小のシステム名を Oracle Forms Load Balancer Server に問い合わせます。
4. Forms CGI-bin 実行可能ファイルでは、HTML ファイルを動的に作成し、負荷の最も小さいシステムを Forms Server を実行するシステムに指定して、ユーザーの Web ブラウザにこの HTML ファイルを返します。
5. ユーザーの Web ブラウザでは、HTML ファイルで指定されたホストから Java アプレットをダウンロードするように要求します。
6. Java アプレットでは、特定の Form Builder アプリケーション（.FMX ファイル）を求める要求を Forms Server に送信します。
7. サーバーでは、Forms Server ランタイム・エンジンと通信します。（サーバーは、アプリケーションの起動遅延が最小になるように、使用可能な一連のランタイム・エンジンをメンテナンスします。）アクティブなユーザーは、専用のランタイム・エンジンを受け取ります。
8. サーバーでは、ランタイム・エンジンと、ダイレクト・ソケット、HTTP または HTTPS との接続を確立し、ソケット、HTTP または HTTPS 情報を Java アプレットへ送信します。Java アプレットでは、ランタイム・エンジンと、ダイレクト・ソケット、HTTP または HTTPS との接続を確立します。これで、Java アプレットとランタイム・エンジンは直接通信するようになり、サーバーは他のユーザーからの起動要求を受け入れることが可能になります。（この時点では、アプリケーション・サーバーおよび Forms Server のいずれも、アプレットとランタイム・エンジン間の通信に携わっていません。）Java

アプレットでは、ユーザーの Web ブラウザのメイン・ウィンドウに、アプリケーションのユーザー・インタフェースを表示します。

9. ランタイム・エンジンは、データ・ソースに応じて、Net8 または ODBC (Open Database Connectivity) を介して直接データベースと通信します。
10. Oracle Forms Load Balancer Client では、Oracle Forms Load Balancer Server へ負荷情報の送信を続けます。新規サービス要求のルートは、この情報に基づいて決定されます。

このようにして、Oracle Forms および Oracle Forms Load Balancer Server では、Oracle Fail Safe を使用せずに可用性の高いフォームを提供します。ただし、Oracle Forms Load Balancer Server で障害が発生する場合があります。Oracle Forms Load Balancer Server が使用できなくなった場合、(手順 3 にある) Oracle Forms CGI-bin 実行可能ファイルでは最小負荷のシステムに関する情報を受け取ることができません。かわりに、Forms CGI-bin では、ユーザーのブラウザを Oracle Forms Load Balancer Server の MetricsServerErrorURL パラメータで指定した URL にリダイレクトします。

Oracle Fail Safe を使用して可用性の高い Oracle Forms Load Balancer Server を構成することにより、構成内の障害箇所をなくすることができます。

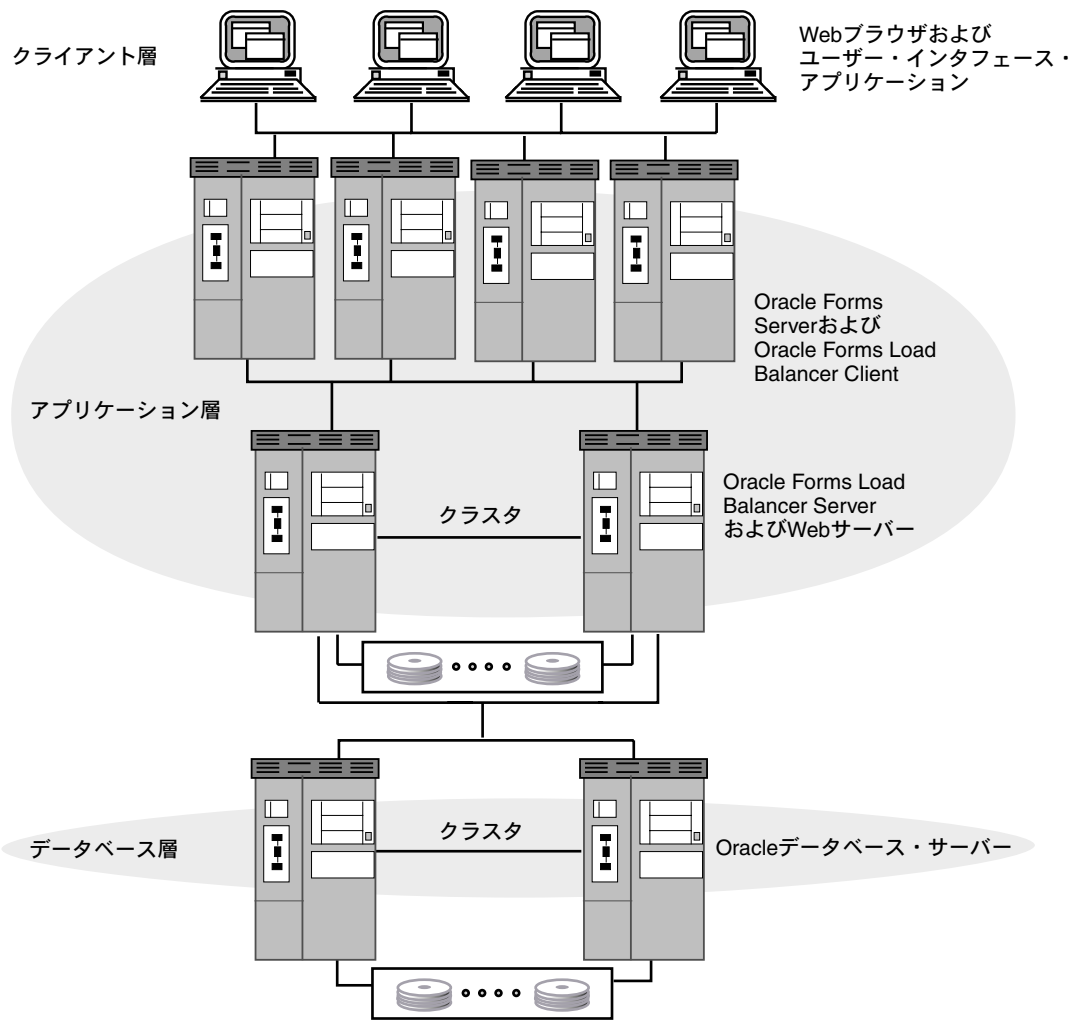
図 9-1 に、Oracle Forms Load Balancer Server と Oracle Web サーバーがデータベースとは別のクラスタに常駐する階層化実装を示します。Oracle Forms Server および Oracle Forms Load Balancer Client は、いくつかのクラスタ化されていないシステム上に常駐しています。

図 9-1 の 3 つの層は次のとおりです。

- クライアント層： Web ブラウザ、ユーザー・インタフェース・アプリケーション。
- アプリケーション層： 2 つのノードで構成されるクラスタ上の Oracle Forms Load Balancer Server および Web サーバー。クラスタ化されていない一連のシステム上の Oracle Forms Server および Oracle Forms Load Balancer Client。
- データベース層： 2 つのノードで構成されるクラスタ上の Oracle データベース・サーバー。

階層化実装では、インターネット・ビジネス・ソリューションに対して高可用性を増分的に展開できます。たとえば、Oracle Fail Safe を使用して、Oracle Forms Load Balancer Server またはデータベースのコンポーネントで障害の発生する可能性のある箇所を削除したり、この 2 つの可用性が高まるようにクラスタ上に構成することができます。

図 9-1 Oracle Forms Load Balancer を使用した Oracle Forms の 3 層アーキテクチャ



9.2 スタンドアロン Oracle Forms Load Balancer Server の検出

Oracle Fail Safe Manager を使用して、可用性が向上するように Oracle Forms Load Balancer Server を構成します。Oracle Fail Safe Manager インタフェースでスタンドアロン Oracle Forms Load Balancer Server を表示するには、まずスタンドアロン Oracle Forms Load Balancer Server を検出する必要があります。Oracle Fail Safe Manager では、Oracle Forms Load Balancer Server を検出する際、Windows サービス マネージャ内のリストを調べて Oracle Forms Load Balancer Server エントリを見つけ、可用性が高まるように構成されているかどうかを判断します。Oracle Fail Safe では、新しく検出されたサービスが Oracle Fail Safe Manager のツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

9.3 Oracle Forms Load Balancer Server のグループへの追加

可用性が向上するように Oracle Forms Load Balancer Server を構成するには、Oracle Forms Load Balancer Server をグループへ追加します。Oracle Fail Safe には、既存の Oracle Forms Load Balancer Server のグループへの追加や、Oracle Forms Load Balancer Server の新規作成とグループへの追加が可能なウィザードがあります。Oracle Forms Load Balancer Server をグループに追加すると、Oracle Fail Safe では Oracle Forms Load Balancer Server が依存する他のすべてのリソースを追加します。通常、このグループには次のリソースが含まれます。

- Oracle Forms Load Balancer Server
- 1 つ以上の仮想アドレス

Web サーバー（Oracle Application Server または Oracle HTTP Server）および Forms Server でアクセスするデータベースをグループに追加することにより、Web アプリケーション全体の可用性を向上させることができます。Oracle Forms Load Balancer Server、Web サーバーおよびデータベースを追加するグループは、異なるグループでも構いません。

注意： Oracle Forms Load Balancer Server 自体はフェイルオーバー後にリカバリされますが、フェイルオーバー前にデータベースにコミットされなかったフォーム入力データはすべて失われます。フェイルオーバー中に失われたコミットの完了していないフォーム入力データは、ユーザーまたはアプリケーションによる再入力が必要です。

9.3.1 構成前に

Oracle Forms Load Balancer Server をグループに追加する前に、次のことに注意してください。

- Oracle Forms Load Balancer Server 実行可能ファイルは、Oracle Forms Load Balancer Server の可能所有者である各クラスタ・ノードのプライベート・ディスク上にある Oracle ホームにインストールします。
- その他の必要な Oracle 製品実行可能ファイルは、Oracle Forms Load Balancer Server の可能所有者である各クラスタ・ノード上の Oracle ホーム（複数でも可）にインストールします。
- グループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

9.3.2 構成手順

Oracle Fail Safe により、Oracle Forms Load Balancer Server の可用性が高まるように必要な手順が自動化されます。Oracle Fail Safe では、既存の Oracle Forms Load Balancer Server のグループへの追加、Oracle Forms Load Balancer Server の新規作成とグループへの追加が可能です。

表 9-1 に、可用性が高まるように Oracle Forms Load Balancer Server を構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択します。

表 9-1 Oracle Forms Load Balancer Server の構成手順

手順	処置	説明
1	Oracle Forms Load Balancer Server の可能所有者となるクラスタの各ノードのプライベート・ディスク上に、Oracle Forms Load Balancer Server ソフトウェアがインストールされていることを確認	インストールの詳細情報は、Oracle Forms のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に Oracle Forms Load Balancer Server コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Load Balancer Server コンポーネントを選択します。これにより、Oracle Forms Load Balancer Server の DLL ファイルと Windows レジストリ変数がインストールされます。詳しい指示は、『Oracle Fail Safe インストール・ガイド』を参照してください。
3	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「<Oracle_Home>」→「Oracle Fail Safe Manager」

表 9-1 Oracle Forms Load Balancer Server の構成手順（続き）

手順	処置	説明
4	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
5	グループの作成および1つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）
6	Oracle Forms Load Balancer Server のグループへの追加	「リソース」→「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードを使用すると、Oracle Forms Load Balancer Server の選択 ¹ 、データ・ポートおよびリクエスト・ポートの指定、Oracle Forms Load Balancer Client の最大数の指定が可能です。
7	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
8	Web サーバーをフェイルオーバー用に構成	Web サーバーを新規に作成して追加し、Oracle Forms アプリケーションを Web 上で使用できるようにします。Web サーバーを作成してグループに追加する方法の詳細は、第 12 章を参照してください。

¹ 既存の Oracle Forms Load Balancer Server をグループに追加する場合は、「リソースをグループに追加」ウィザードで（「Load Balancer Server 名」ボックスから）その名前を選択できます。Oracle Forms Load Balancer Server を新規に作成して追加する場合は、クラスタ内で一意となるような名前を指定してください。

9.3.3 Oracle Forms Load Balancer Server の構成データ

Oracle Fail Safe Manager には、可用性が向上するように Oracle Forms Load Balancer Server を構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- Oracle Forms Load Balancer Server と関連付ける仮想アドレス
- Oracle Forms Load Balancer Server のクラスタ内で一意の名前
- Oracle Forms Load Balancer Server が Oracle Forms Load Balancer Client からのロード・データをリスニングするデータ・ポート

- Oracle Forms Load Balancer Server が Forms Web CGI による最小負荷のホストへの要求をリスニングするリクエスト・ポート
- Oracle Forms Load Balancer Server にアクセスする Oracle Forms Load Balancer Client の最大数

Oracle Forms Load Balancer Server をグループに追加するクラスタワイド操作は、同じ名前のクラスタ・リソースが他に存在する場合に失敗します。

次の各項では、Oracle Forms Load Balancer Server をグループに追加するためのデータ要件を詳細に説明します。

9.3.3.1 ノードの選択

Oracle Forms Load Balancer Server をグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、[図 9-2](#) に示すように、選択済ノードのリストを指定して、Oracle Forms Load Balancer Server の可能所有者となるノードを指定するように求められます。特定のノードを Oracle Forms Load Balancer Server の可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

[2.6.6 項](#)では、可能所有者の概念について詳細に説明します。

図 9-2 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle Forms Load Balancer Server をグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを Oracle Forms Load Balancer Server の可能所有者とするのかを指定するように求められます。このような場合、[図 9-3](#) に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 9-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



9.3.3.2 Oracle Forms Load Balancer Server リスナーの仮想アドレス

Oracle Forms Load Balancer Server を追加するグループには、少なくとも1つの仮想アドレスが含まれている必要があります。図 9-4 に示すように、Oracle Forms Load Balancer Server を追加するグループ内に複数の仮想アドレスが含まれている場合、Oracle Forms Load Balancer Server と関連付ける仮想アドレスを選択するように要求されます。

図 9-4 「仮想アドレスの選択」ウィザード・ページ



9.3.3.3 Oracle Forms Load Balancer Server の識別情報

図 9-5 に示すように、「リソースをグループに追加」ウィザードでは、Oracle Forms Load Balancer Server の識別情報に関する次の情報が要求されます。

- Oracle Forms Load Balancer Server 名

既存の Load Balancer Server をグループに追加するには、既存の Oracle Forms Load Balancer Server を選択します。Oracle Fail Safe に新規 Oracle Forms Load Balancer Server を作成するように指示するには、クラスタ内で一意の名前を入力します。

- データ・ポート

これは、Oracle Forms Load Balancer Server が Oracle Forms Load Balancer Client からのロード・データをリスニングする TCP/IP ポート番号です。Oracle Forms Load Balancer Server のデータ・ポート値は、Oracle Forms Load Balancer Client すべてに対して指定したデータ・ポート値と同じである必要があります。

Oracle Forms Load Balancer Server を新規作成する場合、デフォルト値は 9010 です。既存の Oracle Forms Load Balancer Server のデフォルト値は、Oracle Forms Load Balancer Server ソフトウェアのインストール時に指定された値です。デフォルト値以外の値は、デフォルト値（9010）が他のプログラムですでに使用されている場合を除いて指定しないことをお勧めします。

- リクエスト・ポート

これは、Oracle Forms Load Balancer Server が Forms Web CGI による最小負荷のホストへの要求をリスニングする TCP/IP ポート番号です。この値は、formsweb.cfg ファイルに MetricServerPort パラメータとして記述されます。このファイルの詳細は、Oracle Forms のマニュアルを参照してください。

Oracle Forms Load Balancer Server を新規作成する場合、デフォルト値は 9020 です。既存の Oracle Forms Load Balancer Server のデフォルト値は、Oracle Forms Load Balancer Server ソフトウェアのインストール時に指定された値です。デフォルト値以外の値は、デフォルト値（9020）が他のプログラムですでに使用されている場合を除いて指定しないことをお勧めします。

- Oracle Forms Load Balancer Server にアクセスする Oracle Forms Load Balancer Client の最大数

必須の起動パラメータ変更の詳細は、[9.5 項](#)を参照してください。

ウィザードでは、Oracle Forms Load Balancer Server のトレース・レベルに設定する値を要求しません。デフォルトでは、Oracle Fail Safe は Concurrent Manager を新規作成する際、Oracle Forms Load Balancer Server のトレース・レベルをゼロ（トレースなし）に設定します。既存の Oracle Forms Load Balancer Server をグループに追加すると、Oracle Fail Safe ではトレース・レベルに Oracle Forms Load Balancer Server の現在の設定を使用します。トレース・レベルにより、Oracle Forms Load Balancer Server トレース・ファイルに記述する情報量が決定します。トレース・レベルを変更するには、Oracle Fail Safe のツリー・ビューから「Oracle Forms Load Balancer Server」を選択し、「パラメータ」タブをクリックして必要な変更を行います。

図 9-5 「Load Balancer Server の識別情報」ウィザード・ページ

リソースをグループに追加 - Load Balancer Serverの識別情報 ステップ3/3

グループ名 My_group

このグループに追加したいLoad Balancer Serverはどれですか?
既存のLoad Balancer Serverの名前から選択できます。または、
一意な名前を入力することによりOracle Fail Safeで新しいLoad
Balancer Serverを作成できます。

Load Balancer Server名 Test_LBS

データ・ポートは何ですか?
データ・ポート: 9010

リクエスト・ポートは何ですか?
リクエスト・ポート: 9020

最大クライアント数は何ですか?
最大: 1000

< 戻る(B) 完了(F) キャンセル ヘルプ

9.4 Oracle Forms Load Balancer Server のセキュリティ要件

Oracle Forms Load Balancer Server をクラスタ環境で実行する場合、特別な権限は不要です。Oracle のリソースおよびアプリケーションを管理し、Oracle Fail Safe Manager を介して操作を実行するための適切な権限については、[4.3 項](#)を参照してください。

9.5 起動パラメータの変更

次の各項では、次のようなことを可能にするために行う起動パラメータの変更について説明します。

- Oracle Forms Server CGI-bin 実行可能ファイルから、Oracle Forms Load Balancer Server へのアクセスを可能にすること
- Oracle Forms Load Balancer Client で、Oracle Forms Load Balancer Server へ負荷に関する情報を送信できるようにすること

- Oracle Forms Load Balancer Server で、各 Oracle Forms Server に作業負荷を割り当てられるようにすること

さらに、9.5.4 項では、可用性を向上させた Oracle Forms Load Balancer Server の起動パラメータを変更する方法を説明します。

9.5.1 Forms CGI-bin 実行可能ファイルの起動パラメータの変更

Oracle Forms Server CGI-bin 実行可能ファイルから可用性の高い Oracle Forms Load Balancer Server へアクセスできるようにするには、

<Oracle_Home>%FORMS60%Server%formsweb.cfg ファイルを次のように変更します。

- Oracle Forms Load Balancer Server の「リクエスト・ポート」起動パラメータと一致するように、MetricsServerPort パラメータを調整します。
- ServerHost パラメータを「%LeastLoadedHost%」に設定します。名前解決にドメイン名が必要な場合、ドメイン名を ServerHost パラメータに追加して「%LeastLoadedHost%.oracle.com」などのようにします。

9.5.2 Oracle Forms Load Balancer Client の起動パラメータの変更

Oracle Forms Load Balancer Client から可用性の高い Oracle Forms Load Balancer Server へ負荷に関する情報を送信できるようにするには、次に説明する起動パラメータを設定あるいは変更する必要があります。(Oracle Forms Load Balancer Client をインストールする際、インストール手順で、Oracle Forms Load Balancer Client で使用する次の起動パラメータを指定するように求められます。)

- データ・ホスト

デフォルト値は *localhost* です。クラスタ化されていない環境では、Oracle Forms Load Balancer Server を実行しているシステムの完全なホスト名です。ただし、クラスタ内に構成された Oracle Forms Load Balancer Server の場合、物理ノードを指定しないでください。かわりに、Oracle Forms Load Balancer Server と関連付ける仮想サーバー・アドレスを指定してください。

- データ・ポート

これは、Oracle Forms Load Balancer Server に関連付けられた各 Oracle Forms Load Balancer Client の Oracle Forms Load Balancer Server に指定した値と同じ値にする必要があります。

9.5.3 Oracle Forms Server の起動パラメータの変更

Oracle Forms Load Balancer Server で、各 Oracle Forms Server に作業負荷を割り当てられるようにするには、Oracle Forms Server のポート起動パラメータと一致するように、

<Oracle_Home>%FORMS60%Server%formsweb.cfg ファイルの ServerPort パラメータを調整します。

9.5.4 Oracle Forms Load Balancer Server の起動パラメータの変更

可用性が高まるように構成された Oracle Forms Load Balancer Server のデータ・ポート、リクエスト・ポート、クライアントの最大数およびトレース・レベルを変更するには、次のようになります。

1. Oracle Fail Safe Manager のツリー・ビューで、パラメータを変更する Oracle Forms Load Balancer Server を選択します。
2. 「パラメータ」タブをクリックします。
3. パラメータを調整します。
4. 「適用」をクリックします。

9.6 可用性の高い Oracle Forms Load Balancer Server に関連する問題のトラブルシューティング

Fail Safe リソースのトラブルシューティングに関する一般的な情報は、[第 6 章](#)に記載しています。Oracle Forms Load Balancer Server のトラブルシューティングに関する一般的な情報は、Oracle Forms のマニュアルを参照してください。

多くの場合、問題のトラブルシューティングでは、まず「クラスタの検証」あるいは「グループの検証」コマンドを発行します。[第 6 章](#)に、これらのツールの一般的な説明を記載しています。

Oracle Forms Load Balancer Server を含むグループに「グループの検証」コマンドを発行すると、Oracle Fail Safe では、クラスタ・レジストリにある情報を構成ファイル内に記述されている情報と比較します。情報が一致しない場合、Oracle Fail Safe により問題を修正するかどうかを尋ねられます。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合

可用性を高めるための Oracle Forms Server の構成

Oracle Fail Safe により、Web ベースの Oracle Forms アプリケーションの可用性が向上します。(Oracle Forms アプリケーションを Web 上で使用可能にするには、Web サーバーを作成して、Oracle Forms Server が含まれたグループに追加する必要があります。)

Oracle Fail Safe には、可用性が向上するように Oracle Forms Load Balancer Server を構成するウィザードがあり、Oracle Forms アプリケーションの可用性が高めることもできます。一般に、Oracle Forms Server よりも Oracle Forms Load Balancer Server の可用性が高まるように構成すると効果的です。可用性が向上するように Oracle Forms Load Balancer Server を構成する方法の詳細は、[第 9 章](#)を参照してください。

この章では、次の項目について説明します。

項目	参照
概要	10.1 項
スタンドアロン Oracle Forms Server の検出	10.2 項
Oracle Forms Server のグループへの追加	10.3 項
Oracle Forms Server のセキュリティ要件	10.4 項
Oracle Forms Server へのクライアント接続	10.5 項
可用性の高い Oracle Forms Server に関連する問題のトラブルシューティング	10.6 項

10.1 概要

Oracle Fail Safe は、Web ベースの実装で新規および既存の Oracle Forms Server を構成する際に役立ちます。Oracle Fail Safe により、コンテンツを提供する Web サーバー、アプリケーション・ロジックを提供する Oracle Forms Server、およびデータを格納して提供する Oracle データベースを含めた、インターネット・ソリューション全体の可用性を高めることができます。

Web ベースの実装では、アプリケーション・コンポーネントとデータベース・コンポーネントの両方をクラスター・ノードにインストールできるため、インターネット・アプリケーションのあらゆるレベルで可用性の高い構成にすることができます。階層化実装は、インターネット用に Oracle Forms アプリケーションを構成する一般的な方法で、単一のバックエンド・データベースにより、複数のシステムで多数の Oracle Forms Server が実行されます。

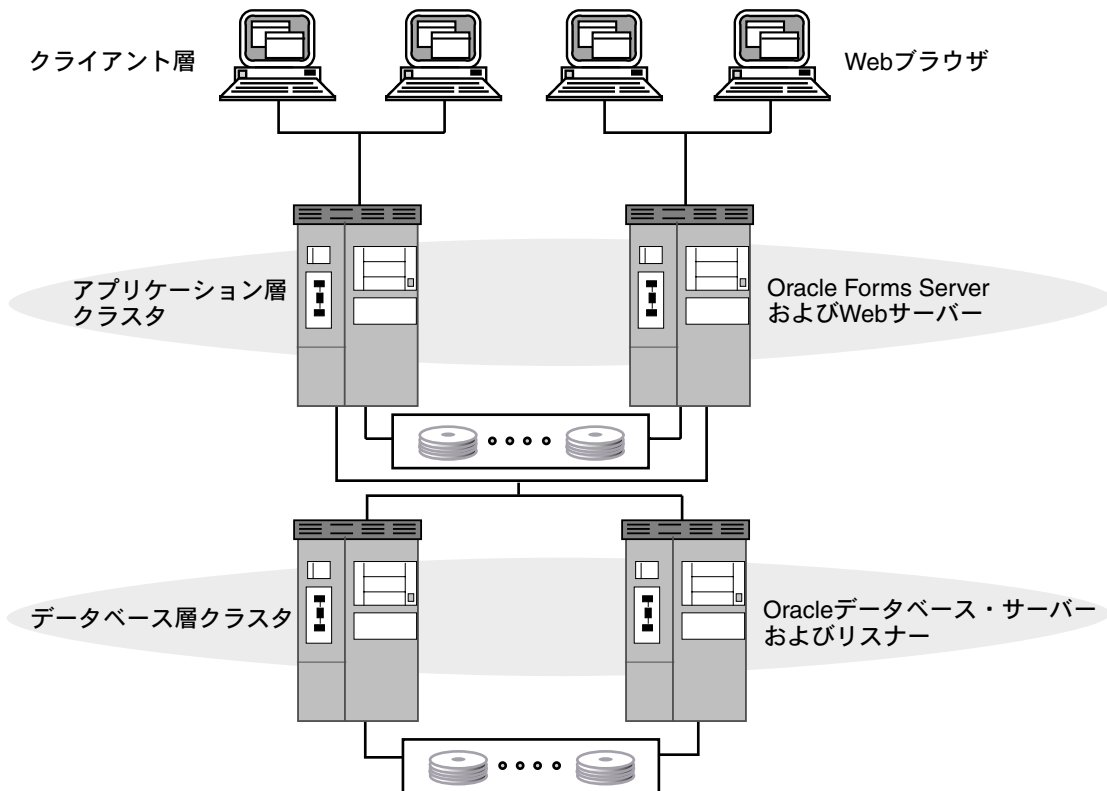
図 10-1 に、Oracle Forms Server と Oracle Web サーバーがデータベースとは別のクラスターに常駐する階層化実装を示します。

図 10-1 の 3 つの層は次のとおりです。

- クライアント層： Web ブラウザ、ユーザー・インタフェース・アプリケーション
- アプリケーション層： Oracle Forms Server、Web サーバー
- データベース層： Oracle データベース・サーバー、リスナー

階層化実装では、インターネット・ビジネス・ソリューションに対して高可用性を増分的に展開できます。たとえば、Oracle Fail Safe を使用して、アプリケーションまたはデータベースのコンポーネントで障害の発生する可能性のある箇所を削除したり、この 2 つの層の可用性が高まるようにクラスター上に構成することができます。

図 10-1 Oracle Forms の 3 層アーキテクチャ



10.2 スタンドアロン Oracle Forms Server の検出

Oracle Fail Safe Manager を使用して、可用性が向上するように Oracle Forms Server を構成します。Oracle Fail Safe Manager インタフェースでスタンドアロン Oracle Forms Server を表示するには、まずスタンドアロン Oracle Forms Server を検出する必要があります。

Oracle Fail Safe Server では、Oracle Forms Server を検出する際、Windows サービス マネージャ内のリストを調べて Oracle Forms Server エントリを見つけ、可用性が高まるように構成されているかどうかを判断します。Oracle Fail Safe では、新しく検出されたサービスが Oracle Fail Safe Manager のツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

10.3 Oracle Forms Server のグループへの追加

可用性が高まるように Oracle Forms Server を構成するには、少なくとも 1 つの仮想アドレスが含まれたグループに Oracle Forms Server を追加します。Oracle Fail Safe により、Oracle Forms Server が依存する他のすべてのリソースが追加されます。通常、このグループには次のリソースが含まれます。

- Oracle Forms Server
- Oracle Forms Server ファイル (.html、.fmx、.mmx および .pll) が配置される 1 つ以上のクラスタ・ディスク
- 1 つ以上の仮想アドレス

既存の Oracle Forms アプリケーションは、変更することなく、可用性が向上するように構成した Oracle Forms Server とともに使用できます。

ユーザーが Oracle Forms Server により提供されるフォームにアクセスする場合は、Oracle Forms Server が含まれたグループに Web サーバーを追加する必要があります。Web サーバーは仮想アドレスに依存します。Web サーバーの仮想アドレスは、ユーザーが Web 上の Forms アプリケーションにアクセスする際に入力する URL の一部になります。

どのクラスタ・ノードが Oracle Forms Server のホストであるかにかかわらず、フェイルオーバーの前後ともに同じ URL で Web ベースのフォームにアクセスできるように、ユーザーは仮想アドレスを使用して URL を指定します。障害が発生し、Oracle Forms Server と Web サーバーが別のノードにフェイルオーバーした後、ユーザーは Web ブラウザに URL を再ロードしてフォームに再接続できます。

注意： Oracle Forms Server 自体はフェイルオーバー後にリカバリされますが、フェイルオーバー前にデータベースにコミットされなかったフォーム入力データはすべて失われます。フェイルオーバー中に失われたコミットの完了していないフォーム入力データは、ユーザーまたはアプリケーションによる再入力が必要です。

10.3.1 構成前に

Oracle Forms Server をグループに追加する前に、次のことに注意してください。

- Oracle Forms 実行可能ファイルは、Oracle Forms Server の可能所有者である各クラスタ・ノードのプライベート・ディスク上にある Oracle ホームにインストールします。
- その他の必要な Oracle 製品実行可能ファイルは、Oracle Forms Server の可能所有者である各クラスタ・ノード上の Oracle ホーム（複数でも可）にインストールします。
- Oracle Forms Server データ・ファイルは、Oracle Forms Server の可能所有者であるクラスタ・ノードによるフェイルオーバー発生後にもアクセスできるように、すべてクラスタ・ディスク上に配置します。

- グループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

Oracle Forms Server の可用性を高めると、ユーザーには、アプリケーション全体の可用性が向上したような印象を与えます。さらに可用性を高めるには、フォームでアクセスする Oracle データベース・サーバーを、Oracle Forms Server のホストになっているクラスタ、それとは別のクラスタのどちらにおいても可用性が向上するように構成します。

10.3.2 構成手順

Oracle Fail Safe により、Oracle Forms Server の可用性が高まるように必要な手順が自動化されます。Oracle Fail Safe では、既存の Oracle Forms Server のグループへの追加、Oracle Forms Server の新規作成とグループへの追加が可能です。

表 10-1 に、可用性が高まるように Oracle Forms Server を構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、Oracle Fail Safe のチュートリアルとオンライン・ヘルプを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択するか、「ヘルプ」→「チュートリアル」を選択します。

表 10-1 Oracle Forms Server の構成手順

手順	処置	説明
1	Oracle Forms Server の可能所有者となるクラスタの各ノードのプライベート・ディスク上に、Oracle Forms ソフトウェアがインストールされていることを確認	インストールの詳細情報は、Oracle Forms のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に Oracle Forms Server コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle Forms コンポーネントを選択します。これにより、Oracle Forms Server の DLL ファイルと Windows レジストリ変数がインストールされます。詳しい指示は、『Oracle Fail Safe インストール・ガイド』を参照してください。
3	クラスタ・ディスクへの Oracle Forms アプリケーション・ファイルのコピー	ファイルおよび実行可能ファイル (.html、.fmx、.pll、.mmx ファイルなど) は、共有記憶装置インターコネクト上のクラスタ・ディスクに配置する必要があります。
4	スタンドアロン・ノード上での Oracle Forms アプリケーションのテスト	Oracle Forms Server をグループに追加する前に、単一ノード上の Oracle Forms アプリケーションをスタンドアロン・モードでテストします。たとえば、Forms ランタイム・コンポーネントと統合ソース・レベル・デバッグを使用して、このテストを実施します。
5	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「<Oracle_Home>」→「Oracle Fail Safe Manager」
6	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。

表 10-1 Oracle Forms Server の構成手順（続き）

手順	処置	説明
7	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）
8	Oracle Forms Server の グループへの追加	Oracle Fail Safe Manager で、「リソース」→「グループに追加」を選択し、「リソースをグループに追加」ウィザードを実行します。このウィザードを使用すると、Oracle Forms Server の選択 ¹ 、可能所有者ノードの指定、Forms ファイルの場所の指定、および Forms アプリケーションで使用されるデータベースの選択が可能です。
9	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
10	Web サーバーをフェイル オーバー用に構成	Web サーバーを新規に作成して追加し、Oracle Forms アプリケーションを Web 上で使用できるようにします。Web サーバーを作成してグループに追加する方法の詳細は、第 12 章を参照してください。

¹ 既存の Oracle Forms Server をグループに追加する場合は、「リソースをグループに追加」ウィザードで（「Forms Server 名」ボックスから）その名前を選択できます。Oracle Forms Server を新規に作成して追加する場合は、使用する名前がクラスター全体で一意になるようにしてください。

10.3.3 Oracle Forms Server の構成データ

可用性が向上するように Oracle Forms Server を構成するには、次の情報が必要です。

- Oracle Forms Server の可能所有者ノード（クラスターが3つ以上のノードから構成される場合、あるいは2つのノードから構成されるクラスターで1つのノードが使用できない場合）
- Oracle Forms Server に関連付けられた仮想アドレス（グループに複数の仮想アドレスが含まれる場合）
- Forms Server の識別情報
- Oracle Forms アプリケーション・ファイルのパス
- Oracle Forms アプリケーションがアクセスする Oracle データベース（複数の場合もあり）

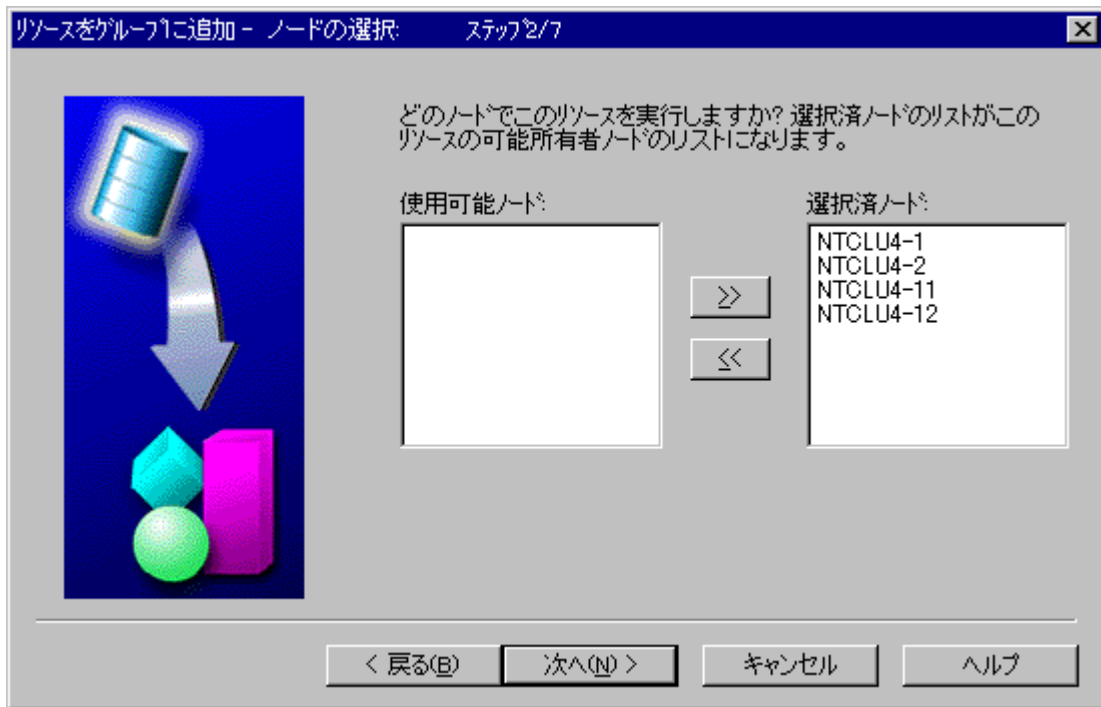
次の項では、構成情報の要件を詳細に説明します。

10.3.3.1 ノードの選択

Oracle Forms Server をグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、[図 10-2](#) に示すように、選択済ノードのリストを指定して、Oracle Forms Server の可能所有者となるノードを指定するように求められます。特定のノードをデータベースの可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

[2.6.6 項](#)では、可能所有者の概念について詳細に説明します。

図 10-2 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle Forms Server をグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを Oracle Forms Server の可能所有者とするのかを指定するように求められます。このような場合、[図 10-3](#) に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 10-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



10.3.3.2 Forms Server の仮想アドレス

Oracle Forms Server をグループに追加するには、そのグループに少なくとも 1 つの仮想アドレスが含まれている必要があります。図 10-4 に示すように、Oracle Forms Server を追加するグループ内に複数の仮想アドレスが含まれている場合、Oracle Forms Server と関連付ける仮想アドレスを選択するように要求されます。

仮想アドレスは、次の Windows レジストリ・キーに格納されています。

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\
OracleFormsServer-<ServiceName>\Host
```

図 10-4 Oracle Forms の「仮想アドレスの選択」ウィザード・ページ



10.3.3.3 Forms Server の識別情報

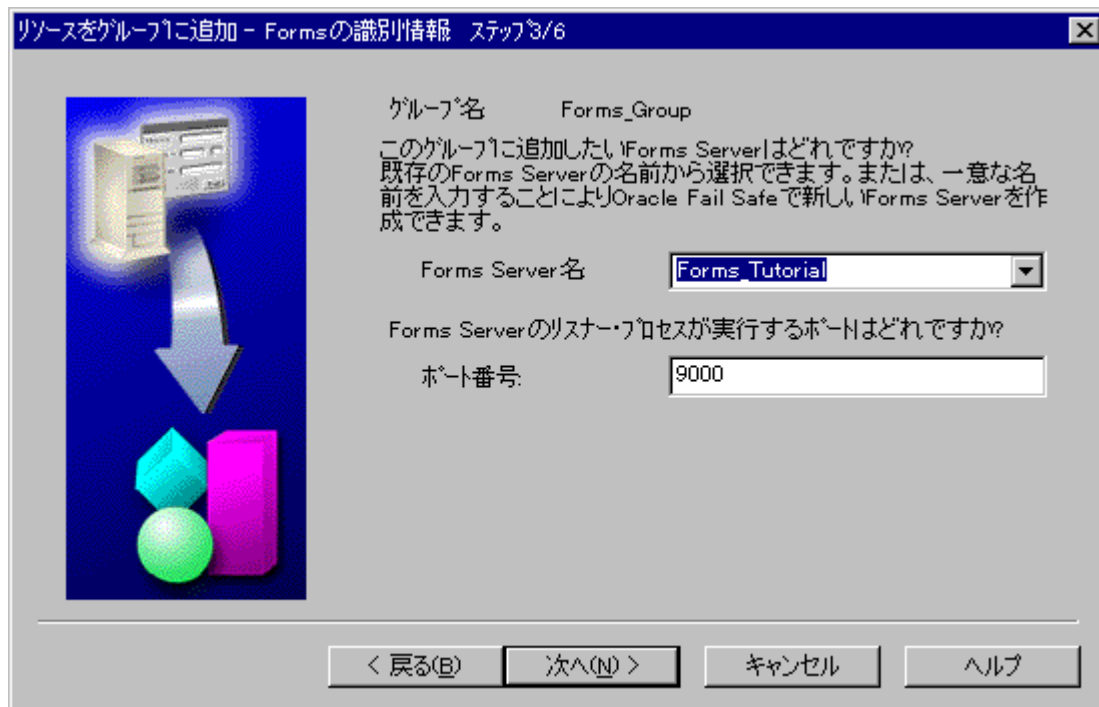
図 10-5 に示すように、Oracle Forms Server の識別情報は、Oracle Forms Server の名前と、Oracle Forms Server が実行されるポートで構成されます。Forms Server 名とポート番号は、クラスタ内で一意になるようにします。ここに指定するポート番号は、アプリケーションの HTML ファイルまたは URL に指定したポート番号と同一にする必要があります。

Forms Server は現在実行されているシステム上のすべての IP アドレスをリスニングするため、ポート番号は、他のアプリケーションで使用されていないものを指定してください。デフォルトでは、Oracle Application Server は、それ自身が稼働しているシステムの上すべての IP アドレスをリスニングするので、注意してください。12.7.2 項を参照してください。

ポート番号は、次の Windows レジストリ・キーに格納されています。

```
HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥
OracleFormsServer-<ServiceName>¥Port
```

図 10-5 「Forms の識別情報」ウィザード・ページ



リソースをグループに追加 - Formsの識別情報 ステップ3/6

グループ名 Forms_Group

このグループに追加したいForms Serverはどれですか?
既存のForms Serverの名前から選択できます。または、一意な名前を入力することによりOracle Fail Safeで新しいForms Serverを作成できます。

Forms Server名 Forms_Tutorial

Forms Serverのリスナー・プロセスが実行するポートはどれですか?

ポート番号 9000

< 戻る(B) 次へ(N) > キャンセル ヘルプ

10.3.3.4 Formsのパラメータ

図 10-6 に示すように、このウィザード・ページで、Oracle Forms リスナー・プロセスを実行する接続モード、および Oracle Forms Server で作成してプールに保持するワーカー・プロセス数を指定します。

接続モードは、次の設定のいずれかになります。

- Socket

これは、TCP/IP の標準プログラミング・インタフェースです。この方法は効率的であり、容易に使用できます。Forms Server は、ネットワーク化されたホスト・サーバー上で稼働し、ユーザー・システムで稼働中のクライアントから接続する際の指定ソケットまたはポートをリスニングします。この方法を有効にするには、クライアントおよびサーバーのシステムが、ネットワーク上で直接通信可能である必要があります。このモードでは、サーバー側プロキシを使用できません。サーバーとクライアントのシステムが、インターネットなど安全性の低いネットワークによって分断されている場合、Socket モードは、場合によって嚴重なセキュリティ上の意味を持ちます。ファイアウォールが存在する場合、Socket モードは機能しません。

- HTTP

この接続モードを使用すると、HTTP ソケット接続で通信が確立されます。Oracle Forms Server では、ソケットを使用した専用接続ではなく、クライアントからの HTTP 接続をリスニングします。Oracle Forms Server とクライアント間の内部メッセージ機能はすべて、HTTP パケット内にカプセル化されます。HTTP トラフィックのみを許可するサイトは、構成にほとんど変更を加えることなく、既存のファイアウォールを介して Oracle Forms アプリケーションを配置できます。

- HTTPS

この接続モードを使用すると、SSL (Secure Sockets Layer) を使用した HTTP ソケット接続で通信が確立されます。Oracle Forms Server では、SSL をトランスポート・プロトコルとして使用し、プライバシー、整合性およびサーバー認証を可能にします。SSL は、アプリケーション・レベルの 1 つ下のトランスポート・レベルで機能します。つまり SSL では、Telnet、FTP および HTTP などのアプリケーション・レベルのプロトコルで処理が行われる前に、メッセージを暗号化および復号化できます。

クライアントとサーバーの暗号化レベルに応じて、FORMS60_HTTP_NEGOTIATE_DOWN 環境変数を変更する必要があります。詳細は、Oracle Forms のマニュアルを参照してください。

さらに、HTTPS モードを使用する場合は、Oracle Wallet Manager をインストールし、認証要求の作成および認証管理を行う必要があります。詳細は、Oracle Forms のマニュアルを参照してください。

プール・パラメータでは、Oracle Forms Server で作成し、保持する必要のあるワーカー・プロセス (ifweb60.exe) の数を指定します。ワーカー・プロセスは、1 ユーザーに対して 1 つ割り当てられます。プール値を 4 に指定した場合、4 人のユーザーが遅延なく作業を開始できますが、5 人目のユーザーからは、ワーカー・プロセスを開始するたびに遅延が生じます。予想ユーザー数に基づいてワーカー・プロセス・プールのメンテナンスを行うと、アプリケーションの起動遅延を最小にすることができます。

接続モードおよびプール・パラメータは、それぞれ次の Windows レジストリ・キーに格納されています。

- HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥OracleFormsServer-<ServiceName>¥Mode
- HKEY_LOCAL_MACHINE¥SYSTEM¥CurrentControlSet¥Services¥OracleFormsServer-<ServiceName>¥Pool

図 10-6 「Forms のパラメータ」ウィザード・ページ

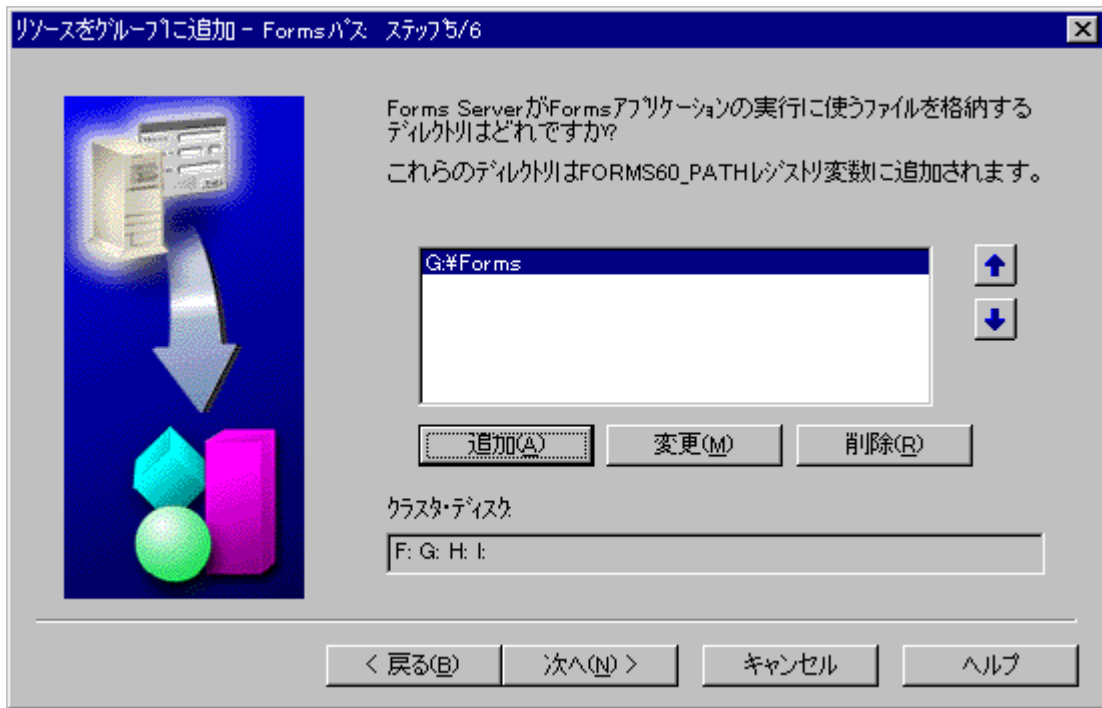
10.3.3.5 Oracle Forms アプリケーション・ファイルのパス

図 10-7 に示すように、このウィザード・ページでは、Oracle Forms アプリケーション・ファイル（.html、.fmx、.mmx および .pll）を配置するディスクおよびディレクトリを指定します。指定した内容は、FORMS60_PATH Windows レジストリ変数に追加されます。ディスクは、共有クラスタ・ディスクを指定してください。

ウィザードが完了すると、指定したディスクが、Oracle Forms Server を追加するグループに追加されます。共有クラスタ・ディスクを指定することで、Oracle Forms Server ファイルは、フェイルオーバー時に Oracle Forms Server を引き継ぐクラスタ・ノードからも使用可能になります。

このウィザード・ページでのディレクトリの表示順序は重要です。特定のファイルを検索する際、Oracle Forms Server では最初に見つかった該当ファイルを使用します。これにより、望ましくない結果になることもあります。たとえば、指定したファイルが最初の 2 つのディレクトリに存在しており、ユーザーが使用を考えているファイルが 2 番目のディレクトリにあるような場合です。

図 10-7 「Forms パス」 ウィザード・ページ



10.3.3.6 Oracle Forms アプリケーションによりアクセスされる Oracle データベース

図 10-8 に示すように、このページでは、Oracle Forms アプリケーションによりアクセスされる Oracle データベースを指定します。

Oracle Forms Server では、サービスを提供しているフォームを介したデータの表示および更新時にどのデータベース（複数可）を使用するのかを認識する必要があります。Oracle Fail Safe では、TNSNAMES.ORA ファイルを Oracle Forms Server の可能所有者であるすべてのクラスタ・ノードのデータベース情報で更新します。これにより、実行されているノードにかかわらず、Oracle Forms Server で接続情報が使用できます。

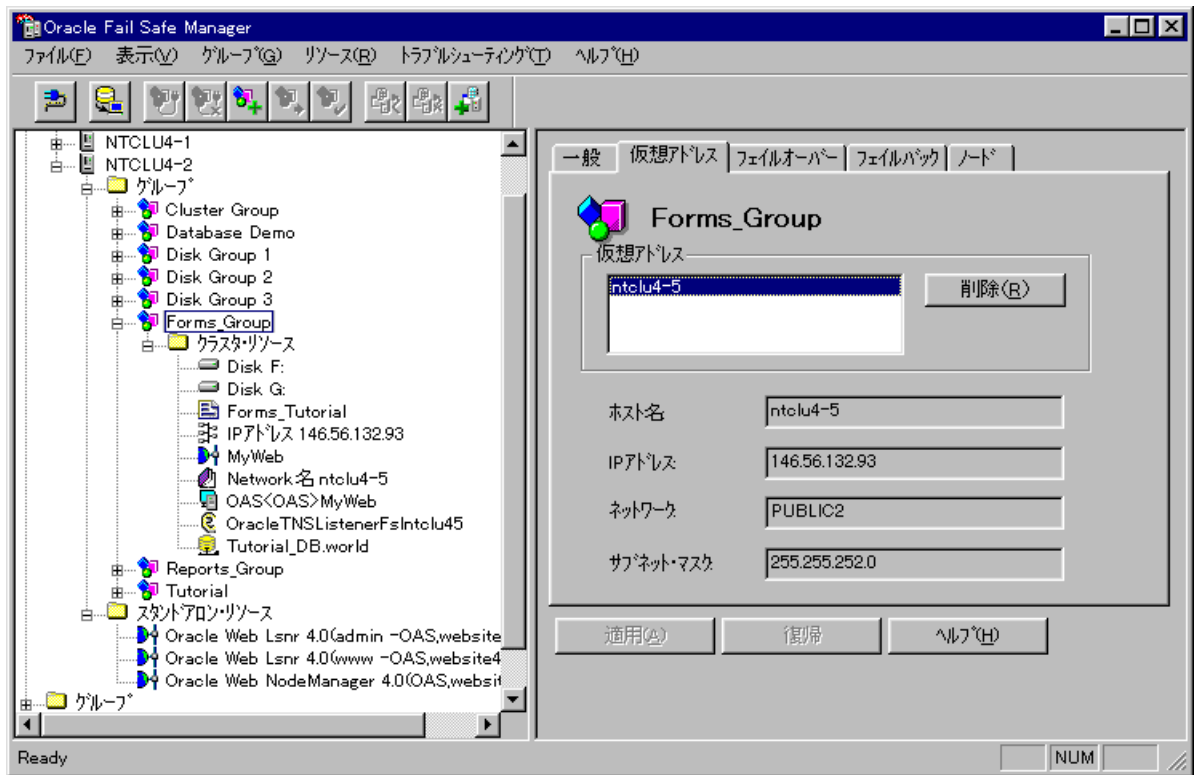
Oracle Fail Safe では、「選択済」ボックスにデータベースを追加しても、Oracle Forms Server を追加するグループにそのデータベースは追加されません。確実に可用性を高めるには、フェイルセーフの Oracle Forms Server によりアクセスされるデータベースをグループに追加する必要があります。このデータベースを Oracle Forms Server と同じグループに追加する必要はありません。

図 10-8 「Forms のデータベース」ウィザード・ページ



図 10-9 は Oracle Fail Safe Manager のツリー・ビューで、可用性が高まるように構成された Oracle Forms Server を示しています。この Oracle Forms アプリケーションでは、Tutorial_DB という Oracle データベースを使用します。このツリー・ビューには、Oracle Application Server Web リスナーも表示されています。Web でフォームを表示できるように、このリスナーは Forms_Group に追加されます。

図 10-9 Web ベースの Forms を示すツリー・ビュー



10.4 Oracle Forms Server のセキュリティ要件

Oracle Forms Server をクラスタ環境で実行する場合、特別な権限は不要です。Oracle のリソースおよびアプリケーションを管理し、Oracle Fail Safe Manager を介して操作を実行するための適切な権限については、[4.3 項](#)を参照してください。

10.5 Oracle Forms Server へのクライアント接続

ユーザーは、可用性を高めるように構成された Oracle Forms Server に直接接続しません。可用性が向上するように構成された Oracle Forms アプリケーションには、Web ブラウザを使用してアクセスする必要があります。

このため、ユーザーに完全な高可用性ソリューションを提供するには、Web リスナーの可用性も高める必要があります。Web リスナーは、仮想アドレスで動作するように構成します。ユーザーは、Web ブラウザで Web リスナーの仮想アドレスと HTML ファイルを指定してフォームにアクセスします。たとえば、Web サーバーの設定によっては、ユーザーは次のような URL を入力します。

```
http://ntclu4-5/forms.html:801
```

この URL では、ntclu4-5 が Web リスナーの仮想アドレス、forms.html が Oracle Forms アプリケーションの HTML ファイル、801 が Web リスナーが動作するポート番号です。

詳細は次を参照してください。

- 可用性が向上するように Web リスナーを構成する方法の詳細は、[第 12 章](#)を参照してください。
- フォームの可用性を高める Oracle Forms Server と Web サーバーの構成例は、Oracle Fail Safe のチュートリアルを参照してください。Oracle Fail Safe のチュートリアルにアクセスするには、Oracle Fail Safe Manager のメニュー・バーで「ヘルプ」→「チュートリアル」を選択してください。

10.6 可用性の高い Oracle Forms Server に関連する問題のトラブルシューティング

次の項では、可用性が向上するように構成した Oracle Forms Server を使用する際に発生する具体的な問題のトラブルシューティングについて説明します。Fail Safe リソースのトラブルシューティングに関する一般的な情報は、[第 6 章](#)に記載しています。Oracle Forms Server のトラブルシューティングに関する一般的な情報は、Oracle Forms のマニュアルを参照してください。

Oracle Forms Server を含むグループに「グループの検証」コマンドを発行すると、Oracle Fail Safe では、クラスタ・レジストリにある情報を構成ファイル内に記述されている情報と比較します。情報が一致しない場合、Oracle Fail Safe により問題を修正するかどうかを尋ねられます。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合

10.6.1 ユーザーが Oracle Forms Server にアクセスできない

ユーザーが Oracle Forms Server にアクセスできない場合は、次の処置を実行して問題を検出します。

- 「グループの検証」操作を実行して、Oracle Forms Server のリソース構成を検証します。
6.1.2 項の「グループの検証」操作の説明を参照してください。
- Forms Server リソースがオンラインになっていることを確認します。

Oracle Fail Safe Manager のツリー・ビューで、Oracle Forms Server リソースをクリックし、「一般」タブをクリックして、プロパティ・ボックスの「状態」フィールドの値を検証します。状態がオフラインの場合は、メニュー・バーの「リソース」→「オンラインに設定」を選択します。

- Oracle Forms Server リソースに指定されているポート番号が、Oracle Forms アプリケーションの HTML ファイルに指定されているポート番号と同じであることを確認します。

Oracle Fail Safe Manager のツリー・ビューで Oracle Forms Server リソースをクリックし、「パラメータ」タブをクリックしてポート番号を検証します。

10.6.2 Web ブラウザで Java アプレットをダウンロードできない

Web ブラウザで Java アプレットをダウンロードしようとしたときにエラーが発生した場合は、Web サーバー内の仮想ディレクトリ設定を調べ、Oracle Forms アプリケーションの HTML ファイルに指定されている仮想ディレクトリと比較します。たとえば、仮想ディレクトリが `WEB_CODE=<Oracle_Home>%forms60%java` と指定してある場合、Oracle Forms アプリケーションの静的 HTML ファイルには次の行が含まれている必要があります。

```
java_CODEBASE="/WEB_CODE/"
java_ARCHIVE="/WEB_CODE/F60all.jar"
```

10.6.3 Java アプレットを実行できない

Java アプレットが実行されておらず、クラス例外エラーを受け取った場合は、次の処置を実行します。

- 互換性のあるバージョンの JInitiator ソフトウェアを使用していることを確認します (『Oracle Fail Safe リリース・ノート』のソフトウェア互換表を参照)。たとえば、Netscape Navigator で JInitiator ソフトウェアのバージョンを表示するには、「ヘルプ」→「Plug-ins について」を選択します。

- Oracle Forms アプリケーションの HTML ファイルにある JInitiator タグを確認します。特に、参照されている JInitiator のバージョンを調べます。たとえば、次のようになります。

```
<PARAM NAME="type" VALUE="application/x-jinit-applet;version=1.1.7.15.1">
.
.
.
<EMBED type="application/s-jinit-applet;version=1.1.7.15.1">
```

10.6.4 Web ブラウザでフェイルオーバー後に Java アプレットを再ロードできない

フェイルオーバーの発生後、サーバーから Java アプレットを再ロードできない場合は、次の処置を実行します。

- Web ブラウザで、互換性のあるバージョンの JInitiator ソフトウェアを使用していることを確認します（『Oracle Fail Safe リリース・ノート』のソフトウェア互換表を参照）。たとえば、Netscape Navigator で JInitiator ソフトウェアのバージョンを表示するには、「ヘルプ」→「Plug-ins について」を選択します。
- Oracle Forms アプリケーションの HTML ファイルにある JInitiator タグを確認します。アプレット・キャッシュをオフにするタグは、次のように設定してください。

```
<PARAM NAME="jinit_appletcache" VALUE="off">
.
.
.
jinit_APPLETCACHE="off"
```

10.6.5 アプリケーションで Forms アプリケーションが見つからない

Oracle Forms アプリケーションが見つからないというエラー・メッセージを受け取った場合は、次の処置を試みます。

- HTML ファイルで次の module 構文を確認します。

```
PARAM NAME="serverArgs" VALUE="module=welcome.fmx userid=...">
.
.
.
serverArgs="module=welcome.fmx userid=..."
```

- 参照されているファイルが指定場所にあること、または FORMS60_PATH レジストリ変数に .fmx ファイルの場所が含まれていることを確認します。

10.6.6 Oracle Forms アプリケーションが正しく実行されない

Oracle Forms アプリケーションが Web ブラウザで正しく実行されない場合は、次の処置を試みます。

- フォームを Forms ランタイムでローカルに実行します。Forms ランタイムを起動するには、Windows のタスクバーで、次の順に選択します。
「スタート」→「プログラム」→「Oracle Forms 6i」→「Forms Runtime」
- フォームがアクセスしているデータベースの可用性を検証するために、データベースに接続して問合せを実行します。

10.6.7 Web ブラウザでマルチフォーム・アプリケーションの追加ページにアクセスできない

Forms の .fmx ファイルの場所が、FORMS60_PATH レジストリ変数内に次のように指定されていることを確認します。

```
//HKEY_LOCAL_MACHINE/SOFTWARE/ORACLE/FORMS60_PATH
```

可用性を高めるための Oracle Reports Server の構成

Oracle Fail Safe では、(クライアント / サーバー型および Web ベースの両方の) Oracle Reports アプリケーションに高い可用性を提供します。(Oracle Reports を Web 上で使用できるようにするには、Web サーバーを作成してグループに追加する必要があります。)

Oracle Fail Safe を使用して Oracle Reports を構成すると、可用性が向上して次のような利点が得られます。

- Oracle Reports アプリケーションのダウン時間が最小になります。
- Oracle Reports Server にスケジュールしたジョブが保護されます。
- マスター Oracle Reports Server での障害発生を防ぎます。

この章では、次の項目を説明します。

項目	参照
概要および実装のオプション	11.1 項
スタンドアロン Oracle Reports Server の検出	11.2 項
Oracle Reports Server のグループへの追加	11.3 項
Oracle Reports Server へのクライアント接続	11.4 項
マスター / スレーブ実装のための構成後の手順	11.5 項
可用性の高い Oracle Reports Server へのユーザー・アクセスの制御	11.6 項
Oracle Reports のスケジューリング	11.7 項
可用性の高い Oracle Reports Server に関連する問題のトラブルシューティング	11.8 項

11.1 概要および実装のオプション

Oracle Fail Safe を使用すると、次のいずれかの実装によって、新規および既存の Oracle Reports Server を可用性が向上するように構成できます。

- Web ベースの実装
- クライアント / サーバー実装
- マスター / スレーブ実装（前述の実装 2 つのうちいずれかを使用して設定可能）

次の各項で、これらの実装について詳細に説明します。

層とは、Oracle Reports Server アーキテクチャを構成するコンポーネントの論理的な場所のことです。ただし、各層は、互いに同じクラスタまたはクラスタ・ノードにも、異なるクラスタまたはクラスタ・ノードにも常駐させることができます。また、クラスタ化されていないシステム上にも常駐可能です。クラスタ・ノード上にインストールされたコンポーネントとともに Oracle Fail Safe を使用すると、Web サーバー、Oracle Reports Server および Oracle データベースの可用性が向上し、可用性の高い Web ベースのビジネス・ソリューションを展開できます。これが理想的なソリューションです。ただし、必ずしもすべてのコンポーネントの可用性を高める必要はありません。業務上最も重要なコンポーネントのみを、可用性が高まるように構成できます。

11.1.1 Web ベースの実装

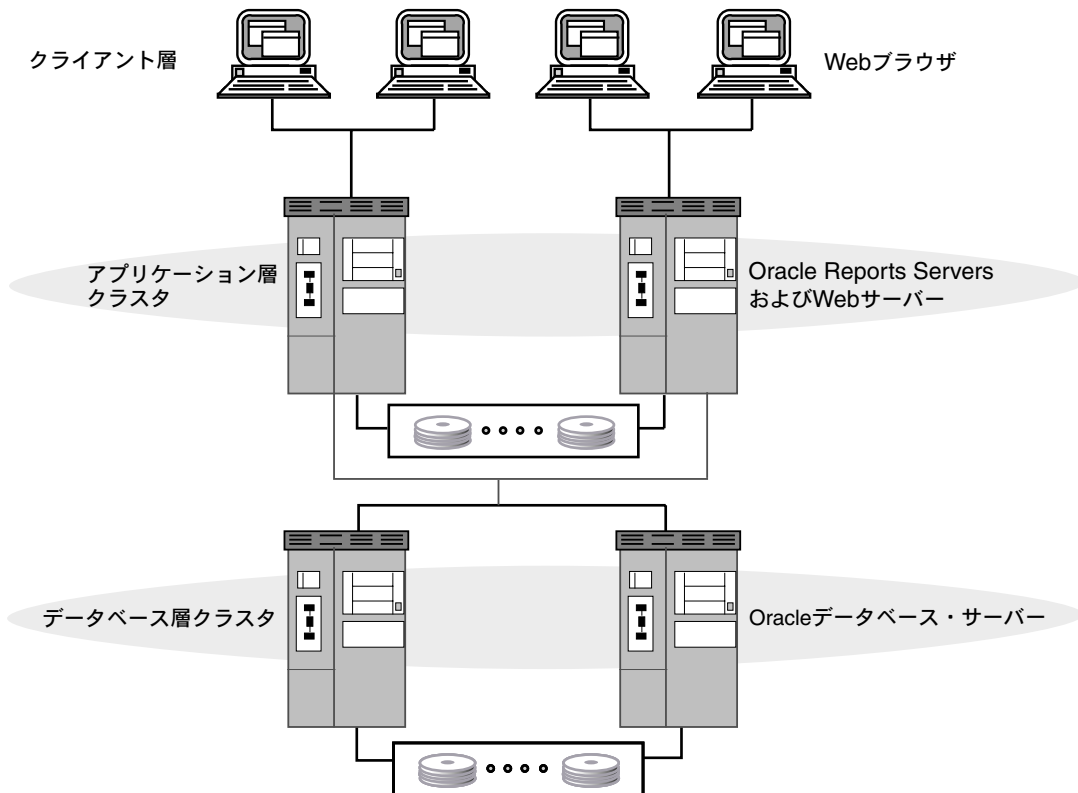
Oracle Reports Server を Reports Web カートリッジ、Reports Web CGI または Reports サブレットとともに使用して、Web ブラウザからレポートを実行できます。（Reports Web カートリッジ、Reports Web CGI および Reports サブレットの設定方法の詳細は、Oracle Reports のマニュアルを参照してください。）

Web ベースの実装では、Oracle Reports Server アーキテクチャは次のような 3 層または 4 層構成になります。

- クライアント層
Web ブラウザが含まれます。
- アプリケーション層
Web サーバーと Oracle Reports Server の両方が単層として含まれる場合と、Web サーバー層と Oracle Reports Server 層の 2 層に分かれる場合があります。
- データベース層
Oracle Reports Server によりアクセスされるデータベースが含まれます。

図 11-1 に、Web 環境における Oracle Reports Server の可用性の高い 3 層構成を示します。この構成では、Oracle Reports Server 層と Web サーバー層が 1 つになってアプリケーション層を形成しています。その他の構成は、Oracle Reports のマニュアルを参照してください。

図 11-1 Oracle Reports の Web 実装のための 3 層アーキテクチャ



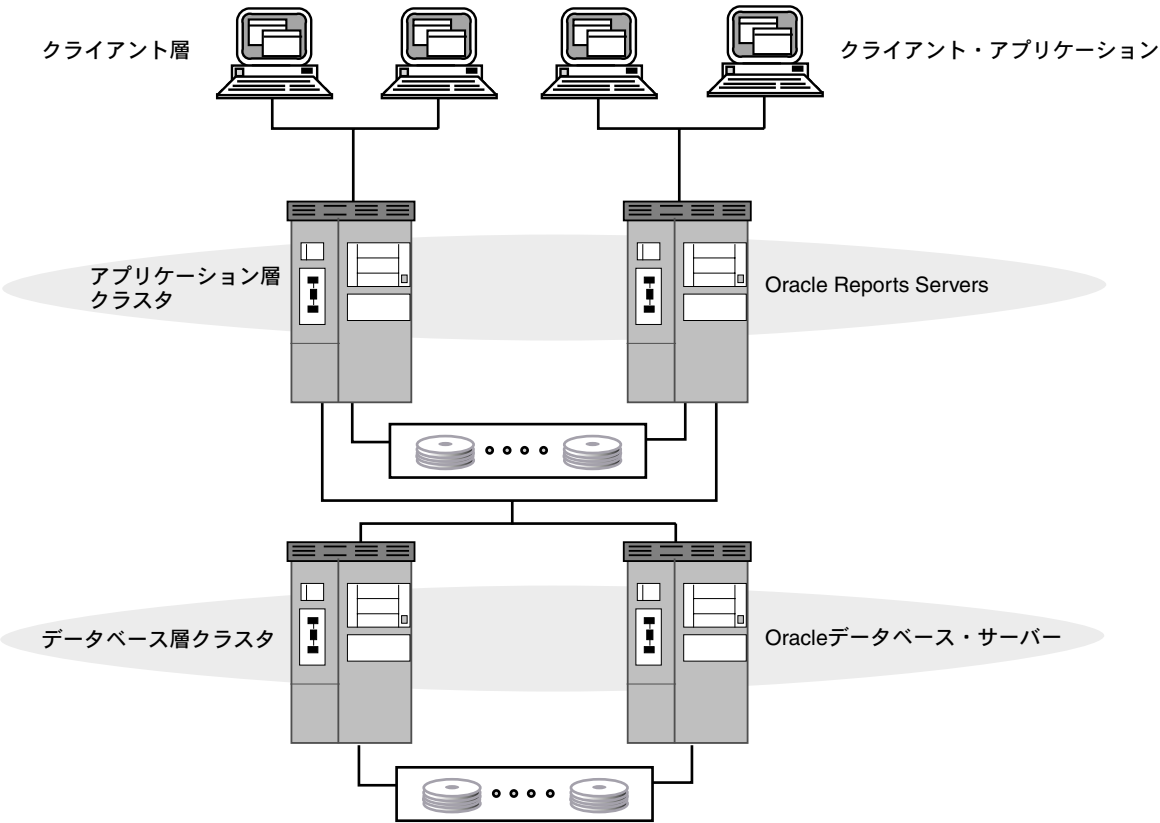
11.1.2 クライアント / サーバー実装

クライアント / サーバー実装では、Oracle Reports Server アーキテクチャは次のような 3 層構成になります。

- クライアント層
クライアント・アプリケーションが含まれます。
- アプリケーション層
Oracle Reports Server が含まれます。
- データベース層
Oracle Reports Server によりアクセスされるデータベースが含まれます。

図 11-2 に、クライアント / サーバー環境における可用性の高い Oracle Reports Server の一般的な構成を示します。

図 11-2 Oracle Reports のクライアント / サーバー実装のための 3 層アーキテクチャ



11.1.3 マスター / スレーブ実装

マスター / スレーブ実装では、パフォーマンスとロード・バランスが向上するように Oracle Reports Server を構成できます。

注意： Oracle Fail Safe のマニュアルでは、**マスター / スレーブ実装**という用語を、Oracle Reports のマニュアルでのクラスタ構成と同じ意味で使用しています。

この構成では、複数の Oracle Reports Server を使用し、1 つをマスターとして、その他をスレーブとして指定します。ユーザーは、マスター Oracle Reports Server にレポート要求を送ります。Oracle Reports Server では、ラウンドロビン・アルゴリズムを使用してレポート要求をスレーブ・サーバーに転送します。いずれかのスレーブが使用不能な場合、マスター Oracle Reports Server はそのスレーブには要求を送りません。

このように、Oracle Fail Safe を使用しなくても、Oracle Reports により可用性の高いレポートが提供されます。ただし、マスター Oracle Reports Server で障害が発生する場合があります。マスター Oracle Reports Server を使用できなくなると、マスター / スレーブ構成全体が使用不能になります。Oracle Fail Safe を使用して可用性の高いマスター Oracle Reports Server を構成することによって、構成内の障害箇所をなくすることができます。

Oracle Fail Safe 環境におけるマスター / スレーブ実装の構成後要件の詳細は、[11.5 項](#)を参照してください。

11.2 スタンドアロン Oracle Reports Server の検出

Oracle Fail Safe Manager を使用して、可用性が高まるように Oracle Reports Server を構成します。Oracle Fail Safe Manager インタフェースでスタンドアロン Oracle Reports Server を表示するには、まずスタンドアロン Oracle Reports Server を検出する必要があります。Oracle Fail Safe Server は、Oracle Reports Server を検出する際、Windows サービス マネージャ内のリストを調べて Oracle Reports Server エントリを見つけ、それが可用性を高めるように構成されているかどうかを判断します。Oracle Fail Safe では、新しく検出されたサービスが Oracle Fail Safe Manager のツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

11.3 Oracle Reports Server のグループへの追加

可用性が高まるように Oracle Reports Server を構成するには、少なくとも 1 つの仮想アドレスが含まれたグループに Oracle Reports Server を追加します。Oracle Fail Safe により、Oracle Reports Server が依存する他のすべてのリソースが追加されます。通常、このグループには次のリソースが含まれます。

- Oracle Reports Server
- キャッシュ・ディレクトリ、ソース・ディレクトリおよびスケジュールしたジョブが配置される 1 つ以上のクラスタ・ディスク
- 1 つ以上の仮想アドレス

Oracle Reports Server を Web 実装にする場合、アプリケーション全体の可用性の向上は、Web サーバー（Oracle Application Server Web サーバーまたは Oracle HTTP Server）をグループに追加することによっても可能です。Web サーバーを追加するグループは、Oracle Reports Server を追加したグループと同じである必要はありません。

Oracle Reports Server をマスター / スレーブ構成におけるマスター・サーバーにする場合も、他の Oracle Reports Server と同じようにグループに追加することによって、可用性の高い Oracle Reports Server を構成します。[11.5 項](#)に、構成後に実行する手順について説明しています。

11.3.1 構成前に

Oracle Reports Server をグループに追加する前に、次のことに注意してください。

- Oracle Reports Server 実行可能ファイルは、Oracle Reports Server を実行するように構成された各クラスタ・ノードのプライベート・ディスクにインストールします。
- その他の必要な Oracle 製品の実行可能ファイルはすべて、Oracle Reports Server を実行するように構成された各クラスタ・ノードのプライベート・ディスクにインストールします。
- Oracle Reports Server のデータ・ファイルと出力ファイルは、どのクラスタ・ノードが Oracle Reports Server のホストであってもアクセスできるように、クラスタ・ディスク上に配置します。
- グループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

11.3.2 構成手順

Oracle Fail Safe には、可用性の高い Oracle Reports Server の構成に必要な手順を自動化するウィザードがあります。既存の Oracle Reports Server のグループへの追加や、Oracle Reports Server の新規作成と追加が可能です。

表 11-1 に、可用性が高まるように Oracle Reports Server を構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、オンライン・ヘルプとチュートリアルを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択するか、または「ヘルプ」→「チュートリアル」を選択します。

表 11-1 Oracle Reports Server の構成手順

手順	処置	説明
1	Oracle Reports ソフトウェアを実行可能にするクラスタ内の各ノードのプライベート・ディスク上に、Oracle Reports ソフトウェアがインストールされていることを確認	インストールの詳細情報は、Oracle Reports のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に、Oracle Reports Server コンポーネントを実行可能にする各クラスタ・ノードに Oracle Reports Server コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle Reports Server コンポーネントを選択します。これで、Oracle Reports の DLL ファイルと NT レジストリ変数がインストールされます。詳しい指示は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
3	Oracle Reports の定義ファイル (.rdf または .rep) をソース・ディレクトリにコピー	定義ファイルは共有記憶装置インターコネクト上のクラスタ・ディスクに配置する必要があります。
4	スタンドアロン・ノード上での Oracle Reports アプリケーションのテスト	Oracle Reports Server をグループに追加する前に、単一ノード上の Oracle Reports アプリケーションをスタンドアロン・モードでテストします。たとえば、任意の書式で任意の宛先にレポートを実行します。
5	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「<Oracle_Home>」→「Oracle Fail Safe Manager」
6	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
7	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）
8	Oracle Reports Server のグループへの追加	Oracle Fail Safe Manager で、「リソース」→「グループに追加」を選択し、「リソースをグループに追加」ウィザードを実行します。このウィザードを使用すると、Oracle Reports Server の選択、そのポート番号の指定、ソース・ファイルの場所の指定、および Reports アプリケーションで使用されるデータベースの選択などが可能です。

表 11-1 Oracle Reports Server の構成手順（続き）

手順	処置	説明
9	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
10	各クライアント・システム上の TNSNAMES.ORA ファイルの変更	仮想サーバーを認識するようにクライアントを構成（各クライアント・システムで TNSNAMES.ORA ファイルを変更）します（11.4.1 項を参照）。マスター / スレーブ実装を使用している場合は、マスター・システムとスレーブ・システム上の TNSNAMES.ORA ファイルを変更します（11.5 項を参照）。
11	Web サーバーをフェイルオーバー用に構成	Web サーバーを新規に作成して追加し、Oracle Reports アプリケーションの Web 上での可用性を向上させます。Web サーバーを作成してグループに追加する方法の詳細は、第 12 章を参照してください。

11.3.3 Oracle Reports Server の構成データ

Oracle Fail Safe Manager には、可用性が向上するように Oracle Reports Server を構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- Oracle Reports Server の可能所有者ノード（クラスタが 3 つ以上のノードから構成される場合、あるいは 2 つのノードから構成されるクラスタで 1 つのノードが使用できない場合）
- Oracle Reports Server に関連付ける仮想アドレス（グループに複数の仮想アドレスが含まれる場合）
- Oracle Reports Server の名前およびそれが実行されるポート
- Oracle Reports Server のキャッシュ・ディレクトリ、ソース・ディレクトリおよびジョブ・ディレクトリの場所
- Oracle Reports Server によりアクセスされるデータベース
- Oracle Reports Server の実行に使用されるアカウント

次の項で、これらの要件を詳細に説明します。

11.3.3.1 ノードの選択

Oracle Reports Server をグループに追加する際に、クラスタが 3 つ以上のノードから構成されている場合は、図 11-3 に示すように、選択済ノードのリストを指定して、Oracle Reports Server の可能所有者となるノードを指定するように求められます。特定のノードを Oracle Reports Server の可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

2.6.6 項では、可能所有者の概念について詳細に説明します。

図 11-3 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle Reports Server をグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを Oracle Reports Server の可能所有者とするのかを指定するように求められます。このような場合、[図 11-4](#)に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 11-4 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



11.3.3.2 仮想アドレス

Oracle Reports Server をグループに追加するには、そのグループに少なくとも 1 つの仮想アドレスが含まれている必要があります。図 11-5 に示すように、Oracle Reports Server を追加するグループ内に複数の仮想アドレスが含まれている場合、Oracle Reports Server と関連付ける仮想アドレスを選択するように要求されます。Oracle Reports アプリケーションにクライアント・アプリケーションからアクセスする場合、アプリケーションから Oracle Reports Server への接続時に指定するホストはその仮想アドレスになります。Oracle Reports アプリケーションに可用性の高い Web サイトからアクセスする場合、ユーザーが接続時に指定する URL は、Reports Server の名前と Web サーバーの仮想アドレスとなります。たとえば、Oracle Reports Server と Web サーバーの両方を可用性の高い構成にした場合、次のような URL を指定して Web 上のレポートにアクセスします。

```
http://ntclu4-6:4000/web_cgi/rwcgi60.exe?server=Reports_Tutorial+report=sample+userid=internal/internal@Tutorial_DB.world+destype=cache+desformat=HTML
```

この URL の構成は次のとおりです。

- ntclu4-6:4000 は、Web サーバーの仮想アドレスおよびポートです。
- web_cgi は、rwcgi60.exe 実行可能ファイルが含まれた物理ディレクトリを指す仮想ディレクトリです。
- Reports_Tutorial は、(TNSNAMES.ORA 内に指定されている) Oracle Reports Server の名前です。
- Tutorial_DB.world は、Oracle Reports Server によりアクセスされるデータベースです。

図 11-5 Oracle Reports の「仮想アドレスの選択」ウィザード・ページ



11.3.3.3 Oracle Reports Server の識別情報

図 11-6 に示すように、Oracle Reports Server の識別情報は、Oracle Reports Server の名前と、Oracle Reports Server が実行されるポートで構成されます。同じクラスタ上で稼働する複数の Oracle Reports Server に対して、同一のポート番号と仮想アドレスの組合せは使用できません。2 つの Oracle Reports Server で同じ仮想アドレスを使用する場合は、異なるポート番号を使用する必要があります。

図 11-6 「Reports の識別情報」ウィザード・ページ



リソースをグループに追加 - Reportsの識別情報 ステップ2/5

グループ名 Reports_Group

このグループに追加したい Reports Server はどれですか?
既存の Reports Server の名前から選択できます。または、
一意な名前を入力することにより Oracle Fail Safe で新しい
Reports Server を作成できます。

Reports Server 名 Reports_Tutorial

Reports Server のリスナー・プロセスが実行するポートはどれ
ですか?

ポート番号: 1949

< 戻る(B) 次へ(N) > キャンセル ヘルプ

11.3.3.4 ソース・ディレクトリ、キャッシュ・ディレクトリおよびジョブ・ディレクトリの場所

これらのディレクトリは、Reports アプリケーションの定義ファイルを配置する場所、出力を書き込む場所、およびジョブのスケジュール情報を配置する場所を指定するものです。既存の Reports Server の場合、これらの値がウィザードのこのページに入力されます。新規の Reports Server の場合、Oracle Fail Safe では、Oracle Reports Server 構成ファイル (<reports-server-name>.ora、ここで <reports-server-name> は Oracle Reports Server の名前) 作成時に指定した値を使用します。Oracle Reports Server 構成ファイルは、常に <Oracle_Home>\Reports60\Server\<reports-server-name>.ora 内にあります。

図 11-7 に示すように、新規の Reports Server に対して次のディレクトリを指定する必要があります。

- キャッシュ・ディレクトリ

Oracle Reports Server の出力ファイルが書き込まれるディレクトリを指定します。このディレクトリ指定は、クラスタ・ディスクおよびディレクトリ、またはファイル共有のいずれかです。

- ソース・ディレクトリ

.rdf または .rep ファイルを配置するディレクトリを指定します。このディレクトリ指定は、クラスタ・ディスクおよびディレクトリ、またはファイル共有のいずれかです。

- ジョブ・ディレクトリ

Oracle Reports Server 作業の実行スケジュールを保持する .dat ファイルを配置するディレクトリを指定します。このディレクトリ指定は、クラスタ・ディスクおよびディレクトリでなければなりません。ジョブをスケジュールしない場合でも、このディレクトリは必ず指定してください。


Oracle Reports Server を新規に作成してグループに追加するときに設定する <reports-server-name>.ora ファイルのパラメータは、次のとおりです。

- キャッシュ・ディレクトリ = cachedir パラメータ
- ソース・ディレクトリ = sourcedir パラメータ
- スケジュールしたジョブ・ディレクトリ = persistfile パラメータ

persistfile パラメータは、<reports-server-name>.dat ファイルの配置場所を示します。<reports-server-name>.dat ファイルには、Oracle Reports Server ジョブの実行スケジュールが含まれます。実行されているノードに関係なく Oracle Reports Server でスケジュールを検索できるように、persistfile パラメータがクラスタ・ディスクを示すようにします。

図 11-7 「Reports のディスク」ウィザード・ページ

リソースをグループに追加 - Reports のディスク ステップ 3/5



キャッシュ・ディレクトリは Reports Server が出力ファイルを書き込む場所です。このディレクトリはどこですか？

キャッシュ・ディレクトリ:

ソース・ディレクトリは Reports 定義ファイル(.rdf)を含みます。このディレクトリはどこですか？

ソース・ディレクトリ:

スケジューリングされたジョブのディレクトリには、Reports のスケジューリング情報が含まれます。このディレクトリはどこですか？

ジョブ・ディレクトリ:

クラスタ・ディスク

< 戻る(B) 次へ(N) > キャンセル ヘルプ

11.3.3.5 Oracle Reports Server によりアクセスされるデータベース

図 11-8 に示すように、Oracle Reports Server では、サービスを提供している Reports アプリケーションを介したデータの表示および更新にどのデータベース（複数も可）を使用するかを認識する必要があります。Oracle Fail Safe では、実行されているノードにかかわらず Oracle Reports Server で接続情報を使用できるように、TNSNAMES.ORA ファイルを Oracle Reports Server の可能所有者であるすべてのクラスタ・ノードのデータベース情報で更新します。

スタンドアロン・データベース、または可用性が高まるように構成したデータベースを指定できます。ただし、ビジネス・ソリューション全体の可用性を高めるには、フェイルセーフの Oracle Reports Server によりアクセスされるデータベースをグループに追加する必要があります。このデータベースを Oracle Reports Server と同じグループに追加する必要はありません。

図 11-8 「Reports のデータベース」ウィザード・ページ



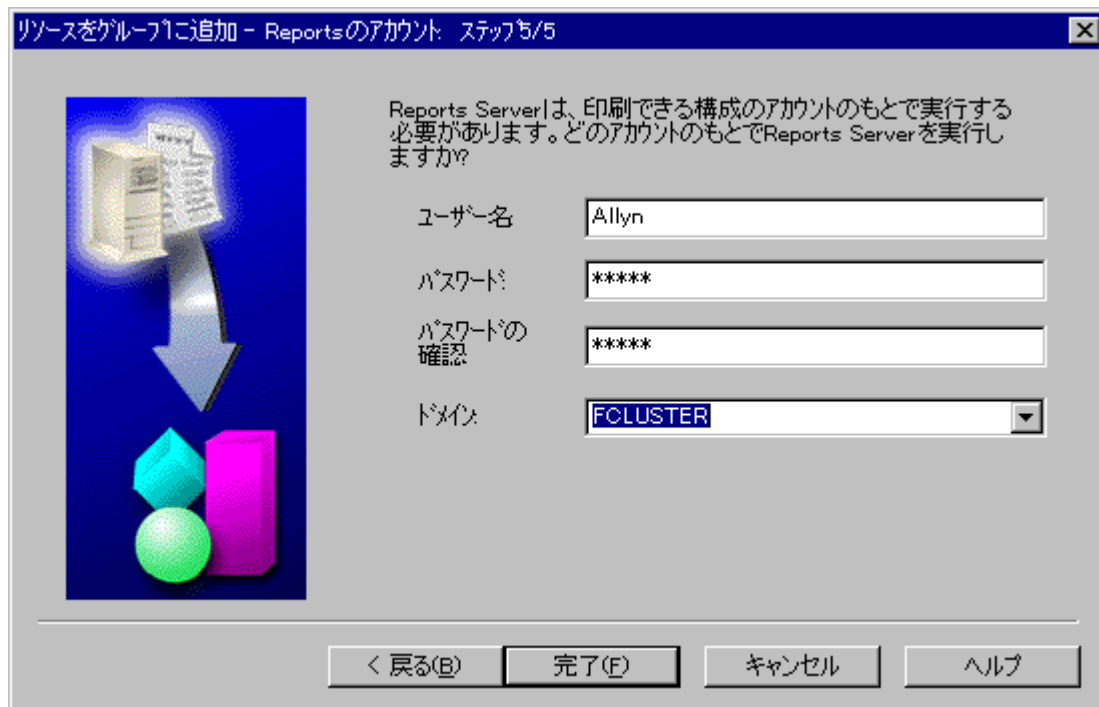
11.3.3.6 Oracle Reports Server の実行に使用されるアカウント

Oracle Reports Server が適切に動作するには、1つ以上のプリンタにアクセスできる必要があります。¹したがって、プリンタへのアクセス権を持つユーザー・アカウントを指定する必要があります。レポートをプリンタに出力しない場合でも、Oracle Reports Server は、プリンタへのアクセス権を持ったアカウントから実行する必要があります。図 11-9 は、この情報を指定するウィザード・ページです。

Oracle Reports Server がサービスとして稼働している場合、Windows の「ユーザー マネージャ」でユーザー・アカウントの「パスワードを無期限にする」オプションが選択されており、そのユーザー・アカウントが Oracle Reports Server の実行と Reports ファイルのアクセスに適切なグループのメンバーであることを確認します。また、デフォルト・プリンタが設定され、ユーザーがそのプリンタに対する印刷権限を持っていることも確認してください。

¹ プリンタは実際になくてもかまいませんが、プリンタ・ドライバはインストールする必要があります。

図 11-9 「Reports のアカウント」ウィザード・ページ



リソースをグループ1に追加 - Reports のアカウント: ステップ 5/5

Reports Server は、印刷できる構成のアカウントのもとで実行する必要があります。どのアカウントのもとで Reports Server を実行しますか?

ユーザー名: Allyn

パスワード: *****

パスワードの確認: *****

ドメイン: FCLUSTER

< 戻る(B) 完了(F) キャンセル ヘルプ

11.4 Oracle Reports Server へのクライアント接続

次の各項では、Oracle Fail Safe を使用して可用性が高まるように構成された Oracle Reports Server にクライアントから接続する方法について説明します。既存の Oracle Reports アプリケーションは、変更することなく、Oracle Fail Safe を使用して構成されている Oracle Reports Server とともに使用できます。

あるクラスタ・ノードが停止または障害を起こした場合、Oracle Reports Server を含むグループは別のクラスタ・ノードに自動的にフェイルオーバーし、ユーザーは数秒以内に Oracle Reports へのアクセスを再開できます。ユーザーが障害の発生した Oracle Reports アプリケーションを Web 上で再開するには、Web ブラウザに URL を再ロードします。Oracle Reports Server コンポーネントの可用性が高められたことで、ユーザーにはアプリケーション全体の可用性が向上したような印象を与えます。

注意： Oracle Reports Server 自体はフェイルオーバー後にリカバリされますが、実行中のレポートの整合性やコンテキストは失われます。フェイルオーバー前に生成されたレポート情報はすべて失われるため、ユーザーまたはアプリケーションによるレポートの再実行が必要です。

Web 上に可用性の高いレポートを表示するには、次のようにします。

1. 仮想アドレスをリスニングするように Web リスナーを構成します。ユーザーは、Web ブラウザで Web リスナーの仮想アドレスと静的 HTML ファイルを指定してレポートにアクセスします。たとえば、Web サーバーの設定によっては、ユーザーは次のような URL を入力します。

```
http://ntclu4-5/reports.html:801
```

この URL では、ntclu4-5 が Web リスナーの仮想アドレス、reports.html が Oracle Reports アプリケーションの静的 HTML ファイル、801 が Web リスナーが動作するポート番号です。Web サイトのホーム・ページの .htm ファイルを編集して、Oracle Reports Server と Web サーバーを追加したグループの仮想アドレス、および Web サーバーのポート番号を参照します。HREF 書式は、次のとおりです。

```
<A HREF="http://virtual_address:port_number/">
```

2. 次のように、(Web サーバーの可能所有者である) 全クラスタ・ノード上の Microsoft Windows レジストリ・キーの値を調整して、Web サーバーで Oracle Reports Server 出力を検索する仮想ディレクトリを反映します。

HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE の下に、次のキーを定義します。

- REPORTS60_PHYSICAL_MAP: Oracle Reports Server キャッシュ (出力) ファイルが書き込まれるディレクトリを指定します。
- REPORTS60_VIRTUAL_MAP: Web サーバーで Oracle Reports Server 出力を検索する仮想ディレクトリを指定します。
- キー・マッピング・ファイルを使用する場合、REPORTS60_CGIMAP 環境変数が (Web サーバーの可能所有者である) 全クラスタ・ノード上に定義され、キー・マッピング・ファイルの検索場所が指定されていることを確認します。

11.4.1 Oracle Reports Server へのクライアント・アクセスのための TNSNAMES.ORA ファイルの更新

Oracle Fail Safe では、Oracle Reports Server の可能所有者であるクラスタ・ノード、および「リソースをグループに追加」ウィザードを実行しているシステムの TNSNAMES.ORA ファイルを自動的に更新し、Oracle Reports Server をグループに追加します。

ただし、クライアントがノードの物理アドレスではなく仮想アドレスで Oracle Reports Server にアクセスできるように、クライアント・システム上の TNSNAMES.ORA ファイルを編集する必要があります。

注意： Web ブラウザを使用して Oracle Reports Server にアクセスするクライアント・システムでは、TNSNAMES.ORA ファイルを更新する必要はありません。

Oracle Fail Safe 環境で Oracle Reports Server にアクセスする必要のあるクライアント・システムを更新するには、次の手順に従います。

1. 次のいずれかの方法で、Oracle Reports Server で使用する仮想アドレスを判断します。
 - ツリー・ビューで Oracle Reports Server リソースを選択し、「依存性」タブをクリックします。リストされている仮想アドレスを書きとめます。
 - クラスタ・ノードの 1 つで TNSNAMES.ORA ファイルの Oracle Reports Server エントリを検索します。(Oracle Fail Safe では、Oracle Reports Server の可能所有者であるクラスタ・ノード上の TNSNAMES.ORA ファイルを更新します。)
2. Oracle Reports Server にアクセスが必要なクライアント・システム上の TNSNAMES.ORA ファイルを編集します。Oracle Reports Server の TNSNAMES.ORA ファイルのエントリは、次のようになります。

```
reportserver.world =
  (ADDRESS =
    (PROTOCOL = TCP)
    (Host = NTCLU-162) <- 仮想アドレス
    (Port=9100)          <- 「リソースをグループに追加」ウィザードで
                           指定されているポート
  )
```

11.5 マスター / スレーブ実装のための構成後の手順

マスター / スレーブ構成を使用する場合、可用性が高まるように Oracle Reports Server を構成した後、さらにいくつかの手順が必要になります。

この項では、マスター / スレーブの Oracle Reports Server の実装方法について手順ごとに説明します。この項の例で説明する内容は、次のとおりです。

- マスター / スレーブの例に必要なバックグラウンド情報 (11.5.1 項)
- マスター / スレーブ間の通信を使用可能にする方法 (11.5.2 項)
- 既存の実装へのサーバーの追加 (11.5.3 項)

11.5.1 マスター / スレーブ例の概要

表 11-2 に、マスター / スレーブ構成例に使用するサーバー・システムを示します。

表 11-2 マスター / スレーブ実装の例

Reports Server 名	システム名	マスターまたはスレーブ	TNS 名
Reports_Master	ntclu4-3	マスター	ntclu4-5 (仮想アドレス)
nt-2	nt-2	スレーブ	nt-2
sun-1	sun-1	スレーブ	sun-1

各システムの状態は次のとおりです。

- Oracle Reports Server コンポーネントがインストールされています。
- 中央ファイル・サーバーが実行され、(レポート定義ファイルが格納される) ソース・ディレクトリ、(キャッシュに入れられたすべてのレポートの出力先の) キャッシュ・ディレクトリ、(Oracle Reports Server ジョブのスケジューリング・データ・ファイルが置かれる) ジョブ・ディレクトリの 3 つのディレクトリとともに設定されています。この中央ファイル・サーバーは、Oracle Reports Server のグループへの追加、ソース・ディレクトリ、キャッシュ・ディレクトリおよびジョブ・ディレクトリの指定時に指定した共有クラスター・ディスクです。

各 Oracle Reports ランタイム・エンジンではその出力を中央キャッシュに書き込み、各 Oracle Reports ランタイム・エンジンでは中央ソース・ディレクトリからレポート定義ファイルを読み込みます。中央ソース・ディレクトリがあるため、すべてのランタイム・エンジンで同じレポートを実行することになります。また、更新したレポート定義ファイルを複数の場所にコピーする必要がなくなります。中央キャッシュにより、マスター・サーバーでは重複するジョブに対してサービスを提供でき、ジョブは各スレーブ・サーバーのローカル・ディスクを経由せずに指定許容範囲内で実行されます。

- (出力が常にデフォルト・プリンタに送られる場合以外は) すべての Oracle Reports ランタイム・エンジンで、プリンタに同一の別名が使用されます。

11.5.2 マスター / スレーブ間の通信を使用可能にする方法

マスター / スレーブ間の通信を使用可能にするには、マスター Oracle Reports Server を実行可能なすべてのクラスタ・ノード上と、スレーブ Oracle Reports Server を実行可能なマシン上で、TNSNAMES.ORA ファイルと Oracle Reports Server 構成ファイルを変更する必要があります。

これは、マスター・サーバーが可用性を高めるように構成されているかどうかには関係ありません。ただし、マスター・サーバーが可用性を高めるように構成されている場合は、マスター・サーバーが実行されているマシンのホスト名ではなく、TNSNAMES.ORA ファイル内のマスター・サーバーの仮想アドレスを指定してください。次の各項で、手順を詳細に説明します。

(この例では、Reports_Master.world、nt-2.world および sun-1.world がサーバー・インスタンス名、.world が SQLNET.ORA ファイルの NAMES.DEFAULT_DOMAIN 設定に指定されているドメインです。NAMES.DEFAULT_DOMAIN 設定が SQLNET.ORA ファイル内に定義されていない場合は、サーバー・インスタンス名から .world を省いてください。)

11.5.2.1 マスター・サーバーのための調整

Reports_Master サーバーが含まれたグループが常駐するクラスタ・ノードで、マスター・サーバーに対するスレーブ・サーバーの特定、各スレーブ・サーバーの中央キャッシュ・ディレクトリとランタイム・パラメータの指定を次の手順に従って実行します。

1. テキスト・エディタで、Reports_Master サーバーが実行されているクラスタ・ノードの `<Oracle_Home>%NET80%ADMIN` ディレクトリに入っている TNSNAMES.ORA ファイルを開き、次の行を追加してスレーブ・サーバーを指定します。

```
nt-2.world=(ADDRESS=(PROTOCOL=tcp) (HOST=nt-2) (PORT=1949))
sun-1.world=(ADDRESS=(PROTOCOL=tcp) (HOST=sun-1) (PORT=1949))
```

2. `<Oracle_Home>%REPORT60%SERVER` ディレクトリに入っている Reports_Master.ORA ファイル (マスター Oracle Reports Server 構成ファイル) を編集し、次のように CLUSTERCONFIG パラメータを設定してマスターに対するスレーブ・サーバーを指定します。

```
sourcedir="x:%Source"
cachedir="x:%Cache"
persistfile="x:%Jobs"
.
.
.
clusterconfig="(server=nt-2
minengine=0
maxengine=2
```

```
initengine=2
cachedir="W:¥Cache")
(server=sun-1
minengine=0
maxengine=2
initengine=2
cachedir="/share/Cache")"
```

3. Reports_Master サーバーを一度オフラインにした後、Oracle Fail Safe Manager または FSCMD コマンドを使用して再度オンラインにします。これを実行するまで、新しい構成は有効になりません。
4. 構成ファイル (TNSNAMES.ORA および Reports_Master.ORA) が、マスター Reports Server の可能所有者である他のクラスタ・ノードで更新されるように、Reports_Master サーバーが含まれたグループを検証します。

次の点に注意してください。

- 各クラスタ・ノードは、X: ドライブを使用して中央ファイル・サーバーにマップされます。
- サーバー・パラメータは、スレーブ・サーバーの TNS サービス・エントリ名です。
- minengine、maxengine および initengine の値は、例として指定されています。マスター / スレーブ実装では、これらの値を指定する必要はありません。スレーブ・サーバーに対するこれらのパラメータ設定の詳細は、Oracle Reports のマニュアルを参照してください。
- cachedir パラメータは、マスター・サーバーの中央キャッシュ・ディレクトリです。
- NT マシンと UNIX マシンとでは、キャッシュ・ディレクトリの設定が異なります。すべてのサーバーを、同じ定義による共有ファイル・システムと対応させる必要はありません。(この例では、マスターが X: ドライブに、NT スレーブは W: ドライブに、Sun (UNIX) スレーブは /share/Cache ネットワーク共有ドライブにマップされます。ただし、すべての定義は、可用性が高まるように Oracle Reports Server を構成した際に追加した同一の共有クラスタ・ディスクを指しています。)
- スレーブ・サーバーの REPORT60_PATH 環境変数は、sun-1 システムの場合 /share/Source に、nt-2 システムの場合 W:¥Source に設定する必要があります。

11.5.2.2 スレーブ・サーバーのための調整

次の手順では、各スレーブ・サーバーに対する調整について説明します。各スレーブ・システムが仮想サーバー・アドレスで Reports_Master サーバーを検出できるように、各システムの TNSNAMES.ORA ファイルを更新することが主な目的です。

1. テキスト・エディタで <Oracle_Home>¥NET80¥ADMIN ディレクトリ内の TNSNAMES.ORA ファイルを開き、次の行を追加して、スレーブ Oracle Reports Server によりアクセスされるマスター Oracle Reports Server およびデータベースを指定します。

```
Reports_Master.world=(ADDRESS=(PROTOCOL=tcp) (HOST=ntclu4-5) (PORT=1949))
Reports_db.world=(ADDRESS=(PROTOCOL=tcp) (HOST=ntclu4-7) (PORT=7687))
```

Reports_Master.world のホストは、可用性の高いマスター Oracle Reports Server の仮想アドレスです。同様に、Oracle Reports Server によりアクセスされるデータベースの可用性を高めた場合、そのデータベースの HOST エントリはデータベースの仮想アドレスにする必要があります。

2. <Oracle_Home>%REPORT60%SERVER ディレクトリ内の nt-2.ora ファイル (Oracle Reports Server 構成ファイル) を開き、INITENGINE パラメータを 0 に設定します。これによって、マスター・サーバーにより起動されるランタイム・エンジンは、起動時に作成されたランタイム・エンジンに限られます。
3. sun-1 システムで手順 1 と 2 を繰り返します。ただし、手順 2 では、<Oracle_Home>%Reports60%Server ディレクトリ内の sun-1.ora 構成ファイルを編集してください。

11.5.3 既存のマスター / スレーブ実装へのスレーブ・サーバーの追加

既存のマスター / スレーブ実装に新規のスレーブ・サーバーを追加する場合、マスターとスレーブ両方のサーバー・システムの TNSNAMES.ORA ファイルと Oracle Reports Server 構成ファイルを調整する必要があります。

表 11-2 に示したような実装において、Sun Solaris システム (sun-2) で稼働している sun-2 という名前のスレーブ・サーバーを追加するとします。次の各項では、Reports_Master サーバーが常駐するクラスタ、およびスレーブが稼働しているシステムで行う必要のある変更について説明します。

この例では、nt-2.world、sun-1.world および sun-2.world がサーバー・インスタンスの名前で、.world が SQLNET.ORA ファイルの NAMES.DEFAULT_DOMAIN 設定に指定されているドメインです。NAMES.DEFAULT_DOMAIN 設定が SQLNET.ORA ファイル内に定義されていない場合は、サーバー・インスタンス名から .world を省いてください。

11.5.3.1 マスター Oracle Reports Server のための調整

Reports_Master サーバーが含まれたグループが常駐するクラスタ・ノードで、次のようにします。

1. <Oracle_Home>%NET80%ADMIN ディレクトリにある TNSNAMES.ORA ファイルを開き、次のエントリを追加します。

```
sun-2.world=(ADDRESS=(PROTOCOL=tcp) (HOST=sun-2) (PORT=1949))
```

2. Reports_Master.ora ファイル (Oracle Reports Server 構成ファイル) を開き、次の太字部分を既存の CLUSTERCONFIG パラメータに追加します。

```
clusterconfig="(server=nt-2
minengine=0
maxengine=2
```

```

initengine=2
cachedir="W:¥Cache")
(server=sun-1
minengine=0
maxengine=2
initengine=2
cachedir="/share/Cache")
(server=sun-2
minengine=0
maxengine=4
initengine=4
cachedir="/share/Cache")"

```

3. Reports_Master サーバーを一度オフラインにした後、Oracle Fail Safe Manager または FSCMD コマンドを使用して再度オンラインにします。これを実行するまで、新しい構成は有効になりません。
4. 構成ファイル (TNSNAMES.ORA および Reports_Master.ORA) が、マスター Reports Server の可能所有者である他のクラスタ・ノードで更新されるように、Reports_Master サーバーが含まれたグループを検証します。

11.5.3.2 スレーブ・サーバーのための調整

次の手順では、スレーブ・サーバーに対する調整について説明します。スレーブ・システムが仮想サーバー・アドレスで Reports_Master サーバーを検出できるように、システムの TNSNAMES.ORA ファイルを更新することが主な目的です。

1. <Oracle_Home>/NET80/ADMIN ディレクトリ内の TNSNAMES.ORA ファイルを開き、次の行を追加して、スレーブ Oracle Reports Server によりアクセスされるマスター Oracle Reports Server およびデータベースを指定します。

```

Reports_Master.world=(ADDRESS=(PROTOCOL=tcp) (HOST=ntclu4-5) (PORT=1949))
Reports_db.world=(ADDRESS=(PROTOCOL=tcp) (HOST=ntclu4-7) (PORT=7687))

```

Reports_Master.world のホストは、可用性の高いマスター Oracle Reports Server の仮想アドレスです。同様に、Oracle Reports Server によりアクセスされるデータベースの可用性を高めた場合、そのデータベースの HOST エントリはデータベースの仮想アドレスにする必要があります。

2. <Oracle_Home>/REPORT60/SERVER ディレクトリ内の sun-2.ora ファイル (Oracle Reports Server 構成ファイル) を開き、INITENGINE パラメータを 0 に設定します。(これによって、マスター・サーバーにより起動されるランタイム・エンジンは、起動時に作成されたランタイム・エンジンに限られます。) さらに、次のエントリを追加します。

```

Reports_Master.world=(ADDRESS=(PROTOCOL=tcp) (HOST=ntclu4-5) (PORT=1949))

```

この例では、ntclu4-5 は、マスター Oracle Reports Server が実行されている物理ホストではなく、マスター Oracle Reports Server と関連付けられた仮想アドレスです。

11.6 可用性の高い Oracle Reports Server へのユーザー・アクセスの制御

Oracle Reports リリース 6i では Oracle Reports へのユーザー・アクセス制御に Oracle WebDB を使用できます。可用性が向上するように構成した Oracle Reports Server に対して、この方法で Oracle WebDB を使用できます。Oracle Reports リリース 6i のマニュアルにある指示に従ってください。ただし、次のネットワーク調整を行って、Oracle Reports Server の物理アドレスのかわりに仮想アドレスを使用するようにしてください。この例では、可用性が高まるように Oracle WebDB インスタンスを構成することを示します。

1. Oracle Fail Safe Manager を使用し、レポートへのアクセス制御に使用する Oracle WebDB インスタンスの可用性を向上させます。(Oracle WebDB インスタンスをグループに追加します。)
2. 次のように、Oracle WebDB サービス・エントリを TNSNAMES.ORA に追加します。
 - a. Oracle Reports Server を含むグループが常駐するクラスタ・ノード上の `<Oracle_Home>\NET80\ADMIN` ディレクトリにある TNSNAMES.ORA ファイルを開きます。
 - b. 次の Oracle WebDB サービス・エントリを追加します。

```
sec-rep.world =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = webdb-virtual-address.domain)
      (PORT= 1521)
    )
    (CONNECT_DATA = (SID = ORCL)
```

3. (Oracle Reports Server を含むグループが常駐するクラスタ・ノード上の `<Oracle_Home>\REPORT60\SERVER` ディレクトリにある) `report-server.ora` 構成ファイルを開き、次の構文を追加します。

```
SECURITYTNSNAME="sec-rep"
```

この例で `sec-rep` は、TNSNAMES.ORA ファイルで定義された Oracle WebDB サーバー・インスタンスの仮想アドレスです。

4. Oracle Reports Server を一度オフラインにした後、Oracle Fail Safe Manager または FSCMD コマンドを使用して再度オンラインにします。これを実行するまで、新しい構成は有効になりません。
5. Oracle Reports Server の可能所有者である他のクラスタ・ノードで手順 2 と 3 を繰り返します。

11.7 Oracle Reports のスケジューリング

スタンドアロンの Oracle Reports Server と同様に、可用性の高い Oracle Reports Server に対してもジョブをスケジュールできます。ジョブのスケジュール情報は、通常ファイル (<reports-server-name>.dat) に格納されます。Oracle Reports リリース 6i では、ジョブの実行中に必ず Oracle Reports Server キュー・アクティビティのスナップショットをとるようにデータベースを設定できます。

可用性が高まるように Oracle Reports Server を構成する場合、「リソースをグループに追加」ウィザードにより、ジョブ・スケジュール・ファイル (<reports-server-name>.dat) の場所を指定するように要求されます。Oracle Fail Safe では、Oracle Reports Server が稼働しているクラスタ・ノードにかかわらず Oracle Reports Server で .dat ファイルを検出できるように、共有クラスタ・ディスク上に <reports-server-name>.dat ファイルを配置する必要があります。フェイルオーバーにおいても、後で実行するようにスケジュールされている Oracle Reports アプリケーションは、スケジュールどおりに実行されます。つまり、スケジュール済のレポート実行はフェイルオーバーによる影響を受けません。

ただし、フェイルオーバーが発生すると、実行中の Oracle Reports アプリケーションはすべて失われます。Oracle Reports アプリケーションの再試行は、そのアプリケーションを再試行するようにスケジュールしており、再試行の回数が上限に達していない場合に限られます。

11.7.1 ジョブ・リポジトリを可用性の高い Oracle Reports Server とともに使用

Oracle Reports リリース 6i では、ジョブの実行中に必ず Oracle Reports Server キュー・アクティビティのスナップショットをとるようにデータベースを設定する機能を提供します。可用性が向上するように構成した Oracle Reports Server に対して、この方法でデータベースを使用できます。Oracle Reports 6i のマニュアルの指示に従ってください。ただし、次のことも必ず実行してください。

1. ジョブ・リポジトリとして使用するデータベースをグループに追加し、このデータベースの可用性を向上させます。
2. 次のように、データベース・サービス・エントリを TNSNAMES.ORA に追加します。
 - a. Oracle Reports Server を含むグループが常駐するクラスタ・ノード上の <Oracle_Home>\NET80¥ADMIN ディレクトリにある TNSNAMES.ORA ファイルを開きます。
 - b. 次のデータベース・サービス・エントリを追加します。

```
sec_rep.world =
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)
      (HOST = database-virtual-address.domain)
      (PORT= 1521)
```

- ```
)
(CONNECT_DATA = (SID = ORCL)
```
3. Oracle Reports Server を含むグループが常駐するノード上の Oracle Reports Server 構成ファイルに、REPOSITORYCONN パラメータを追加します。このパラメータの追加については、Oracle Reports リリース 6i のマニュアルに記述されている指示に従ってください。
  4. Oracle Reports Server の可能所有者である他のクラスタ・ノードで手順 2 と 3 を繰り返します。
  5. Oracle Fail Safe Manager あるいは FSCMD コマンドを使用して、Oracle Reports Server の停止と再起動を実行し、構成ファイルの変更が受け入れられるようにします。

## 11.8 可用性の高い Oracle Reports Server に関連する問題のトラブルシューティング

次の各項では、可用性が高まるように構成した Oracle Reports Server の使用時に発生する具体的な問題のトラブルシューティングについて説明します。Fail Safe リソースのトラブルシューティングに関する一般的な情報は、[第 6 章](#)に記載しています。Oracle Reports Server のトラブルシューティングに関する一般的な情報は、Oracle Reports リリース 6i のマニュアルを参照してください。

Oracle Reports Server を含むグループに「グループの検証」コマンドを発行すると、Oracle Fail Safe では、クラスタ・レジストリにある情報を構成ファイル内に記述されている情報と比較します。情報が一致しない場合、Oracle Fail Safe により問題を修正するかどうかを尋ねられます。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合

### 11.8.1 Web サーバーで CGI の RWCGI60.EXE ファイルを見つけられない

CGI 実行可能ファイルが見つからないというエラー・メッセージを受け取った場合は、次のことを実行します。

- Web サーバー構成で仮想ディレクトリが正しく設定されていることを確認します。
- RWCGI60.EXE 実行可能ファイルが関連物理ディレクトリにあるかどうかを確認します。

## 11.8.2 Oracle Reports Server で RWCGI60.EXE ファイルをダウンロード

Oracle Reports Server で RWCGI60.EXE ファイルを実行せずにダウンロードしようとする場合、Web サーバー構成内の許可を確認してください。仮想ディレクトリ（たとえば、web\_cgi など）は、「実行」に設定されている必要があります。

## 11.8.3 ユーザーが Oracle Reports Server と通信できない

ユーザーが Oracle Reports Server と通信できない場合は、次の処置を実行します。

- Oracle Reports Server リソースがオンラインになっていることを確認します。
- Web ブラウザに URL として指定したサーバー名、または HTML ファイルに指定されているサーバー名が、正しく設定されていることを確認します。次の例では、太字の部分でサーバー名を示します。

```

```

サーバー名は TNSNAMES.ORA ファイルに定義する必要があります。

## 11.8.4 Web ブラウザで Oracle Reports 定義ファイルが見つからない

Web ブラウザで Oracle Reports 定義（.rdf または .rep）ファイルを見つけられない場合は、次の処置を実行します。

- Web ブラウザに URL の一部として指定した Oracle Reports 名、または HTML ファイルに指定されている Oracle Reports 名が、正しく設定されていることを確認します。次の例では、太字の部分で Oracle Reports 名を示します。

```

```

ファイル名の .rdf または .rep ファイル拡張子を指定する必要はありません。

- Oracle Reports 定義ファイルがソース・ディレクトリにあることを確認します。（ソース・ディレクトリは、クラスタ・ディスク上の Oracle Reports 定義（.rdf または .rep）ファイルを含むディレクトリです。）

## 11.8.5 Oracle Reports アプリケーションが正しく実行されない

Oracle Reports アプリケーションが Web ブラウザで正しく実行されない場合は、次のことを実行します。

- レポートを Reports ランタイムでローカルに実行します。Reports ランタイムを起動するには、Windows のタスクバーで、次の順に選択します。  
「スタート」→「プログラム」→「Oracle Reports 6i」→「Reports Runtime」
- データベースに接続して問合せを実行し、Oracle Reports アプリケーションがアクセスしているデータベースが使用できるかどうかを確認します。

## 11.8.6 ユーザーが Reports を出力できない

ユーザーが Reports アプリケーションから出力できない場合は、次の処置を実行します。

- Web サーバー仮想ディレクトリ（出力）が Oracle Reports キャッシュ・ディレクトリを指していることを確認します。
- 次の Windows レジストリ・エンジンが正しく設定されていることを確認します。
  - //HKEY\_LOCAL\_MACHINE/SOFTWARE/ORACLE/REPORTS60\_PHYSICAL\_MAP はキャッシュ・ディレクトリを指している必要があります。
  - //HKEY\_LOCAL\_MACHINE/SOFTWARE/ORACLE/REPORTS60\_VIRTUAL\_MAP は、Web サーバーの仮想ディレクトリを指している必要があります。

---

## 可用性を高めるための Web サーバーの構成

---

Oracle Fail Safe を使用すると、可用性が高まるように Oracle Application Server (OAS) または Oracle HTTP Server を構成できます。

あるクラスタ・ノードが停止または障害を起こした場合、Web サーバーを含むグループは別のクラスタ・ノードに自動的にフェイルオーバーし、クライアントは、通常、数秒以内に Web サイトへのアクセスを再開できます。Web クライアントは、ブラウザの「再読み込み」ボタンまたは「更新」ボタンをクリックするか、または Web サーバーの（仮想アドレスを指す）URL を再入力して、Web サイトに再接続します。

この章では、次の項目について説明します。

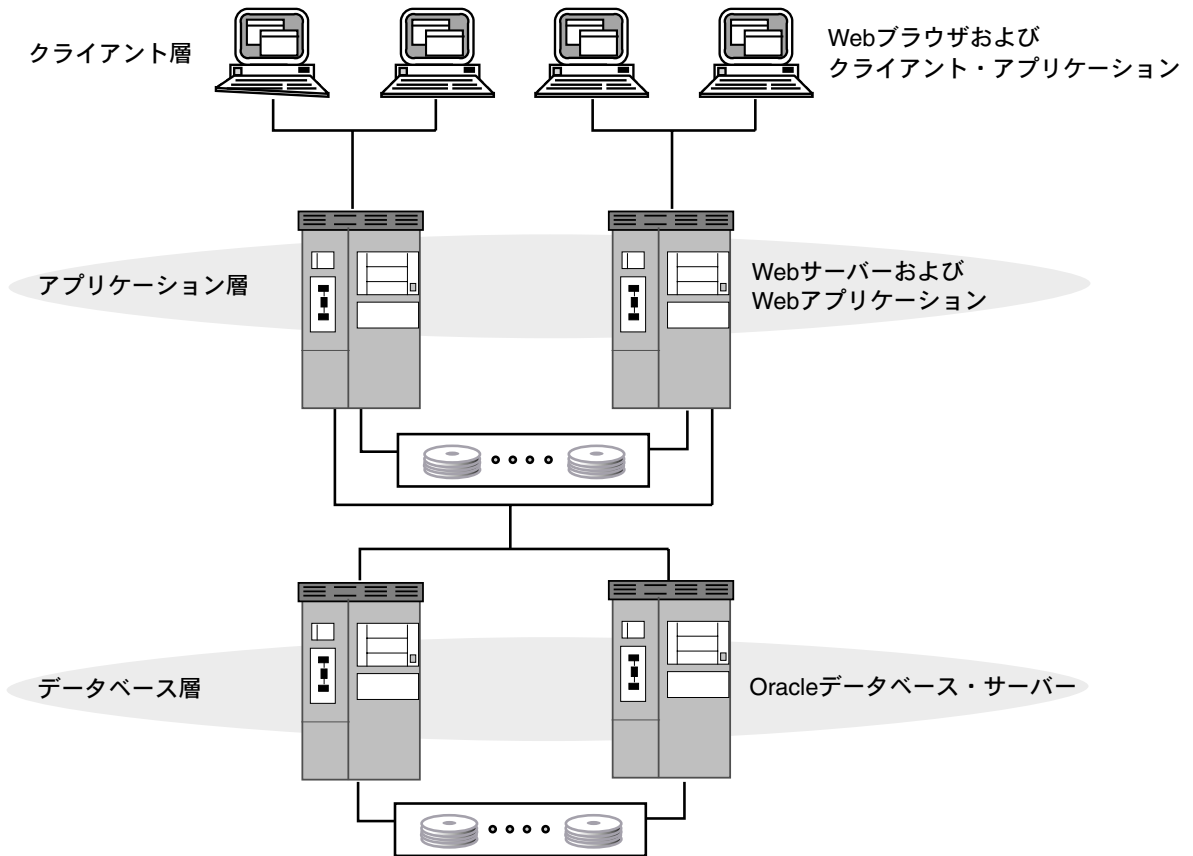
項目	参照
概要	<a href="#">12.1 項</a>
スタンドアロン Oracle Applilcation Server Web リスナーの検出	<a href="#">12.2 項</a>
Oracle Application Server Web リスナーのグループへの追加	<a href="#">12.3 項</a>
Oracle Application Server Web リスナーへのクライアント接続	<a href="#">12.4 項</a>
Oracle Application Server Web リスナーのグループからの削除	<a href="#">12.5 項</a>
Oracle Application Server のセキュリティ要件	<a href="#">12.6 項</a>
Oracle Application Server のインストールに関するヒント	<a href="#">12.7 項</a>
可用性の高い Oracle Application Server Web リスナーに関連する問題のトラブルシューティング	<a href="#">12.8 項</a>
Oracle HTTP Server の概要	<a href="#">12.9 項</a>
スタンドアロン Oracle HTTP Server の検出	<a href="#">12.10 項</a>
Oracle HTTP Server のグループへの追加	<a href="#">12.11 項</a>
Oracle HTTP Server へのクライアント接続	<a href="#">12.12 項</a>

項目	参照
<a href="#">Oracle HTTP Server のグループからの削除</a>	<a href="#">12.13 項</a>
<a href="#">Oracle HTTP Server のセキュリティ要件</a>	<a href="#">12.14 項</a>
<a href="#">可用性の高い Oracle HTTP Server に関連する問題のトラブルシューティング</a>	<a href="#">12.15 項</a>

## 12.1 概要

[図 12-1](#) に、アプリケーション層に可用性が高まるように Oracle Web サーバーおよび Web アプリケーションを構成し、データベース層に Oracle データベースを構成し、クライアント層に Web ブラウザをインストールして 3 層アーキテクチャを作成した、2 つのノードから成るクラスタを示します。

図 12-1 可用性の高い Web 構成のための 3 層アーキテクチャ



多数の冗長なコンポーネントがロード・バランス Web サーバーを介してアクセスされるようなアーキテクチャにすることによって、構成全体の可用性を高めることができます。これには、Oracle Fail Safe で Web サーバーのみをフェイルオーバー用に構成し、障害の検出と正常に機能しているコンポーネントへの要求の再配布をこの Web サーバーに依頼します。1 つ以上のアプリケーション・サーバーとデータベース・サーバーを追加すると、さらに可用性が高まります。これにより、Web サーバーの受信クライアント要求の処理に必要なリソースを常に使用できます。

Oracle Fail Safe により、次のコンポーネントの可用性を高めることができます。

- Web サーバー：
  - Oracle Application Server
  - Oracle HTTP Server
- Web アプリケーション：
  - Oracle Applications Concurrent Manager
  - Oracle Forms
  - Oracle Reports
  - Oracle WebDB
  - 汎用サービスとして構成可能なその他の Oracle Applications コンポーネントおよびアプリケーション
  - ご使用の Web サーバーによりサポートされる標準の Web アプリケーション

たとえば、共通ゲートウェイ・インタフェース (CGI)、Perl (Practical Extraction and Report Language)、Active Server Page (ASP)、Visual Basic または JavaScript を使用する Web アプリケーションは、Oracle Fail Safe を使用して可用性が高まるように構成することができます。(Oracle Application Server カートリッジは標準 Web アプリケーションと見なされないため、Oracle Fail Safe ではサポートされません。)

---

**注意：** 異なるバージョンの Java 実装が実行される様々なブラウザで Web アプリケーションを実行する際の問題を解決するため、JInitiator の使用をお勧めします。JInitiator を使用することによって、異なるリリースのブラウザ間でも Java 仮想マシンの一貫性が保たれます。JInitiator の詳細は、JInitiator のマニュアルを参照してください。

---

## 12.2 スタンドアロン Oracle Application Server Web リスナーの検出

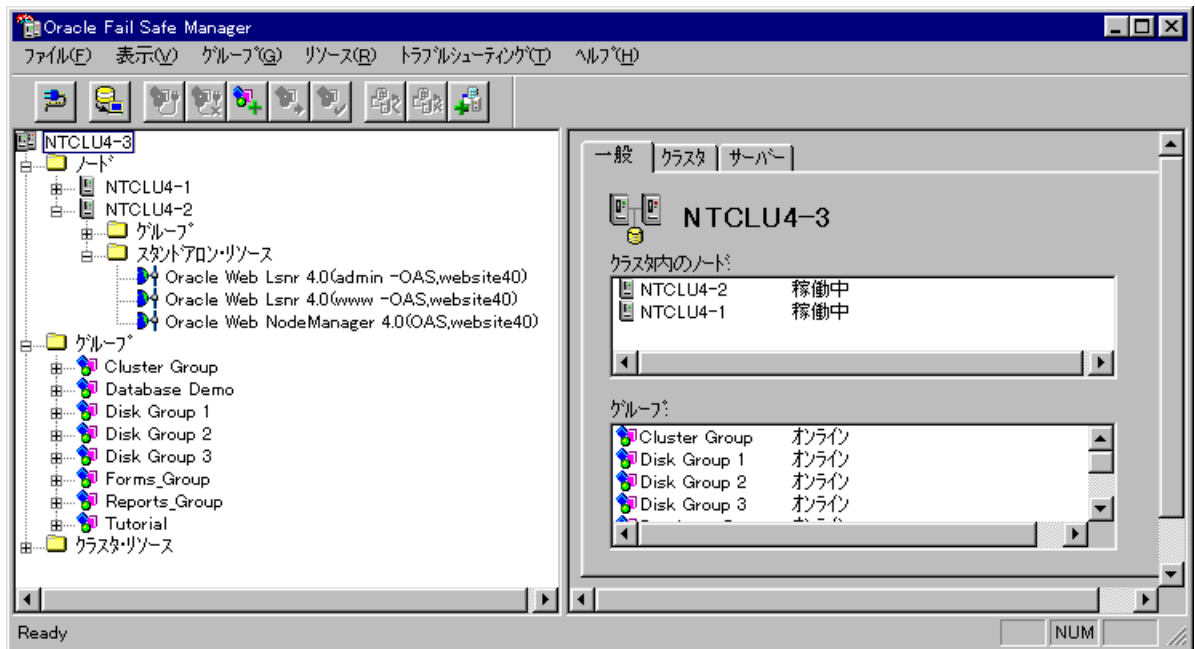
Oracle Fail Safe Manager インタフェースでスタンドアロン Oracle Application Server Web リスナーを表示するには、まずスタンドアロン Oracle Application Server Web リスナーを検出する必要があります。Oracle Fail Safe Server は、各クラスタ・ノード上のすべての Windows 汎用サービスの中で「OracleWWWListener40」で始まる名前を検索することにより、クラスタの各ノード上のスタンドアロン Oracle Application Server Web リスナーを検出します。クラスタ・リソースではないサービスが検出されると、スタンドアロン・リソースとして一覧表示されます。他のリソースとは異なり、Oracle Application Server Web リスナーはサービス名ではなく、サービス表示名で表示されるので注意してください。これによ



り、Oracle Fail Safe では、別の Oracle Application Server インストールで同じ名前を持つ Web リスナーを区別できます。

図 12-2 は、Oracle Application Server インストールの一部としてインストールされているスタンドアロンの Oracle Application Server Web リスナーが Oracle Fail Safe により検出された後の、Oracle Fail Safe Manager のツリー・ビューを示しています。デフォルトの Oracle Application Server Web リスナーが、ツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

図 12-2 スタンドアロン Oracle Application Server Web リスナー



## 12.3 Oracle Application Server Web リスナーのグループへの追加

可用性が高まるように Oracle Application Server を構成するには、Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用して Web リスナーをグループに追加します。指定するグループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

「リソースをグループに追加」ウィザードの最初のページには、これから Oracle Application Server をグループに追加することが示されますが、このウィザードでは実際には、関連する Oracle Application Server Web リスナーの詳細を入力するように要求されます。Oracle

Application Server Web リスナーをグループに追加すると、Oracle Fail Safe により、関連する Oracle Application Server も同様に追加されます。Oracle Application Server Web リスナーをグループに追加すると、サーバーの可用性が高まります。

Oracle Fail Safe を使用すると、Oracle Application Server Web リスナーを新規に作成したり、新規または既存の Oracle Application Server Web リスナーをグループに追加することができます。通常、このグループには次のリソースが含まれます。

- 仮想アドレス（IP アドレスとネットワーク名）
- Oracle Application Web リスナーおよび対応する Oracle Application Server プロセス
- Oracle Application Web リスナーのルート・ディレクトリおよび物理ディレクトリに指定したすべてのディスク

Oracle Application Server Web リスナーをグループに追加するクラスタワイド操作によっても、Oracle Application Server リソースが（可能所有者ノードである）すべてのクラスタ・ノード上で起動されます（まだ起動されていない場合）。この操作は、最長 5 分間かかりますが、その間、Oracle Application Server または Oracle Fail Safe Server（あるいはその両方）は、次のようにオンライン保留状態になることがあります。

- 「リソースをグループに追加」クラスタワイド操作中、Oracle Application Server プロセスは、最長 5 分間オンライン保留状態になることがあります。
- Oracle Application Server が Oracle Fail Safe を使用して構成されている場合、クラスタ・ノードをリブートした後、Oracle Fail Safe Server リソースは最長 5 分間（MSCS クラスタ アドミニストレータ内で）オンライン保留状態になることがあります。

アーキテクチャおよび概念に関する詳細は、Oracle Application Server のマニュアルを参照してください。

### 12.3.1 構成前に

Oracle Application Server をグループに追加する前に、次のことに注意してください。

- Oracle Application Server 実行可能ファイルは、各クラスタ・ノードのプライベート・ディスク上の Oracle ホームにインストールします。
- その他の必要な Oracle 製品の実行可能ファイルはすべて、各クラスタ・ノードのプライベート・ディスク上の Oracle ホームにインストールします。

製品のマニュアルを参照して、同じ Oracle ホームにインストールできる製品とできない製品を調べてください。

- Oracle Application Server データ・ファイルはすべて、どのクラスタ・ノードが Oracle Application Server のホストになっているかに関係なくアクセスできるようにクラスタ・ディスク上に配置するか、または、ローカル・ディスク上の同じ場所に複製します。
- Oracle Application Server Web リスナーをグループに追加（または作成して追加）する場合は、そのプロパティ（ルート・ディレクトリ、イニシャル・ファイルおよび物理

ディレクトリ) がすべて指定されている必要があります。詳細は、[12.3.3 項](#)を参照してください。

- Oracle Application Server Web リスナーをグループに追加（または作成して追加）する場合、仮想アドレスおよびポートの組合せは、別コンポーネント（別の Web サーバーなど）で使用されていないものにします。Oracle Application Server をインストールしたときに作成した Web リスナーで使用されている組合せも使用できません。詳細は、[12.7.2 項](#)を参照してください。
- Oracle Application Server をインストールすると、www、admin および node の 3 つの Web リスナーがデフォルトで作成されます。詳細は、[12.7.2 項](#)を参照してください。
- グループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。

## 12.3.2 Oracle Application Server の構成手順

Oracle Fail Safe を使用して Oracle Application Server の可用性が高まるように必要な手順は、Oracle Fail Safe Manager によって自動化されます。Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードで、リソース・タイプに Oracle Application Server を指定してください。

[表 12-1](#) に、可用性が高まるように Oracle Application Server を構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプとチュートリアルを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択するか、または「ヘルプ」→「チュートリアル」を選択します。

**表 12-1 Oracle Application Server の構成手順**

手順	処置	説明
1	Oracle Application Server の可能所有者ノードとなる各クラスター・ノードのプライベート・ディスク上に、Oracle Application Server がインストールされていることを確認	インストールの詳細情報は、Oracle Application Server のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に Oracle Application Server コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle Application Server コンポーネントを選択します。これで、DLL ファイルと NT レジストリ変数がインストールされます。詳しい指示は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
3	Web ページ・ファイル、Web アプリケーション・ファイルおよび Java ファイルのクラスター・ディスクへのコピー	Web アプリケーションのデータ・ファイルと実行可能ファイル（たとえば、HTML、CGI、ASP、Visual Basic、Java、.jar など）は、共有記憶装置インターコネクトのクラスター・ディスク上に配置する必要があります。 <sup>1</sup>

表 12-1 Oracle Application Server の構成手順（続き）

手順	処置	説明
4	各クラスタ・ノード上の Web アプリケーションをスタンドアロン・モードでテスト	Oracle Application Server は、グループに追加する前に、ノードの 1 つでスタンドアロン・モードでテストし、正常に機能することを確認します。
5	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」
6	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロセスを実行します。
7	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）
8	Web サーバーのグループへの追加	「リソース」→「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い Oracle Application Server を構成する際に役立ちます。Oracle Application Server Web リスナーを新規に作成することも、既存の Oracle Application Server Web リスナーを使用することもできます。
9	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
10	その他のアプリケーション・コンポーネントをフェイルオーバー用に構成	その他のアプリケーション・サーバーおよびデータベース・サーバーを Oracle Fail Safe を使用して構成し、Web ベースのアプリケーション全体の可用性が高くなるようにします。

<sup>1</sup> Oracle Application Server の可能所有者ノードであるすべてのノード上でこれらのファイルのパスが同一であり、すべてのクラスタ・ノード上でファイルがまったく同一に維持される場合は、プライベート・ディスク上にこれらのファイルを配置できます（ただし、これは推奨できません）。

12.3.3 Oracle Application Server の構成データ

Oracle Fail Safe Manager には、可用性が向上するように Oracle Application Server を構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- Oracle Application Server Web リスナーの可能所有者ノード（クラスタ内に 3 つ以上のノードが存在する場合）
- Oracle Application Server Web リスナー名
- Oracle Application Server Web リスナーのイニシャル・ファイル、ルート・ディレクトリおよび仮想ディレクトリ

- Web ブラウザで Web リスナーへの HTTP アクセスに使用される仮想アドレスおよびポート

Oracle Application Server をグループに追加するクラスタワイド操作は、次の場合に失敗します。

- Oracle Application Server Web リスナーのルート・ディレクトリが存在しない場合
- Oracle Application Server Web リスナーのイニシャル・ファイルがルート・ディレクトリにない場合
- 同じ名前を持つ別の Oracle Application Server Web リスナーがすでに存在する場合
- 別のリスナー（Oracle Application Server Web リスナー、Oracle HTTP Server、Oracle TNS リスナーなど）が同一の IP アドレスとポートの組合せをリスニングしている場合

Oracle Application Server のグループへの追加またはグループからの削除でエラーが発生した場合は、Windows のイベント ビューア（アプリケーション ログ）を検査し、Oracle Fail Safe または Oracle Web Lsnr からのエラー・メッセージがないかどうかを調べます。

次の各項では、Oracle Application Server をグループに追加するためのデータ要件を詳細に説明します。

### 12.3.3.1 ノードの選択

Oracle Application Server Web リスナーをグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、図 12-3 に示すように、選択済ノードのリストを指定して、Oracle Application Server Web リスナーの可能所有者となるノードを指定するように求められます。特定のノードを Oracle Application Server Web リスナーの可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

2.6.6 項では、「可能所有者ノード」リストの概念について詳細に説明します。

図 12-3 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle Application Server Web リスナーをグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを Oracle Application Server Web リスナーの可能所有者とするのかを指定するように求められます。このような場合、図 12-4 に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 12-4 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



### 12.3.3.2 Oracle Application Server Web リスナーの識別情報

図 12-5 に示すように、「リソースをグループに追加」ウィザードでは、Oracle Application Server Web リスナーの識別情報に関する次の情報が要求されます。

- ノード名

(既存の Oracle Application Server をグループに追加しようとする場合) これは、Web リスナーが現在実行されているノードの名前にします。Oracle Application Server を新規作成してグループに追加する場合は、Oracle Application Server の可能所有者であるどのノードを指定しても構いません。

- Oracle ホーム

これは、Oracle Application Server 実行可能ファイルがインストールされているホームです。

- Web リスナー名

これは、Oracle Application Server Web リスナーの名前です。既存の Oracle Application Server Web リスナーの場合は、ドロップダウン・リストから選択できます。Oracle Fail Safe により Oracle Application Server Web リスナーを作成する場合は、クラスタで Oracle Application Server Web リスナーを一意に識別する名前を指定します。この名前は、英数字 6 文字以下の文字列にしてください。

図 12-5 「OAS Web リスナーの識別情報」ウィザード・ページ



リソースをグループに追加 - OAS Webリスナーの識別情報 ステップ2/4

グループ名 Forms\_Group

このグループに追加したい OAS Webリスナーはどれですか?  
既存の OAS Webリスナーの名前から選択できます。または、一意な  
名前を入力することにより Oracle Fail Safe で新しい OAS  
Webリスナーを作成できます。

ノード名 NTCLU4-1

Oracleホーム OAS

Webリスナー名 MyWeb

< 戻る(B) 次へ(N) > キャンセル ヘルプ

### 12.3.3.3 Oracle Application Server Web リスナーのデフォルト・ファイル

図 12-6 に示すように、「リソースをグループに追加」ウィザードでは、ルート・ディレクトリ、イニシャル・ファイルおよび仮想ディレクトリに関する次の情報を指定するように要求されます。関連する Oracle Application Server をグループに追加するには、イニシャル・ファイルおよびディレクトリがすでに存在する必要があります。



- ルート・ディレクトリ

これは、Web リスナーの仮想ファイル・システムのルートとして機能するクラスタまたはローカル・ファイル・システム内にあり、

URL `http://virtual-address:port-number/` により参照されるディレクトリです。この値は、Oracle Application Server Manager でサーバー・フォームに指定するルート・ディレクトリ値に対応しています。仮想アドレスは、Web リスナーが接続要求を受け入れるアドレスです。詳細は、[12.3.3.4 項](#)で説明しています。

- イニシャル・ファイル

これは、URL がルート・ディレクトリのみを指定している場合に、Oracle Application Server によりルート・ディレクトリから取り出されるデフォルト・ファイルです。たとえば、`index.htm` というイニシャル・ファイルを指定した場合、ユーザーが要求

URL `http://virtual-address:port-number` を入力すると、Oracle Application Server により `http://virtual-address:port-number/index.htm` が返されます。この値は、Oracle Application Server Manager でサーバー・フォームに指定するイニシャル・ファイル値に対応しています。

- 仮想ディレクトリ

仮想ディレクトリを指定すると、Web サイトにアクセスするユーザーから、Web ページのファイル構造と物理位置を隠すことができます。物理ディレクトリが変更された場合でも、ユーザーが指定する URL を変更する必要はありません。「リソースをグループに追加」ウィザードのこの部分に指定する値は、Oracle Application Server Manager のディレクトリ・フォームに対応しています。

Oracle Application Server Web リスナーに対して仮想ディレクトリ・マッピングを指定すると、Oracle Fail Safe により、デフォルトで CGI 実行許可がコード N および R に設定されます。

- 最初のコード N（通常）を指定すると、CGI プログラムはこの仮想ディレクトリでは実行できません。
- 2 番目のコード R（再帰）を指定すると、CGI プログラムはこの仮想ディレクトリのサブディレクトリでは実行できません。

必要な場合は、Oracle Application Server Manager を使用して各仮想ディレクトリの CGI 実行コードを変更します。

**注意：** Oracle Fail Safe Manager では、Oracle Fail Safe Manager からの Oracle Application Server Web リスナーのプロパティの変更はサポートしていません。Oracle Application Server Manager を使用して、各ノード上の Oracle Application Server Web リスナーのプロパティを設定することはできます。新規設定を有効にするには、リスナーを再起動する必要があります。

図 12-6 「OAS Web リスナーのデフォルト・ファイル」ウィザード・ページ

リソースをグループに追加 - OAS Webリスナーのデフォルト・ファイル ステップ 3/4

Webリスナーのルート・ディレクトリはどこですか?

ルート・ディレクトリ

Webリスナーが使うイニシャル・ファイルは何ですか? このファイルはルート・ディレクトリに置く必要があります。

イニシャル・ファイル

Webリスナーで使う仮想ディレクトリは何ですか?

仮想ディレクトリ	物理ディレクトリ
----------	----------

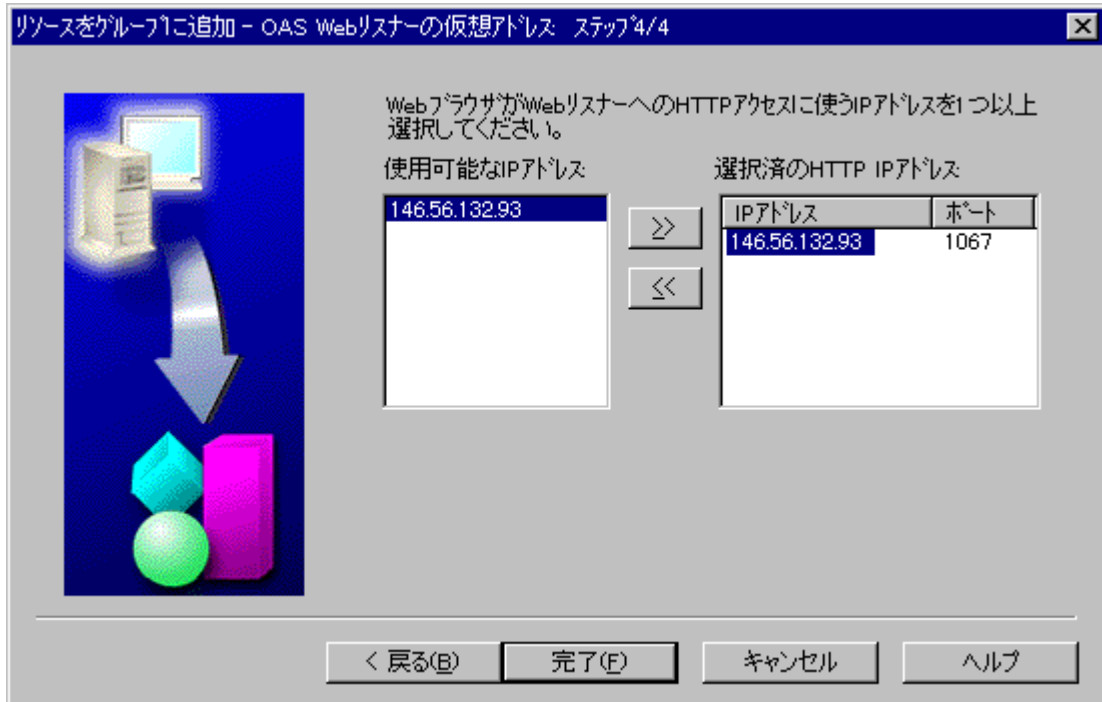
#### 12.3.3.4 Oracle Application Server Web リスナーの仮想アドレス

図 12-7 に示すように、「リソースをグループに追加」ウィザードでは、Web リスナーが要求を受け入れるための 1 つ以上の IP アドレスおよびポートを指定するように要求されます。表示される IP アドレスは、Oracle Application Server Web リスナーを追加するグループに含まれている仮想アドレスです。このリスナーに対する IP アドレスとポート番号の組合せは、クラスタ内で固有になるようにしてください。

Oracle Application Server の一部としてデフォルトのリスナーがインストールされていることに注意してください。デフォルトでは、これらのリスナーは、インストールされているシ

システム上のすべての IP アドレスをリスニングします。Oracle Fail Safe を使用して作成した Oracle Application Server Web リスナーの場合、ノードの IP アドレスのみをリスニングするようにデフォルト・リスナーを再構成しない限り、これらの Web リスナーに指定したポート番号は使用できません。詳細は、[12.7.2 項](#)を参照してください。

図 12-7 「OAS Web リスナーの仮想アドレス」ウィザード・ページ



## 12.4 Oracle Application Server Web リスナーへのクライアント接続

クライアントは、次のように、Web ブラウザでリスナーの仮想アドレスおよびポート番号を URL として指定し、Oracle Application Server Web リスナー（およびリスナーでリスニングする Web サイト）に接続します。

`http://<virtual-address>:<port-number>`

<virtual-address> は Oracle Application Server Web リスナーの仮想アドレス、  
<port-number> は Oracle Application Server Web リスナーのポート番号です。

たとえば、可用性の高い Oracle Application Server Web リスナーの仮想アドレスが ntclu4-6、ポート番号が 1067 の場合、Oracle Application Server Web リスナーに接続するには Web ブラウザで次のように指定します。

`http://ntclu4-6:1067`

## 12.5 Oracle Application Server Web リスナーのグループからの削除

Oracle Application Server Web リスナーをグループから削除するには、削除するリスナーを Oracle Fail Safe Manager のツリー・ビューから選択し、「リソース」メニューで「グループから削除」をクリックします。

Oracle Application Server Web リスナーをグループから削除するには、特別な注意が必要です。Oracle Application Server Web リスナーを削除すると、削除したリスナーはスタンドアロン・リソースになります。スタンドアロン Oracle Application Server Web リスナーは、削除前に属していたグループと同じノード上に配置されます。対応する Oracle Application Server プロセスは、クラスタから削除されます。

ただし、Oracle Application Server Web リスナーの仮想ディレクトリ指定では、まだクラスタ・ディスクを指しています。そのため、(グループから削除した) スタンドアロン Oracle Application Server Web リスナーを起動する前に、次のことを実行してください。

1. 仮想ディレクトリが指すファイルおよびディレクトリを、クラスタ・ディスクからプライベート・ディスクへコピーします。
2. Oracle Application Server Manager を使用し、最初の手順でファイルおよびディレクトリを移動したプライベート・ディスクを指すように、仮想ディレクトリ指定を変更します。

Oracle Web NodeManager4.0 (OAS、website40) または Oracle Web Lsnr 4.0 (admin-OAS、website40) をグループから削除しようとし、サービスを削除するかどうかの問いに「はい」を選択すると、Oracle Fail Safe では、特別な Oracle Application Server Web リスナーは削除できないという警告を返します。これらのリスナーは、Oracle Application Server インストールの一部であり、Oracle Application Server で必須です。

## 12.6 Oracle Application Server のセキュリティ要件

Oracle Application Server を可用性が高まるように構成するには、Oracle Fail Safe の Oracle Fail Safe Server と同じ Windows システム・アカウントでインストールする必要があります。

Oracle のリソースおよびアプリケーションを管理し、Oracle Fail Safe Manager を介して操作を実行するための適切な権限については、[4.3 項](#)を参照してください。

## 12.7 Oracle Application Server のインストールに関するヒント

次の各項では、Oracle Application Server のインストールに関するヒントを記載します。

### 12.7.1 インストールの一般的なヒント

Oracle Application Server は、各クラスタ・ノード上のプライベート・ディスクにインストールし、インストール・オプション（インストール・タイプや、Oracle ホームのディスクおよびディレクトリの場所、Web サイト名、Web リスナーのデフォルトの名前およびポートなどのオプション）はすべて同じものを指定する必要があります。

Oracle Fail Safe では、Oracle Application Server の次のインストール・タイプをクラスタ・ノード上でサポートします。

- 標準
- カスタム
  - 単一ノードとしてのインストール
  - マルチノード・インストールのプライマリ・ノードとしてインストール

Oracle Application Server のインストール後、インストールした Oracle ホームが、Oracle Application Server の可能所有者ノードであるすべてのクラスタ・ノード上の主 Oracle ホームに設定されていることを確認してください。これは、Oracle Fail Safe を使用して Oracle Application Server の可用性が高まるように構成する前に実行してください。Oracle Home Selector を使用して、Oracle Application Server をインストールした Oracle ホームを主 Oracle ホームに設定します。これは、Oracle 製品を追加インストールするたびに実行してください。

### 12.7.2 インストールされる Oracle Application Server Web リスナー

Oracle Application Server をインストールすると、次の 3 つの Web リスナーが作成されます。

- www

これは、デフォルトの Oracle Application Server Web リスナーです。デフォルトでは、このリスナーが、インストールされているシステムの全 IP アドレス上のポート 80 をリスニングします。

この Web リスナーの名前は、Oracle Application Server をインストールする際に変更できますが、デフォルトでは www に設定されます。
- admin

これは、Oracle Application Server の管理 Web リスナーです。デフォルトでは、このリスナーが、インストールされているシステムの全 IP アドレス上のポート 8889 をリスニングします。

この Web リスナーの可用性を高めないようにしてください。可用性が高まるように Web リスナーを構成すると、そのリスナーは仮想アドレスをリスニングすることになります。ただし、Oracle Application Server 管理 Web リスナーは、ノードの IP アドレスをリスニングするように設計されています。可用性が高まるように Web リスナーを構成した場合、そのリンクの一部が動作しなくなります。

■ node

これは、Oracle Application Server のノード・マネージャ Web リスナーです。デフォルトでは、このリスナーが、インストールされているシステムの全 IP アドレス上のポート 8888 をリスニングします。admin Web リスナーの可用性を高めない方がよいのと同じ理由で、この Web リスナーの可用性も高めないでください。

Oracle Fail Safe を使用して作成した Oracle Application Server Web リスナーの場合、ノードの IP アドレスのみをリスニングするようにデフォルト・リスナーを再構成しない限り、これらのデフォルト Web リスナーで使用されるポート番号を指定しないでください。指定した場合、「リソースをグループに追加」操作中に、ポートがすでに使用されていることを示すエラーが返されます。

デフォルト・リスナーのうちの 1 つですでに使用されているポート番号を使用する場合は、次の 2 つの項で説明しているように、デフォルト・リスナーを永久に停止するか、またはノードの IP アドレスのみをリスニングするようにデフォルト・リスナーを再構成します。

Oracle Fail Safe Manager を使用して、Oracle Application Server インストール中に作成された Oracle Application Server Web リスナーを削除することはできません。これらのリスナーは、Oracle Application Server および Oracle Fail Safe Server で必須です。

### 12.7.2.1 ノードの IP アドレスのみをリスニングするようにデフォルト・リスナーを再構成

ノードの IP アドレスのみをリスニングするようにデフォルト・リスナーを再構成するには、次の手順に従います。

1. Oracle Application Server Manager を開きます。
2. ツリー・ビューから Web リスナーを選択します。
3. ツリー・ビューを拡張し、「ネットワーク」を選択します。
4. 「ネットワーク」フォームで、「アドレス」フィールドの「ANY」をノードの IP アドレスに変更します。
5. 「適用」をクリックします。
6. Oracle Application Server Manager を表示している Web ブラウザで、「再読み込み」または「更新」をクリックします。
7. ツリー・ビューから再び Web リスナーを選択します。
8. 右側フレーム内の「ページの更新」アイコン（緑色の縦棒および矢印）をクリックして、リスナーを停止してから再起動し、変更を有効にします。

### 12.7.2.2 デフォルト Web リスナーの停止

デフォルト Web リスナーを停止するには、次の手順に従います。

1. Oracle Application Server Manager を開きます。
2. ツリー・ビューから Web リスナーを選択します。
3. 右側フレーム内の「停止」アイコン（赤い正方形）をクリックします。

## 12.8 可用性の高い Oracle Application Server Web リスナーに関連する問題のトラブルシューティング

Oracle Fail Safe のトラブルシューティング・ツール（「クラスタの検証」および「グループの検証」）の使用に関する一般情報は、[第 6 章](#)に記載しています。Oracle Application Server のトラブルシューティングに関する一般的な情報は、Oracle Application Server のマニュアルを参照してください。

Oracle Application Server Web リスナーに問題が発生した場合は、「クラスタの検証」操作を実行して検証し、問題があれば修正します。たとえば、「クラスタの検証」操作では、Oracle Application Server がインストールされている Oracle ホームが、それぞれのノードで主 Oracle ホームかどうかを調べます。Oracle Application Server のユーティリティを正しく実行するには、主 Oracle ホームである必要があります。Oracle Application Server がインストールされている Oracle ホームを、ノード上の主 Oracle ホームにするには、Oracle Home Selector を使用できます。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合

## 12.9 Oracle HTTP Server の概要

Oracle Fail Safe では、Oracle HTTP Server（powered by Apache）をサポートします。Oracle HTTP Server は、Oracle Internet Application Server（Oracle iAS）のコンポーネントです。

次の項では、Oracle Fail Safe を使用して Oracle HTTP Server の可用性が高まるように構成する方法を説明します。

## 12.10 スタンドアロン Oracle HTTP Server の検出

Oracle Fail Safe Manager を使用して、可用性が向上するように Oracle HTTP Server を構成します。Oracle Fail Safe Manager でスタンドアロン Oracle HTTP Server を表示するには、まずスタンドアロン Oracle HTTP Server を検出する必要があります。

Oracle Fail Safe Server では、Oracle HTTP Server を検出する際、Windows サービス マネージャ内のリストを調べて Oracle HTTP Server エントリを見つけ、可用性が高まるように構成されているかどうかを判断します。

Oracle Fail Safe では、新しく検出された Oracle HTTP Server が Oracle Fail Safe Manager のツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

## 12.11 Oracle HTTP Server のグループへの追加

可用性が高まるように Oracle HTTP Server を構成するには、少なくとも 1 つの仮想アドレスが含まれたグループに Oracle HTTP Server を追加します。Oracle Fail Safe Manager には、既存の Oracle HTTP Server のグループへの追加や、Oracle HTTP Server の新規作成とグループへの追加が可能なウィザードがあります。Oracle Fail Safe により、Oracle HTTP Server が依存する他のすべてのリソースが追加されます。通常、このグループには次のリソースが含まれます。

- 仮想アドレス (IP アドレスとネットワーク名)
- Oracle HTTP Server 名
- Oracle HTTP Server で使用する全ディスク

### 12.11.1 構成前に

Oracle HTTP Server をグループに追加する前に、次のことに注意してください。

- Oracle HTTP Server 実行可能ファイルは、Oracle HTTP Server の可能所有者ノードである各クラスタ・ノードのプライベート・ディスク上にある同じ Oracle ホーム (同じパス) にインストールします。
- その他の必要な Oracle 製品の実行可能ファイルはすべて、各クラスタ・ノードのプライベート・ディスク上の Oracle ホームにインストールします。

製品のマニュアルを参照して、同じ Oracle ホームにインストールできる製品とできない製品を調べてください。



## 12.11.2 Oracle HTTP Server の構成手順

Oracle HTTP Server を作成して可用性の高い構成にする際に必要な手順は、Oracle Fail Safe Manager によって自動化されます。Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードで、リソース・タイプに Oracle HTTP Server を指定してください。

表 12-2 に、可用性が高まるように Oracle HTTP Server を構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプとチュートリアルを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択するか、または「ヘルプ」→「チュートリアル」を選択します。

表 12-2 Oracle HTTP Server の構成手順

手順	処置	説明
1	可能所有者ノードとなる各クラスタ・ノードのプライベート・システム・ディスク上に、Oracle HTTP Server ソフトウェアがインストールされていることを確認	インストールの詳細情報は、Oracle HTTP Server のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に Oracle HTTP Server コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle HTTP Server コンポーネントを選択します。これで、DLL ファイルと NT レジストリ変数がインストールされます。詳しい指示は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
3	Web ページ・ファイル、Web アプリケーション・ファイルおよび Java ファイルのクラスタ・ディスクへのコピー	Web アプリケーションのデータ・ファイルと実行可能ファイル（たとえば、HTML、CGI、ASP、Visual Basic、Java、.jar など）は、共有記憶装置インターコネクトのクラスタ・ディスク上に配置する必要があります。 <sup>1</sup>
4	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」
5	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
6	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）

表 12-2 Oracle HTTP Server の構成手順 (続き)

手順	処置	説明
7	Web サーバーのグループへの追加	Oracle Fail Safe Manager で、「リソース」→「グループに追加」を選択し、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い Oracle HTTP Server を構成する際に役立ちます。新規の Oracle HTTP Server を作成することも、既存の Oracle HTTP Server を使用することも可能です。
8	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
9	その他のアプリケーション・コンポーネントをフェイルオーバー用に構成	その他のアプリケーション・サーバーおよびデータベース・サーバーを Oracle Fail Safe を使用して構成し、Web ベースのアプリケーション全体の可用性が高くなるようにします。

<sup>1</sup> Oracle HTTP Server の可能所有者ノードであるすべてのノード上でこれらのファイルのパスが同一であり、すべてのクラスター・ノード上でファイルがまったく同一に維持される場合は、プライベート・ディスク上にこれらのファイルを配置できません（ただし、これは推奨できません）。

12.11.3 Oracle HTTP Server の構成データ

Oracle Fail Safe Manager には、可用性が向上するように Oracle HTTP Server を構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- Oracle HTTP Server の識別情報（名前、バージョン、Oracle ホーム、Oracle Fail Safe で構成ファイルをコピーするクラスタ・ディスク）。
- Oracle HTTP Server のディレクトリ索引ファイルおよび仮想ディレクトリ。
- Web ブラウザで Oracle HTTP Server への HTTP アクセスに使用する 1 つ以上の仮想アドレスとポート。Web ブラウザで Oracle HTTP Server に SSL（セキュア）アクセスする場合は、オプションで、SSL アクセスに使用する 1 つ以上の仮想アドレスとポートが必要です。

次の項で、これらの要件を詳細に説明します。

12.11.3.1 ノードの選択

Oracle HTTP Server をグループに追加する際に、クラスタが 3 つ以上のノードから構成されている場合は、図 12-8 に示すように、選択済ノードのリストを指定して、Oracle HTTP Server の可能所有者となるノードを指定するように求められます。特定のノードを Oracle HTTP Server の可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

2.6.6 項では、「可能所有者ノード」リストの概念について詳細に説明します。

図 12-8 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle HTTP Server をグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを Oracle HTTP Server の可能所有者とするのかを指定するように求められます。このような場合、[図 12-9](#) に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 12-9 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



### 12.11.3.2 Oracle HTTP Server の識別情報

図 12-10 に示すように、「リソースをグループに追加」ウィザードでは、可用性が高まるように構成する Oracle HTTP Server を作成して一意に識別するための情報を入力するように要求されます。Oracle Fail Safe は、このデータを使用して Oracle HTTP Server をクラスタ内に構成します。また、ユーザーが入力したデータが MSCS に渡されます。データは MSCS に登録されて、Oracle HTTP Server がオンラインまたはオフラインになるとき、あるいは Is Alive ポーリングが実行されるときに使用されます。Oracle Fail Safe では、次の情報が要求されます。

- Oracle HTTP Server 名

Oracle HTTP Server を一意に識別する名前。

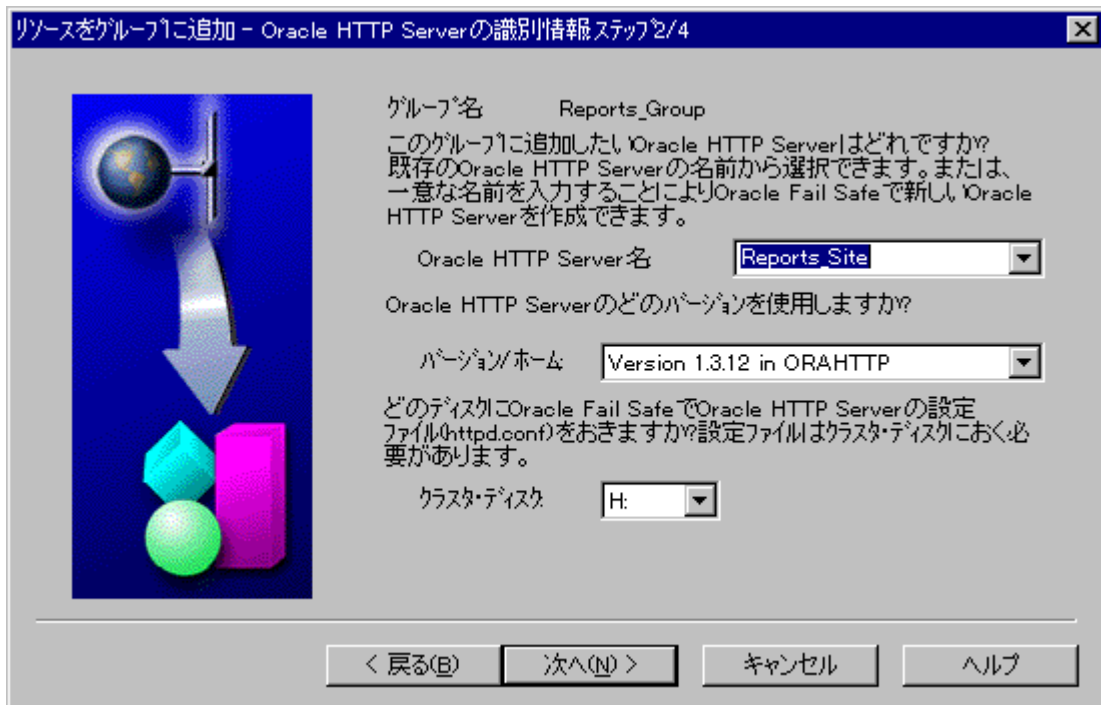
- Oracle HTTP Server の Oracle ホーム

Oracle HTTP Server がインストールされた Oracle ホームの名前。Oracle ホームの名前およびパスは、Oracle HTTP Server の可能所有者である各クラスタ・ノードで同一である必要があります。

- クラスタ・ディスク

Oracle Fail Safe で構成ファイルをコピーする先のクラスタ・ディスク。クラスタ・ディスクおよび Oracle HTTP Server 名は、ドキュメント・ルート・ディレクトリの作成にも使用されます。たとえば、Oracle HTTP Server 名として Oracle\_HTTP\_Test を指定し、クラスタ・ディスクとして U:¥ を選択すると、Oracle Fail Safe では構成ファイルを U:¥Apache¥conf¥httpd\_oracle\_http\_test.conf にコピーし、ドキュメント・ルート・ディレクトリは U:¥Apache¥htdocs となります。

図 12-10 「Oracle HTTP Server の識別情報」ウィザード・ページ



### 12.11.3.3 Oracle HTTP Server のディレクトリ

図 12-11 に示すように、「リソースをグループに追加」ウィザードでは、Oracle HTTP Server のディレクトリ索引ファイルおよび仮想ディレクトリに関する情報を入力するように要求されます。

- ディレクトリ索引ファイル

ディレクトリ索引ファイルは、ユーザーが URL を指定したが特定の Web ページ (<http://www.oracle.com> など) を入力していない場合に、表示される Web ページ (.html、.asp など) を特定します。ディレクトリ索引のリストを指定できます。ディレクトリ索引の最初のファイルが使用できない場合、ディレクトリ索引の 2 番目のファイルが表示されます。ディレクトリ索引の最初と 2 番目のファイルも使用できない場合には、ディレクトリ索引の 3 番目のファイルが表示されます。ウィザード・ページの上部にある「追加」をクリックして、ディレクトリ索引ファイルを指定します。

- 仮想ディレクトリ

仮想ディレクトリを指定すると、Web サイトにアクセスするユーザーから、Web ページの物理位置を隠すことができます。これは、仮想ディレクトリが、Web ページ・ファイルを含む物理ディレクトリのシノニムであるためです。たとえば、/Web\_Code という仮想ディレクトリは、Java アプレットの実行に必要な Java コードを含む物理ディレクトリ D:\MyFolder\FForms60\Java にマップされます。仮想ディレクトリ名には、必ず先頭にスラッシュ (/) を付けてください。

ウィザード・ページの下部にある「追加」をクリックして、仮想ディレクトリを指定します。「追加」をクリックすると、Oracle Fail Safe では小さなダイアログ・ボックスが開きます。このダイアログ・ボックスで、仮想ディレクトリと物理ディレクトリの組合せを同時に指定できます。また、指定ディレクトリからの CGI プログラムの実行を指定できます。

「CGI」オプションをチェックすると、Oracle Fail Safe では仮想ディレクトリでの CGI 実行を可能にします。cgi、.pl および .exe ファイルは、Oracle HTTP Server で CGI スクリプトとして識別されます。

図 12-11 「Oracle HTTP Server のディレクトリ」ウィザード・ページ



### 12.11.3.4 Oracle HTTP Server の仮想アドレス

図 12-12 に示すように、「リソースをグループに追加」ウィザードでは、Web サーバーで受信要求のリスニングに使用される 1 つ以上の IP アドレスを選択するように要求されます。

HTTP（一般）アクセスに使用する IP アドレス、および SSL（セキュア）アクセスに使用する IP アドレスを指定できます。HTTP アクセス用のアドレスを少なくとも 1 つ指定する必要があります。

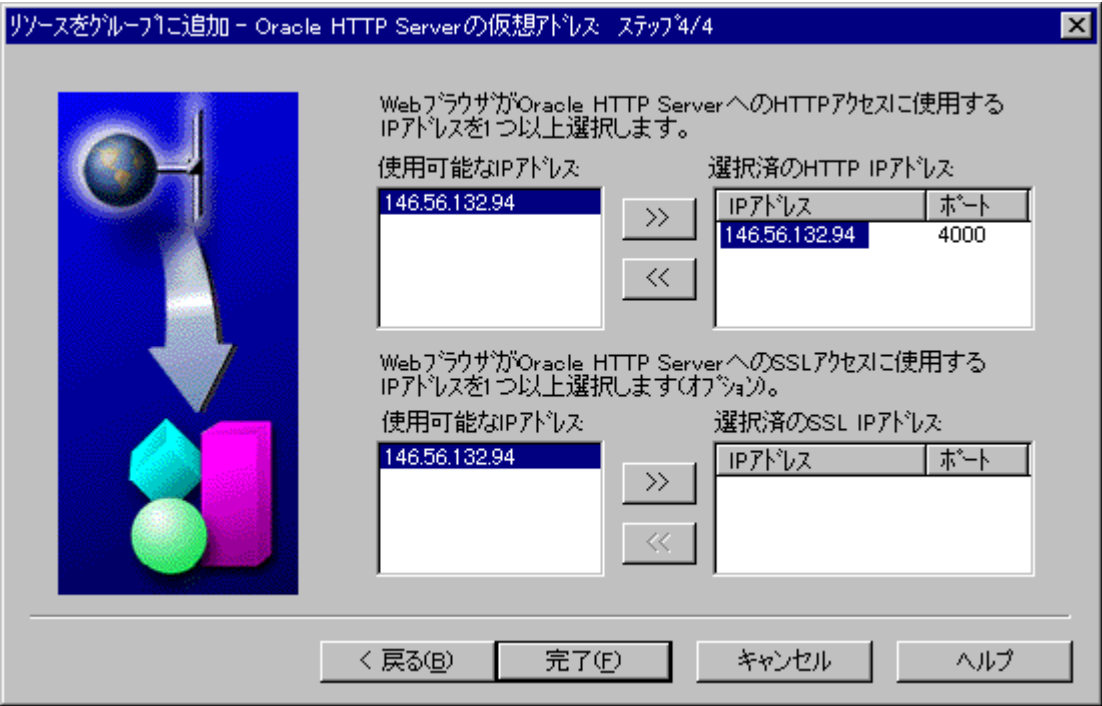
ウィザードに表示される IP アドレスは、仮想アドレスです。アドレスを「使用可能な IP アドレス」リストから「選択済の HTTP IP アドレス」リストあるいは「選択済の SSL IP アドレス」リストへ移動すると、Oracle Fail Safe ではウィンドウが開き、Web サーバーで選択済アドレスをリスニングするポートを指定するように要求されます。

（「リソースをグループに追加」ウィザードを使用して）Oracle HTTP Server への SSL アクセスに使用する IP アドレスを指定した場合、Oracle Fail Safe は SSL をリスニングするように構成ファイルを変更します。ただし、Load\_module.SSL\_module が <IfDefineSSL> のインスタンスの内部にある場合は、仮想ホスト情報も <IfDefineSSL> のインスタンスの内部に入ります。たとえば、次のようになります。

```
<IfDefine SSL>
<VirtualHost 138.2.26.178:180>
SSLEngine on
SSLCertificateFile %conf%ssl.crt%server.crt
SSLCertificateKeyFile conf%ssl.key%server.key
</VirtualHost>
</IfDefine>
```

この場合に SSL サポートをアクティブにするには、<IfDefineSSL> のすべてのインスタンスを構成ファイルから削除する必要があります。

図 12-12 「Oracle HTTP Server の仮想アドレス」 ウィザード・ページ





## 12.12 Oracle HTTP Server へのクライアント接続

クライアントは、次のように、Web ブラウザでサーバーの仮想アドレスおよびポート番号を URL として指定し、Oracle HTTP Server（およびサーバーで処理する Web サイト）に接続します。

```
http://<virtual-address>:<port-number>
```

<virtual-address> は Oracle HTTP Server の仮想アドレス、<port-number> は Oracle HTTP Server のポート番号です。

たとえば、可用性の高い Oracle HTTP Server の仮想アドレスが ntclu4-6、ポート番号が 9768 の場合、Oracle HTTP Server に接続するには Web ブラウザで次のように指定します。

```
http://ntclu4-6:9768
```

## 12.13 Oracle HTTP Server のグループからの削除

Oracle HTTP Server をグループから削除するには、削除するサーバーを Oracle Fail Safe Manager のツリー・ビューから選択し、「リソース」メニューで「グループから削除」をクリックします。

Oracle HTTP Server をグループから削除するには、特別な注意が必要です。Oracle HTTP Server をグループから削除すると、次のようなことが発生します。

1. フェイルセーフ Oracle HTTP Server は、スタンドアロン Oracle HTTP Server になります。
2. スタンドアロン Oracle HTTP Server は、削除前に属していたグループと同じノード上に配置されます。
3. Oracle Fail Safe では、httpd\_<resource\_name>.conf ファイルを、クラスタ・ディスクから、スタンドアロン Oracle HTTP Server と同じノード上の Oracle HTTP Server インストール・ディレクトリ（<Oracle\_Home>%Apache%Apache%conf）にコピーします。
4. Oracle Fail Safe では、スタンドアロン Oracle HTTP Server がリスニングするポート番号を指定するように要求します。
5. Oracle Fail Safe では、Oracle HTTP Server でその node:port のネットワーク・アドレスのみをリスニングするように構成ファイルを変更します。
6. SSL の仮想ホスト設定は変更されませんが、対応する Listen ディレクティブがないため使用されません。

仮想ディレクトリ指定は変更されません。この結果、当該クラスタ・ディスクをまだ指している仮想ディレクトリ指定が存在する場合があります。

そのため、(グループから削除した) スタンドアロン Oracle HTTP Server を起動する前に、次のことを実行してください。

1. 仮想ディレクトリが指すファイルを、クラスタ・ディスクからプライベート・ディスクへコピーします。
2. 仮想ディレクトリ・ポインタが、手順 1 でファイルを移動した先のプライベート・ディスクを指すように、`httpd_<resource_name>.conf` ファイルを変更します。

## 12.14 Oracle HTTP Server のセキュリティ要件

Oracle HTTP Server をクラスタ環境で実行する場合、特別な権限は不要です。

## 12.15 可用性の高い Oracle HTTP Server に関連する問題のトラブルシューティング

Oracle Fail Safe のトラブルシューティング・ツール (「クラスタの検証」および「グループの検証」) の使用に関する一般情報は、[第 6 章](#)に記載しています。Oracle HTTP Server のトラブルシューティングに関する一般的な情報は、Oracle HTTP Server のマニュアルを参照してください。

Oracle HTTP Server に問題が発生した場合は、「クラスタの検証」操作を実行してネットワークとノードの対称性の問題を検証し、問題があれば修正します。「グループの検証」操作を実行して、グループ構成を確認します。たとえば、「グループの検証」操作を実行すると、Oracle Fail Safe では、クラスタ・レジストリにある情報を Oracle HTTP Server 構成ファイル内に記述されている情報と比較します。情報が一致しない場合、Oracle Fail Safe により問題を修正するように求められます。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合
- グループのオンライン化に問題がある場合

次の各項では、可用性が高まるように構成した Oracle HTTP Server の使用時に発生する具体的な問題のトラブルシューティングについて説明します。

## 12.15.1 Oracle HTTP Server のグループへの追加の問題

「リソースをグループに追加」操作で、Oracle Fail Safe は、httpd.conf ファイルを現在の場所からクラスタ・ディスクへコピーします。「リソースをグループに追加」操作の実行中、ファイルの元の場所および Oracle Fail Safe でそのファイルをコピーする場所がウィンドウに表示されます。Oracle HTTP Server のグループへの追加に問題がある場合、次の処置を実行して問題を修正します。

1. 次のコマンドを発行して、元の httpd.conf ファイルの構文が正しいかどうかを確認します。

```
apache -t -f <config-file-spec>
```

この例では、config-file-spec は元の httpd.conf ファイルのファイル指定です。

2. コピーした httpd.conf ファイルが、クラスタ・ディスク上の次のディレクトリにあることを確認します。

```
<server-root>%conf%httpd_<resource-name>.conf
```

この例では、<server-root> は、「リソースをグループに追加」ウィザードでサーバー・ルートに指定したディスクおよびディレクトリです。

3. 次のコマンドラインを使用して、Oracle HTTP Server の起動を試行します。

```
apache.exe -f <server-root>%conf%httpd_<resource-name>.conf
```

4. Oracle Fail Safe でコピーを作成する元の httpd.conf ファイルの名前を変更します。これにより、Oracle Fail Safe では Oracle HTTP Server の新規 httpd.conf ファイルを作成します。元の httpd ファイルにエラーが存在した場合でも、新規作成されたファイルにはエラーは存在しません。

## 12.15.2 ユーザーが Oracle HTTP Server にアクセスできない

ユーザーが Oracle HTTP Server にアクセスできない場合は、次の処置を実行して問題を検出します。

- 「グループの検証」操作を実行して、Oracle HTTP Server のクラスタ・リソース構成を検証します。[6.1.2 項](#)の「グループの検証」操作の説明を参照してください。
- 全システム上の仮想サーバーを ping して、IP アドレスとホスト名が正しく登録されていることを確認します。

### 12.15.3 ユーザーが Oracle HTTP Server Web サイトに接続できない

Web サイトに接続できない場合は、次の処置を実行して問題を検出します。

- 現在グループを所有しているノード上で Web サイトが起動されていることを確認します。
- Web ブラウザのプロキシ設定を確認します。Web ブラウザによりプロキシが使用されている場合は、プロキシ・サーバーが使用されないように、追加アドレス情報（たとえば oracle.com）を URL に追加する必要がある場合もあります。

### 12.15.4 ユーザーが Web サイト上のドキュメントにアクセスできない

Web サイト上のドキュメントにアクセスできない場合は、次の処置を実行します。

- 仮想ディレクトリが物理ディレクトリ指定にマッピングされていることを確認します。
- すべてのファイルとディレクトリがアクセス許可を提供していることを確認します。

---

## 可用性を高めるための Oracle Applications Concurrent Manager の構成

Oracle Fail Safe は、可用性の高い内部 Oracle Applications Concurrent Manager を構成するのに役立ちます。クラスタ・ノードがシャットダウンまたは障害状態に陥ると、内部 Oracle Applications Concurrent Manager を含むグループは自動的に別のクラスタ・ノードへとフェイルオーバーされます。

Oracle Applications データの対話のほとんどは、Applications のフォームを通して行われます。ただし、中には定期的に実行する必要がある報告プログラムもあります。これらのプログラムには計算が大量に含まれることがあるため、対話操作を妨げないように、同時処理サーバーで実行されるように設定できます。

同時処理サーバーで実行されるプロセスは、コンカレント・プログラムと呼ばれます。これは、ユーザーおよび他のユーザーが他のタスクで作業している間にバックグラウンドで実行されます。通常、これらのプログラムは、C で書かれている実行可能プログラムか、Oracle Reports を使用して書かれたレポートです。データベース表に要求を挿入する Applications のフォームを通して、コンカレント・プログラムを実行するための要求を発行します。監視プロセスにより表が読み取られると、同時処理サーバーで実行中のいくつかある Concurrent Manager（ワーカーとも呼ばれる）の 1 つに要求が割り当てられます。Concurrent Manager により、要求が処理され、ログおよび出力ファイルが生成されます。これらのファイルは、同時処理サーバーに格納されます。

**内部 Concurrent Manager** プロセスにより、データベース表への新規要求の監視、他の Concurrent Manager の管理と、要求処理のタイミングおよび担当する Concurrent Manager の割り当てが行われます。Oracle Fail Safe を使用して可用性が高まるように構成するのは、この内部 Concurrent Manager です。

この章では、次の項目について説明します。

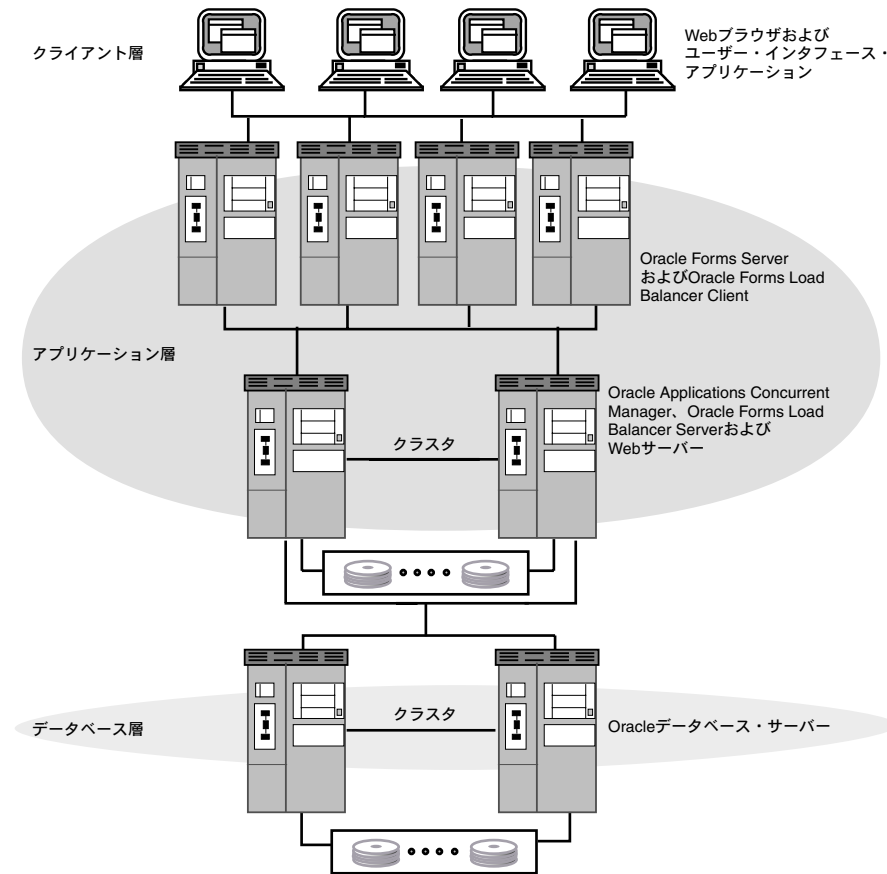
項目	参照
概要	13.1 項
スタンドアロン Oracle Applications Concurrent Manager の検出	13.2 項
Oracle Applications Concurrent Manager のグループへの追加	13.3 項
Oracle Applications Concurrent Manager のセキュリティ要件	13.4 項
可用性の高い Oracle Applications Concurrent Manager に関連する問題のトラブルシューティング	13.5 項

この章では、Oracle Fail Safe を使用して内部 Oracle Applications Concurrent Manger の可用性を高める方法について説明していますが、インターネット・ソリューション全体の可用性を高めるには、内部 Oracle Applications Concurrent Manger により使用されるデータベースの可用性を高めるとともに、任意のアプリケーション（Oracle Forms、Oracle Reports、Web サーバーなど）の可用性も向上させる必要があります。

## 13.1 概要

図 13-1 には、アプリケーション層で高い可用性のために構成されている Oracle Applications Concurrent Manager、Oracle Web サーバーおよび Oracle Forms Load Balancer Server、データベース層で構成されている Oracle データベースと、クライアント層にインストールされている Web ブラウザから成る 3 層アーキテクチャを持つ 2 ノード・クラスタが図示されています。クラスタ化されていないシステムでは、Oracle Forms Load Balancer Server に関連付けられた Oracle Forms Load Balancer Client および Oracle Forms Server がインストールされています。

図 13-1 可用性の高い Oracle Applications Concurrent Manager 構成のための  
3 層アーキテクチャ



## 13.2 スタンドアロン Oracle Applications Concurrent Manager の検出

Oracle Fail Safe Manager を使用して、可用性が高まるように内部 Oracle Applications Concurrent Manager を構成します。Oracle Fail Safe Manager インタフェースでスタンドアロン内部 Oracle Applications Concurrent Manager を表示するには、まずそれらを検出する必要があります。

Oracle Fail Safe Server は、内部 Oracle Applications Concurrent Manager を検出する際、Windows サービス マネージャ内のリストを調べて、対応する ccmsvc.exe のイメージ名エントリを見つけ、可用性が高まるように構成されていないものを判別します。Oracle Fail Safe では、新たに検出された内部 Oracle Applications Concurrent Manager が Oracle Fail Safe Manager のツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

## 13.3 Oracle Applications Concurrent Manager のグループへの追加

可用性が高まるように内部 Oracle Applications Concurrent Manager を構成するには、Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用してそれをグループに追加します。指定するグループには、少なくとも 1 つの仮想アドレスが含まれている必要があります。通常、このグループには次のリソースが含まれます。

- 仮想アドレス (IP アドレスとネットワーク名)。
- 内部 Oracle Applications Concurrent Manager。
- Oracle Applications Concurrent Manager により使用されるすべてのディスク。
- コンカレント・プログラムの実行のために内部 Oracle Applications Concurrent Manager が要求を読み取るデータベース (オプション)。

このデータベースがグループ内にある場合、それが使用するディスクと、データベースへの接続要求のためにグループの仮想アドレス (または複数) をリスニングする Net8 (または SQL\*Net) ネットワーク・リスナーも、グループに含まれます。

- Oracle Forms Server、Oracle Forms Load Balancer Server、Oracle Reports Server または Web サーバー (オプション)。

### 13.3.1 構成前に

内部 Oracle Applications Concurrent Manager をグループに追加する前に、次のことに注意してください。

- Oracle Applications 11i が Concurrent Manager の可能所有者ノードとなるクラスタの各ノードにインストールされていること。
- 内部 Oracle Applications Concurrent Manager が要求をフェッチするために接続するデータベースが存在すること。(このデータベースは、高いレベルの可用性を実現できるようにグループに追加する必要がありますが、これは必須事項ではありません。)



Oracle Applications Concurrent Manager が関連付けられているデータベースが、Concurrent Manager と同じグループに属する場合は、グループに Concurrent Manager を追加する前に、データベースをグループに追加します。これは、Concurrent Manager およびそれが関連するデータベースの間に適切な依存性を設定するために必要です。

- Concurrent Manager により使用される有効なアプリケーション構成 ID が、Concurrent Manager の可能所有者ノードとなるクラスタの各ノード上の Windows レジストリに存在する必要があります。アプリケーション構成 ID により、Concurrent Manager のランタイム環境が識別されます。アプリケーション構成 ID は次の Windows レジストリの場所にあります。ここで、<appl\_config> はアプリケーション構成 ID を示します。

```
HKEY_LOCAL_MACHINE\Software\Oracle\Applications\11.50\<appl_config>
```

通常、アプリケーション構成時の情報は、Oracle Applications 11i のインストール中にレジストリに追加されます。

13.3.2 構成手順

Oracle Fail Safe を使用して内部 Oracle Applications Concurrent Manager の可用性が高まるように必要な手順は、Oracle Fail Safe Manager によって自動化されます。Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードで、リソース・タイプに Oracle Applications Concurrent Manager を指定してください。

表 13-1 に、可用性が高まるように内部 Oracle Applications Concurrent Manager を構成する際に必要な作業の一覧を示します。特定の作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択します。

表 13-1 Oracle Application Concurrent Manager の構成手順

手順	処置	説明
1	各クラスタ・ノードのプライベート・ディスク上に Oracle Applications 11i がインストールされていることを確認	インストールの詳細情報は、Oracle Applications 11i のマニュアルを参照してください。
2	Oracle Fail Safe のインストール時に Oracle Concurrent Manager コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle Concurrent Manager コンポーネントを選択します。これで DLL ファイルをインストールします。詳しい指示は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
3	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「Oracle - <Oracle_Home>」→「Oracle Fail Safe Manager」

表 13-1 Oracle Application Concurrent Manager の構成手順（続き）

手順	処置	説明
4	クラスタの検証	「トラブルシューティング」→「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
5	グループの作成および 1 つ以上の仮想アドレスの追加	「グループ」→「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くので、ここで仮想アドレスをグループに追加します。（仮想アドレスをさらにグループに追加するには、「リソース」→「グループに追加」を選択します。）
6	Oracle Applications Concurrent Manager が関連付けられているデータベースが、Concurrent Manager と同じグループに属する場合、グループに Concurrent Manager を追加する前に、データベースをグループに追加	これは、Concurrent Manager およびそれが関連するデータベースの間に適切な依存性を設定するために必要です。  「リソース」→「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い Oracle データベースを作成および構成する際に役立ちます。
7	内部 Oracle Applications Concurrent Manager をグループに追加	「リソース」→「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い内部 Oracle Applications Concurrent Manager を作成および構成する際に役立ちます。
8	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
9	その他のアプリケーション・コンポーネントをフェイルオーバー用に構成	その他のアプリケーション・サーバーおよびデータベース・サーバーを Oracle Fail Safe を使用して構成し、アプリケーション全体の可用性を向上させます。

13.3.3 Oracle Applications Concurrent Manager の構成データ

Oracle Fail Safe Manager には、可用性が向上するように内部 Oracle Applications Concurrent Manager を構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- Oracle Applications Concurrent Manager と関連付ける仮想アドレス
- 内部 Oracle Applications Concurrent Manager の名前およびアプリケーション構成 ID
- Oracle Applications Concurrent Manager を実行する管理者アカウント
- 内部 Oracle Applications Concurrent Manager により要求がフェッチされるデータベースの接続情報

- 内部 Oracle Applications Concurrent Manager が関連付けられているデータベースのリソース名（データベースも Concurrent Manager と同じグループ内で高い可用性のために構成されている場合）

次の場合、内部 Oracle Applications Concurrent Manager をグループに追加するクラスタワイド操作は失敗します。

- クラスタに同じ名前の Concurrent Manager がすでに存在する場合
- Windows レジストリにアプリケーション構成 ID が存在しない場合
- Concurrent Manager の可能所有者ノードである各ノードの Concurrent Manager の実行アカウントが管理者アカウントではない場合
- Oracle Applications 11i が Concurrent Manager の可能所有者ノードである全クラスタ・ノードにインストールされていない場合
- 内部 Oracle Applications Concurrent Manager の接続先データベースが存在しないか、「リソースをグループに追加」ウィザードにデータベースへの不正な接続情報が入力されたか、このデータベースの TNSNAMES.ORA ファイル内の情報が正しくない場合

Oracle Applications Concurrent Manager をグループへ追加またはグループから削除する際にエラーが発生した場合は、Windows イベント ビューア（アプリケーション・ログ）を調査し、Oracle Fail Safe からのエラー・メッセージの有無を確認します。

次の各項では、内部 Oracle Applications Concurrent Manager をグループに追加するためのデータ要件を詳細に説明します。

### 13.3.3.1 ノードの選択

Oracle Applications Concurrent Manager をグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、[図 13-2](#) に示すように、選択済ノードのリストを指定して、Oracle Applications Concurrent Manager の可能所有者となるノードを指定するように求められます。特定のノードを Oracle Applications Concurrent Manager の可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

[2.6.6 項](#)では、可能所有者ノードの概念について詳細に説明します。

図 13-2 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle Applications Concurrent Manager をグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを Oracle Applications Concurrent Manager の可能所有者とするのかを指定するように求められます。このような場合、図 13-3 に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 13-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ

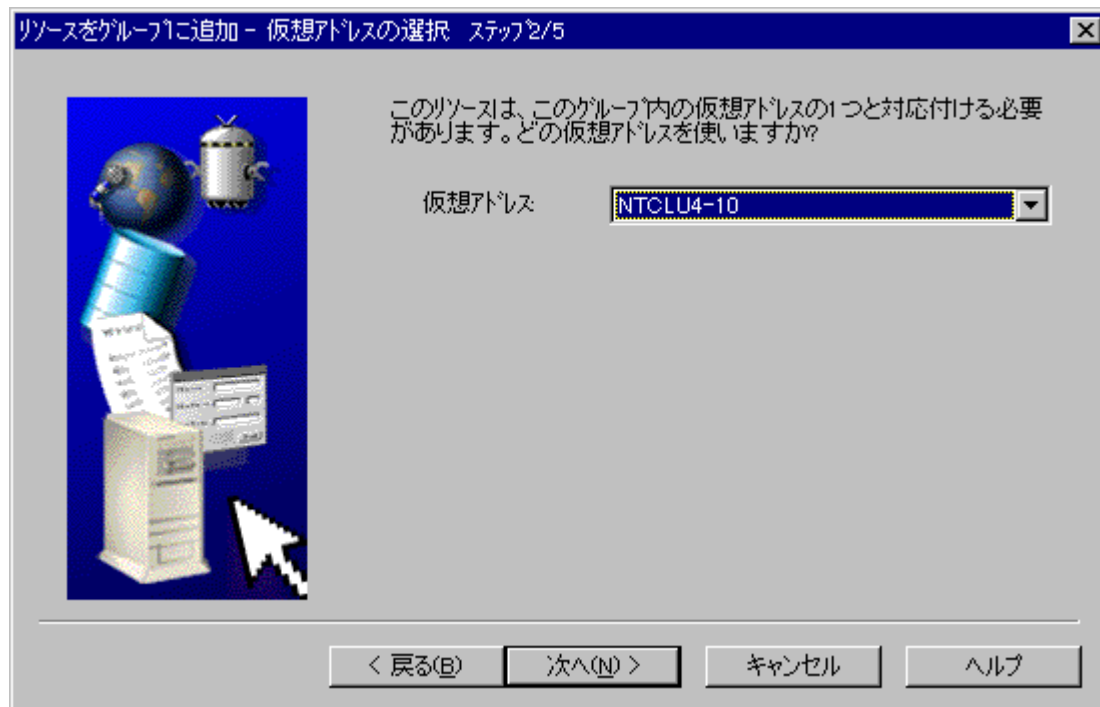


### 13.3.3.2 Oracle Applications Concurrent Manager 仮想アドレス

グループには、内部 Oracle Applications Concurrent Manager を追加する前に、少なくとも 1 つの仮想アドレスを持っている必要があります。図 13-4 に示すように、内部 Oracle Applications Concurrent Manager を追加するグループ内に複数の仮想アドレスが含まれている場合、内部 Oracle Applications Concurrent Manager と関連付ける仮想アドレスを選択するように要求されます。

仮想アドレスは、ホスト（マシン）名として Concurrent Manager によって使用されるアドレスとなります。

図 13-4 Oracle Applications Concurrent Manager の「仮想アドレスの選択」ウィザード・ページ



### 13.3.3.3 Concurrent Manager の識別情報

図 13-5 に示すように、「リソースをグループに追加」ウィザードでは、内部 Oracle Applications Concurrent Manager の識別情報に関する次の情報が要求されます。

- Concurrent Manager 名

クラスタ上の内部 Oracle Applications Concurrent Manager を一意に識別する名前です。

- 構成 ID

これは、アプリケーション構成 ID です。これにより、Windows レジストリ内のエントリが識別されます。

¥¥HKEY\_LOCAL\_MACHINE¥Software¥Oracle¥APPLICATIONS¥11.5.0¥<config-id>

このキーにより、内部 Oracle Applications Concurrent Manager を実行する環境が定義されます。これは、Oracle Applications 11i のインストール時に定義されます。

- ディレクトリ

これにより、Oracle Applications Concurrent Manager ログ・ファイルおよび出力ファイルのクラスタ・ディスクおよび親ディレクトリが指定されます。ログ・ファイルは親ディレクトリのサブディレクトリ ¥log に置かれ、出力ファイルは、親ディレクトリのサブディレクトリ ¥output に置かれます。

たとえば、ディレクトリ X:¥Conf と指定した場合、ログ・ファイルは、X:¥Conf¥log に置かれ、出力ファイルは X:¥Conf¥output に置かれます。

親ディレクトリは、次の Windows レジストリ・キーの APPLCSF 変数に格納されます。

```
¥¥HKEY_LOCAL_MACHINE¥Software¥Oracle¥APPLICATIONS¥11.5.0¥<config-id>
```

図 13-5 「Concurrent Manager の識別情報」ウィザード・ページ

リソースをグループに追加 - Concurrent Managerの識別情報 ステップ3/5

グループ名 My\_group

このグループに追加したい Concurrent Manager はどれですか?  
既存の Concurrent Manager の名前から選択できます。または、  
一意な名前を入力することにより Oracle Fail Safe で新しい  
Concurrent Manager を作成できます。

Concurrent Manager 名 Test\_Conc\_Man

この Concurrent Manager の構成 ID は何ですか?

構成 ID: PROD

ログおよび出力ファイルを格納したいディレクトリは何ですか?  
これらのファイルは共有ディスクに格納される必要があります。

ディレクトリ: X¥Apps\_Logs

< 戻る(B) 次へ(N) > キャンセル ヘルプ

### 13.3.3.4 Concurrent Manager の認証

図 13-6 に示すように、「リソースをグループに追加」ウィザードでは、内部 Oracle Applications Concurrent Manager を実行するアカウントに関する次の情報が要求されます。

- ユーザー名

これは、管理者権限を持つユーザー・アカウントです。これがドメイン・アカウントではない場合は、内部 Oracle Applications Concurrent Manager の可能所有者である各ノードに、このアカウントが存在することを確認する必要があります。

- パスワード

これは、ユーザー・アカウント用のパスワードです。

- ドメイン

これは、ユーザー・アカウント用のドメインです。ユーザー・アカウントがドメイン・アカウントではない場合、このボックスは空白のまま残します。ドメイン・アカウントを指定することは必須ではありませんが、このようなアカウントは管理を容易にするため推奨されます。

図 13-6 「Concurrent Manager の認証」ウィザード・ページ



リソースをグループに追加 - Concurrent Manager の認証 ステップ 4/5

Concurrent Manager は、管理者のアカウントのもとで実行する必要があります。どのアカウントのもとで Concurrent Manager を実行しますか?

ユーザー名: Administrator

パスワード: \*\*\*\*\*

パスワードの確認: \*\*\*\*\*

ドメイン: FCLUSTER

< 戻る(B)    次へ(N) >    キャンセル    ヘルプ



### 13.3.3.5 Concurrent Manager のデータベース

図 13-7 に示すように、「リソースをグループに追加」ウィザードでは、内部 Oracle Applications Concurrent Manager の要求取得先であるデータベースに関する次の情報が要求されます。

- ユーザー名

これは、内部 Oracle Applications Concurrent Manager が読み取る要求が含まれるデータベースにアクセスできるアカウントのユーザー名です。

- パスワード

これは、アカウントのパスワードです。

- サービス名

これは、データベースのサービス名です。

- データベース

これは、指定されたサービス名のデータベース名です。次の両方のケースに該当する場合は、このボックスからデータベースを選択します。

- 内部 Oracle Applications Concurrent Manager の追加先であるグループに Concurrent Manager の接続先のデータベースが含まれている場合
- データベース・リソース名がデータベース・サービス名とは異なる場合

データベース・リソース名とデータベース・サービス名が同じである場合、データベース・ボックスはグレー表示されます。

データベース・リソース名とデータベース・サービス名が異なる場合、データベース・ボックスはグレー表示されず、データベース・リソースを指定するように求められます。Oracle Fail Safe では、接続情報を提供しているデータベースとグループ内のデータベースが同じであることを自動的に判別できません。

内部 Oracle Applications Concurrent Manager の追加対象であるグループと同じグループのデータベースではあるが、Concurrent Manager の接続先データベースではない場合は、データベース・ボックスで選択しないでください。

図 13-7 「Concurrent Manager のデータベース」ウィザード・ページ



リソースをグループに追加 - Concurrent Manager のデータベース ステップ 5/5

Concurrent Manager が関連付けられているデータベースの接続情報は何か?

ユーザー名: User5

パスワード: \*\*\*\*\*

パスワードの確認: \*\*\*\*\*

サービス名: PRODUCTION

この Concurrent Manager が依存するデータベース・リソースを選択します。

データベース: -- なし --

< 戻る(B)    完了(F)    キャンセル    ヘルプ

## 13.4 Oracle Applications Concurrent Manager のセキュリティ要件

内部 Oracle Applications Concurrent Manager を管理するには、管理者権限を持つユーザー・アカウントを使用します。このアカウントは、Concurrent Manager の可能所有者ノードである各クラスタ・ノードに対する管理者権限を持っている必要があります。アカウントには、ドメイン・アカウント（推奨）または各ノードについてローカルのアカウントのいずれかが選択できます。

内部 Oracle Applications Concurrent Manager によるデータベースへの接続に使用される（Concurrent Manager の要求のフェッチ先である）アカウントには、特別な権限は必要ありません。

## 13.5 可用性の高い Oracle Applications Concurrent Manager に関連する問題のトラブルシューティング

Fail Safe リソースのトラブルシューティングに関する一般的な情報は、第 6 章に記載しています。内部 Oracle Applications Concurrent Manager のトラブルシューティングに関する一般情報は、Oracle Applications 11i のマニュアルを参照してください。

高い可用性を持つように構成されている内部 Oracle Applications Concurrent Manager に関連する問題が発生した場合は、「クラスタの検証」操作を実行して有効性を検査し、問題があれば修正します。

内部 Oracle Applications Concurrent Manager が含まれるグループに対して「グループの検証」コマンドを実行すると、Oracle Fail Safe では、Concurrent Manager の可能所有者ノードであるすべてのノードに対し、次のことを行います。

- Concurrent Manager に対する Windows サービスの存在の確認
- データのフェッチ先であるデータベースと Concurrent Manager の接続チェック
- Concurrent Manager の実行アカウントの認証および管理者アカウントであることの確認
- 依存性の有効性検査
- データのフェッチ元であるデータベース（Windows レジストリで指定）が全可能所有者ノードにおいて同一であるかの確認

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合。
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合。
- Concurrent Manager の可能所有者であるノードから誤って Concurrent Manager 用の Windows サービスを削除してしまった場合。「グループの検証」操作により、Concurrent Manager 用の Windows サービスが再作成されます。
- クラスタに新しいノードが追加された場合。

「グループの検証」で問題が検出された場合、Oracle Fail Safe は問題を修正するための応答を求めるか、または問題を詳しく説明したエラー・メッセージを返します。

### 13.5.1 Oracle Applications Concurrent Manager のグループへの追加の問題

スタンドアロン内部 Oracle Applications Concurrent Manager をグループに追加する場合の問題を解決するには、次のことを確認します。

- Concurrent Manager により使用されるデータベースが設定済みでありアクセス可能であること
- Concurrent Manager の可能所有者ノードとなる各ノードの Concurrent Manager の実行アカウントが管理者権限を持つこと
- 「リソースをグループに追加」で指定されたアプリケーション構成 ID が Concurrent Manager の可能所有者ノードであるすべてのクラスター・ノードに存在すること

### 13.5.2 グループのオンライン化の問題

内部 Oracle Applications Concurrent Manager が含まれるグループのオンライン化の際に問題が生じた場合は、次のことを試してください。

- グループを検証します。

(Oracle Fail Safe Manager の「トラブルシューティング」メニューから)「グループの検証」コマンドを使用すると、Oracle Fail Safe ではグループの構成をチェックし、問題が見つかったら修正を行います。「グループの検証」コマンドで問題を修正できない場合は、その問題を手動で解決するヒントとなるエラー・メッセージが返されます。

「グループの検証」コマンドで問題が見つかった場合には、その後の手順についてユーザーの応答が求められます。

- 保留タイムアウト値が十分であることを確認します。

内部 Oracle Applications Concurrent Manager を含むグループがオンライン化に失敗する、または頻繁にフェイルオーバーする場合は、保留タイムアウト値が正しく設定されているかチェックしてください。保留タイムアウト値の設定が低すぎると、オンライン化の失敗、頻繁なフェイルオーバーが発生します。

保留タイムアウト値には、内部 Oracle Applications Concurrent Manager の不安定な状態（使用不能、オフライン、保留状態など）が解消されないままだけの時間が経過すると、クラスター・ソフトウェアでリソースに障害が発生したと判断するかを指定します。たとえば、クラスター・システムに多大な負荷がかかり、内部 Oracle Applications Concurrent Manager があたかも使用不能になったかのように動作したとします。この場合、保留タイムアウト値にデフォルトの 180 秒よりも大きな値を設定してください。

保留タイムアウト値を設定するには、次のようリソース・プロパティを変更します。

1. Oracle Fail Safe Manager のツリー・ビューで、内部 Oracle Applications Concurrent Manager 名を選択します。
2. 「ポリシー」タブをクリックします。

3. 「**保留タイムアウト**」ボックスで保留タイムアウト値を変更します。

保留タイムアウト値には、クラスタ・ソフトウェアでリソースがオンライン（またはオフライン）になってから操作が失敗したと判断されるまでの時間を指定します。設定した値が小さすぎる場合、応答時間が長くなるとクラスタ・システムで誤って使用不能状態と判断する可能性があり、値が大きすぎる場合、障害発生時のフェイルオーバー応答時間が長くなります。このどちらも回避できるように適切な値を設定してください。

### 13.5.3 Concurrent Manager 用の Windows サービスが誤って削除された

Concurrent Manager の可能所有者ノードであるクラスタから、Concurrent Manager 用の Windows サービスを誤って削除してしまった場合は、「グループの検証」操作を実行して Windows サービスを再作成します。

### 13.5.4 Concurrent Manager がオンライン化できない

Concurrent Manager がオンライン化できない場合は、「グループの検証」操作を実行します。「グループの検証」操作により、次のどれによって問題が発生しているのかが特定されます。

- データベース接続文字列に問題があるため、Concurrent Manager は要求のフェッチ先のデータベースに接続できません。
- Concurrent Manager の実行アカウントに管理者権限がありません。

「グループの検証」操作によってこれらのいずれかが問題として挙げられた場合は、Oracle Fail Safe Manager のツリー・ビューでその Concurrent Manager を選択し、「データベース」タブを選択してデータベース接続文字列を変更するか、「パラメータ」タブを選択してその Concurrent Manager 用のユーザー・アカウントを変更します。

これらのどちらも問題の原因に該当しない場合は、Windows イベント ビューアを調べて、Concurrent Manager がオンライン化されない問題の原因を突き止めます。



## 可用性を高めるための Oracle MTS Service の構成

Oracle Fail Safe は、可用性が高まるよう Oracle Service for Microsoft Transaction Server (MTS) を構成する際に役立ちます。(このマニュアルでは、Oracle Service for MTS を Oracle MTS Service と呼んでいます。) クラスタ・ノードが停止または障害状態に陥ると、Oracle MTS Service を含むグループは自動的に別のクラスタ・ノードへとフェイルオーバーします。

この章では、次の項目について説明します。

項目	参照
概要	<a href="#">14.1 項</a>
スタンドアロン Oracle MTS Service の検出	<a href="#">14.2 項</a>
Oracle MTS Service のグループへの追加	<a href="#">14.3 項</a>
Oracle MTS Service のセキュリティ要件	<a href="#">14.4 項</a>
可用性の高い Oracle MTS Service に関連する問題のトラブルシューティング	<a href="#">14.5 項</a>

Microsoft Transaction Server は、Microsoft 独自の COM ベースのトランザクション処理システムであり、インターネットまたはネットワーク・サーバー上で稼働します。Microsoft Transaction Server は、クライアント・コンピュータにかわってアプリケーション要求およびデータベース・トランザクション要求を配布および管理するために使用されます。Microsoft Transaction Server には、次のものが用意されています。

- 分散アプリケーションを開発するための ActiveX/Distributed Component Object Model (DCOM) ベースのプログラミング・モデルと、これらのアプリケーションを配布するランタイム環境

- トランザクションのコンポーネントに対する Atomicity Consistency Isolation Durability (ACID) プロパティ
- コンポーネント・キャッシングやデータベース接続プーリングなどのパフォーマンス拡張機能へのアクセス

Microsoft Transaction Server は、3 層のサーバー中心アーキテクチャ・モデルのコンポーネントです。Microsoft Transaction Server により、プレゼンテーション、ビジネス・ロジックおよびアプリケーションの要素を、ネットワークに接続されている様々なコンピュータ上で明確に分離できます。

特別なデータベース統合がなくても、Oracle データベースに接続する Microsoft Transaction Server に COM コンポーネントを配布できます（サポートされるバージョンのリストは、『Oracle Fail Safe リリース・ノート』を参照してください）。ただし、次の機能には、Oracle MTS Service ソフトウェアのインストールが必要です。

- （Microsoft 管理コンソール エクスプローラの「プロパティ」ダイアログ・ボックスを使用して）COM コンポーネントをトランザクションとしてマークし、Microsoft Transaction Server でトランザクションを制御する機能
- Microsoft Transaction Server で使用されるクライアント側接続プーリング

Oracle MTS Service ソフトウェアがインストールされた後で、COM コンポーネントが接続する各 Oracle データベースに対して特別な Windows NT サービス（Oracle MTS Service と呼ぶ）を作成します。これらのサービスを作成するには、Microsoft 管理コンソール エクスプローラにある Oracle Manager for MTS Service スナップインを使用します。Oracle MTS Service には、次の機能が用意されています。

- Microsoft 独自のすべての OLE（Object, Linking, and Embedding）トランザクションから Oracle トランザクションへのマップ
- リカバリ・シナリオの処理

COM コンポーネントは、Oracle Call Interface (OCI)、Oracle Objects for OLE (OO4O)、Open Database Connectivity (ODBC) または Oracle Provider for OLE DB を使用して作成できます。将来的には、その他のアプリケーション・プログラム・インタフェース (API) も統合される可能性があります。ある特定の Oracle データベースでは、ネットワーク上に 1 つの Oracle MTS Service のみが必要です。

## 14.1 概要

Oracle MTS Service に障害が発生した場合は、Oracle MTS Service を使用するアプリケーション・トランザクションが失敗します。可用性が高まるように Oracle MTS Service を構成することにより、構成内の障害箇所をなくすることができます。

階層化実装は、Microsoft Transaction Server および Oracle MTS Service を使用してソリューションを構成する一般的な方法です。



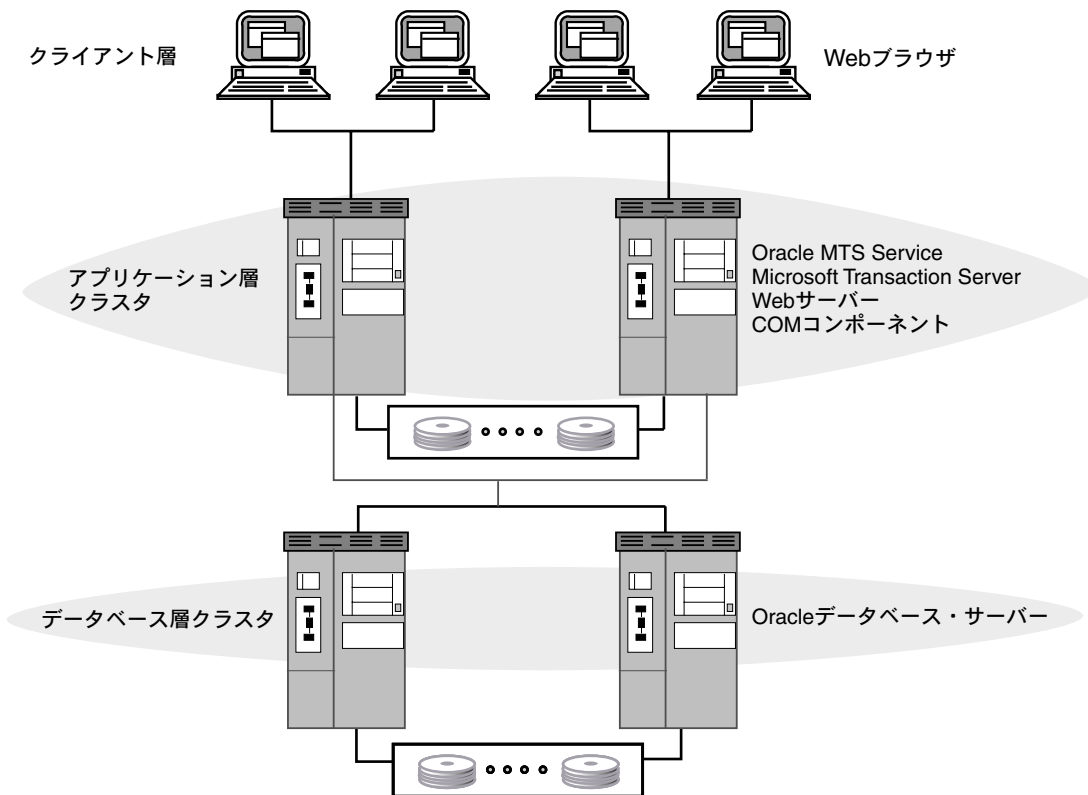
図 14-1 に、Microsoft Transaction Server、Oracle MTS Service、Web サーバーおよび COM コンポーネントがデータベースとは別のクラスタに常駐する階層化実装を示します。

図 14-1 の 3 つの層は次のとおりです。

- クライアント層： Web ブラウザ、ユーザー・インタフェース・アプリケーション
- アプリケーション層： Microsoft Transaction Server、Oracle MTS Service、Web サーバーおよび COM コンポーネント
- データベース層： Oracle データベース・サーバーおよびリスナー

階層化実装では、ソリューションに対して高可用性を増分的に展開できます。たとえば、Oracle Fail Safe を使用し、アプリケーション層またはデータベース層のコンポーネントで障害の発生する可能性のある箇所を排除することや、この 2 つの層を可用性が高まるようにクラスタ上に構成することができます。

図 14-1 可用性の高い Oracle MTS Server のための 3 層アーキテクチャ



## 14.2 スタンドアロン Oracle MTS Service の検出

Oracle Fail Safe Manager を使用し、可用性が高まるようにスタンドアロン Oracle MTS Service を構成します。Oracle Fail Safe Manager インタフェースでスタンドアロン Oracle MTS Service を表示するには、まずスタンドアロン Oracle MTS Service を検出する必要があります。

Oracle Fail Safe Server では、Oracle MTS Service を検出する際、Windows サービス マネージャ内のリストを調べて Oracle MTS Service エントリを見つけ、可用性が高まるように構成されているかどうかを判断します。Oracle Fail Safe では、新しく検出されたサービスが Oracle Fail Safe Manager のツリー・ビューの「スタンドアロン・リソース」フォルダの下に表示されます。

## 14.3 Oracle MTS Service のグループへの追加

可用性が高まるように Oracle MTS Service を構成するには、Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用して Oracle MTS Service をグループに追加します。指定するグループには、Microsoft および 1 つ以上の仮想アドレスがすでに含まれている必要があります。通常、このグループには次のリソースが含まれます。

- オペレーティング・システム・ソフトウェアのインストール時に可用性が高まるように構成された Microsoft
- 仮想アドレス (IP アドレスとネットワーク名)
- 1 つ以上の Oracle MTS Service
- Oracle MTS Service および Microsoft で使用されるすべてのディスク

クラスタでは、任意の時点に 1 つの Microsoft のみが実行でき、分散トランザクションに関連する各データベースに対して 1 つの Oracle MTS Service が必要であることに注意してください。したがって、クラスタ上のすべての Oracle MTS Service は、Microsoft を含むグループに追加される必要があります。

### 14.3.1 構成前に

Oracle MTS Service をグループに追加する前に、次のことに注意してください。

- Microsoft 分散トランザクション・コーディネータ (MS DTC) は、すべてのクラスタ・ノードで可用性が高まるように構成されている必要があります。(Windows 2000 を使用している場合、MS DTC は、クラスタに Windows 2000 をインストールする際にインストールされ、可用性が高まるように構成されます。Windows NT を使用している場合、MS DTC は、クラスタに Windows NT オプション・パックをインストールする際にインストールされ、可用性が高まるように構成されます。)

Windows 2000 を使用している場合、MS DTC リソースはクラスタ・グループに自動的に追加されます。ただし、MS DTC リソースおよび Oracle MTS Service リソースは、同じグループにある必要があります。したがって、Oracle MTS Service リソースを追加す

るグループに MS DTC リソースを手動で移動するか、Oracle Fail Safe Server で MS DTC リソースを移動する必要があります。

MS DTC リソースが新規グループに移動された後、すべてのアプリケーションが新規グループの仮想アドレスを使用して MS DTC にアクセスすることを確認します。

ツリー・ビューの「クラスタ・リソース」フォルダの下にある MS DTC リソースを探すことにより、MS DTC がすべてのクラスタ・ノードで可用性が高まるように構成されていることを確認できます。MS DTC リソースを含むグループを各クラスタ・ノードに移動することにより、MS DTC がすべてのクラスタ・ノードで稼働するように構成されているかどうかを確認できます。

(サポートされるオペレーティング・システム・ソフトウェア・バージョンの詳細は、『Oracle Fail Safe リリース・ノート』を参照してください。)

- Oracle MTS Service ソフトウェアは、すべてのクラスタ・ノード上のプライベート・ディスクにインストールされている必要があります。
- Oracle MTS Service は、分散トランザクションに関係するクラスタ上の各データベースに対して作成される必要があります。
- Microsoft 管理コンソールと Oracle Manager for Microsoft Transaction Server がどのクラスタ・ノードでも実行されていないことを確認します。追加操作は Oracle MTS Service を変更するため、排他的アクセス権が必要です。追加操作を完了した後で、Oracle Manager for Microsoft Transaction Server を再起動できます。

## 14.3.2 構成手順

Oracle Fail Safe を使用して Oracle MTS Service の可用性を高めるために必要な手順は、Oracle Fail Safe Manager により自動化されます。Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードで、リソース・タイプに Oracle MTS Service を指定してください。

表 14-1 に、可用性が高まるように Oracle MTS Service を構成する際に必要な作業の一覧を示します。特定の作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択します。

表 14-1 Oracle MTS Service を構成する手順

手順	処置	説明
1	オペレーティング・システム・ソフトウェアがクラスタにインストールされた際に、可用性が高まるよう Microsoft 分散トランザクション・コーディネータ (MS DTC) が構成されたことの確認	MS DTC の詳細は、Microsoft オペレーティング・システムのマニュアルを参照してください。
2	各クラスタ・ノードのプライベート・ディスク上に Oracle MTS Service ソフトウェアがインストールされていることの確認	Oracle MTS Service ソフトウェアのインストールの詳細は、Oracle データベース・サーバーのマニュアルを参照してください。
3	Oracle Fail Safe をインストールする際に Oracle MTS Service コンポーネントをインストール	Oracle Fail Safe Server をインストールするときに、「使用可能な製品コンポーネント」リストから Oracle MTS Service コンポーネントを選択します。これで DLL ファイルをインストールします。詳しい指示は、『Oracle Fail Safe インストレーション・ガイド』を参照してください。
4	Microsoft 管理コンソールと Oracle Manager for Microsoft Transaction Server がどのクラスタ・ノードでも実行されていないことの確認	追加操作は Oracle MTS Service を変更するため、排他的アクセス権が必要です。追加操作を完了した後で、Oracle Manager for Microsoft Transaction Server を再起動できます。
5	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 <b>「Oracle - &lt;Oracle_Home&gt;」 → 「Oracle Fail Safe Manager」</b>
6	クラスタの検証	<b>「トラブルシューティング」 → 「クラスタの検証」</b> を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
7	MS DTC リソースが適切なグループにあることの確認	Windows 2000 を実行している場合、MS DTC リソースはクラスタ・グループに自動的に追加されます。Oracle MTS Service を追加するグループに MS DTC リソースを移動するか、Oracle Fail Safe Server でこの処理を行う必要があります。
8	必要に応じて、MS DTC を含むグループに仮想アドレスを追加	グループに仮想アドレスを追加するには、「リソース」 → <b>「グループに追加」</b> を選択します。(仮想アドレスは、可用性が高まるように MS DTC が構成された際に追加されています。)

表 14-1 Oracle MTS Service を構成する手順（続き）

手順	処置	説明
9	MS DTC を含むグループに Oracle MTS Service を追加	「リソース」→「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い Oracle MTS Service を作成および構成する際に役立ちます。
10	グループの検証	「トラブルシューティング」→「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。
11	その他のアプリケーション・コンポーネントをフェイルオーバー用に構成	その他のアプリケーション・サーバーおよびデータベース・サーバーを Oracle Fail Safe を使用して構成し、アプリケーション全体の可用性を向上させます。

### 14.3.3 Oracle MTS Service の構成データ

Oracle Fail Safe Manager には、可用性が向上するよう Oracle MTS Service を構成する際に役立つ「リソースをグループに追加」ウィザードがあります。「リソースをグループに追加」ウィザードを使用する場合、次のデータが必要です。

- MS DTC を含むグループの名前
- Oracle MTS Service に関連付ける仮想アドレス（グループに複数の仮想アドレスがある場合）
- Oracle MTS Service 名
- Oracle MTS Service リソース名（Oracle Fail Safe Manager で Oracle MTS Service を参照するために使用する名前）
- Oracle MTS Service に関連付けられているデータベースの接続情報
- Oracle MTS Service のログインに使用するオペレーティング・システム・ドメイン・ユーザー・アカウント

Oracle MTS Service をグループに追加するクラスタワイド操作は、次のいずれかの場合は失敗します。

- 指定した Oracle MTS Service が正しく構成されていない場合。Oracle MTS Service をグループに追加する前に、スタンドアロン・サービスとして実行できることを確認してください。
- MS DTC を含まないグループに Oracle MTS Service の追加を試行した場合。
- Oracle MTS Service の実行に使用するアカウントがドメイン・アカウントでない場合。
- MS DTC と Oracle MTS Service が Concurrent Manager の可能所有者ノードである全クラスタ・ノードにインストールされていない場合。

- Oracle MTS Service の接続先データベースが存在しないか、「リソースをグループに追加」ウィザードにデータベースへの不正な接続情報が入力されたか、このデータベースの TNSNAMES.ORA ファイル内の情報が正しくない場合。

Oracle MTS Service をグループへ追加またはグループから削除する際にエラーが発生した場合は、Windows イベント ビューア（アプリケーション・ログ）を調査し、Oracle Fail Safe からのエラー・メッセージの有無を確認します。

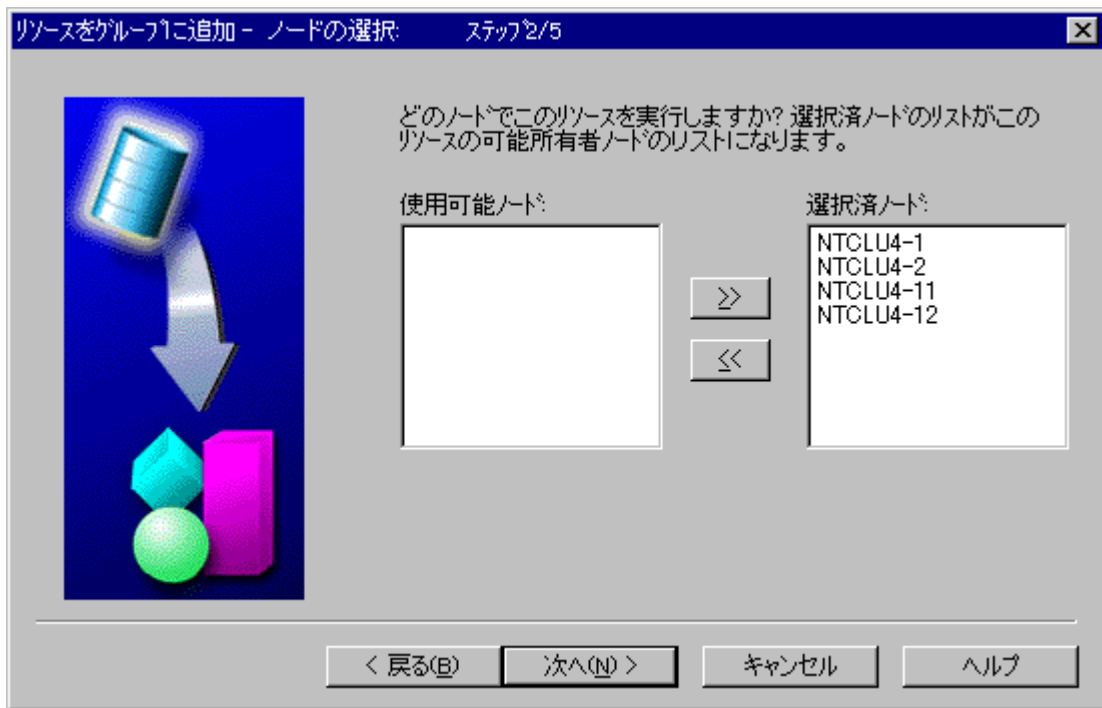
次の各項では、Oracle MTS Service をグループに追加するためのデータ要件を詳細に説明します。

#### 14.3.3.1 ノードの選択

Oracle MTS Service をグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、[図 14-2](#) に示すように、選択済ノードのリストを指定することにより Oracle MTS Service の可能所有者となるノードを指定するように求められます。特定のノードを Oracle MTS Service の可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択し、左矢印をクリックします。

[2.6.6 項](#)では、可能所有者ノードの概念について詳細に説明します。

図 14-2 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



Oracle MTS Service をグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、Oracle MTS Service の可能所有者にするノードを指定するように求められます。このような場合、[図 14-3](#) に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 14-3 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



#### 14.3.3.2 Oracle MTS Service 仮想アドレス

Oracle MTS Service をグループに追加するには、そのグループに少なくとも 1 つの仮想アドレスが含まれている必要があります。図 14-4 に示すように、Oracle MTS Service を追加するグループに複数の仮想アドレスが含まれている場合は、Oracle MTS Service に関連付ける仮想アドレスを選択するように求められます。



図 14-4 Oracle MTS Service 「仮想アドレス」ウィザード・ページ



### 14.3.3.3 Oracle MTS Service 識別情報

図 14-5 に示すように、「リソースをグループに追加」ウィザードでは、Oracle MTS Service の識別情報に関する次の情報が要求されます。

- Oracle MTS Service 名

これは、クラスタ上の既存の Oracle MTS Service の名前です。提供される名前のリストは、Oracle Fail Safe でクラスタ上のスタンドアロン Oracle MTS Service として識別されている名前のリストです。

- リソース名

この名前は、Oracle Fail Safe Manager で Oracle MTS Service を参照するための名前です。この名前は、Oracle Fail Safe Manager ツリー・ビューに表示されます。

この名前は、Oracle MTS Service 名と同じ名前にすることができます。別の名前を指定する場合は、クラスタ間で一意であることを確認してください。

図 14-5 「Oracle MTS Service の識別情報」ウィザード・ページ

リソースをグループに追加 - Oracle MTS Serviceの識別情報:ステップ3/5

グループ名: MSMTS

このグループに追加したい Oracle MTS Service はどれですか?  
Oracle Fail Safeでグループに追加する既存のOracle MTS Service  
を選択してください。

サービス名: OracleMTSService0

このサービスのリソース名は何ですか? これはツリービューで使用される  
名前になります。

リソース名: OracleMTSService0

< 戻る(B)    次へ(N) >    キャンセル    ヘルプ

#### 14.3.3.4 Oracle MTS Service データベース

図 14-6 に示すように、「リソースをグループに追加」ウィザードでは、Oracle MTS Service が関連付けられているデータベースに関する次の情報が要求されます。

- ユーザー名  
これは、データベース・アカウントのユーザー名です。このフィールドは、「リソースをグループに追加」操作中に変更できません。Oracle MTS Service で使用するデータベース・アカウントを変更する場合は、「リソースをグループに追加」操作を実行する前または後に行ってください。
- パスワード  
これは、「ユーザー名」フィールドで指定したアカウントのパスワードです。

- パスワードの確認  
これは、「パスワード」フィールドに入力したパスワードと同じです。
- サービス名  
これは、アカウントが関連付けられているデータベースのサービス名です。

図 14-6 「Oracle MTS Service のデータベース」ウィザード・ページ



リソースをグループに追加 - Oracle MTS Serviceのデータベース:ステップ4/5

Oracle MTS Serviceが関連付けられているデータベースの接続情報は何か?

ユーザー名: MTSSYS

パスワード: \*\*\*\*\*

パスワードの確認: \*\*\*\*\*

サービス名: mtsdemo

< 戻る(B)    次へ(N) >    キャンセル    ヘルプ

#### 14.3.3.5 Oracle MTS Service 認証

「リソースをグループに追加」ウィザードは、Oracle MTS Service がログオンに使用するアカウントがユーザー・アカウントである場合に、認証情報（パスワード）を入力するように要求します。Oracle MTS Service がログオンに使用するアカウントがシステム・アカウントである場合は、図 14-7 に示すように、このページにはデータを入力しません。

- ユーザー名

Oracle MTS Service がログオンに使用するアカウントの名前を表示します。Oracle MTS Service がシステム・アカウントを使用してログオンする場合、このフィールドには文字列 "LocalHost" が表示されます。Oracle MTS Service がユーザー・アカウントを使用してログオンする場合、このフィールドにはそのアカウント名が表示されます。

- パスワード

Oracle MTS Service がシステム・アカウントを使用してログオンする場合は何も表示されません。Oracle MTS Service がユーザー・アカウントを使用してログオンする場合は、ユーザー・アカウントのパスワードを入力します。

- パスワードの確認

Oracle MTS Service がシステム・アカウントを使用してログオンする場合は何も表示されません。Oracle MTS Service がユーザー・アカウントを使用してログオンする場合は、ユーザー・アカウントのパスワードを再入力します。

図 14-7 「Oracle MTS Service の認証」ウィザード・ページ



リソースをグループに追加 - Oracle MTS Serviceの認証:ステップ5/5

サービスがログオンできるWindowsアカウントを指定してください。

ユーザー名: LocalSystem

パスワード:

パスワードの確認:

< 戻る(B)    完了(F)    キャンセル    ヘルプ

## 14.4 Oracle MTS Service のセキュリティ要件

Oracle MTS Service がシステムにログオンするために使用するアカウントは、システム・アカウントまたはドメイン・ユーザー・アカウントである必要があります。

Oracle MTS Service がデータベースに接続するために使用するアカウントには、次のデータベース・ロール、権限および権利が必要です。

- CONNECT、RESOURCE および SELECT\_CATALOG\_ROLE の各データベース・ロール
- FORCE ANY TRANSACTION、CREATE PUBLIC SYNONYM および DROP PUBLIC SYNONYM の各データベース権限
- Oracle データベース内の Oracle MTS Service 表に対する SELECT、INSERT、UPDATE および DELETE の各権限

## 14.5 可用性の高い Oracle MTS Service に関連する問題のトラブルシューティング

Fail Safe リソースのトラブルシューティングに関する一般的な情報は、[第 6 章](#)に記載しています。Oracle MTS Service のトラブルシューティングの一般情報は、『Oracle8 と Microsoft Transaction Server の連携』を参照してください。

可用性が高まるよう構成された Oracle MTS Service に問題が発生した場合は、「クラスタの検証」操作を実行して検証し、問題があれば修正します。

Oracle MTS Service が含まれるグループに対して「グループの検証」コマンドを実行すると、Oracle Fail Safe では、Oracle MTS Service の可能所有者ノードであるすべてのノードに対し、次のことが行われます。

- Oracle MTS Service が存在することをチェックします。
- Oracle MTS Service が、関連付けられているデータベースに接続できることをチェックします。
- (Windows レジストリに記録されているように) Oracle MTS Service が関連付けられているデータベースが、すべての可能所有者ノードで同じであることをチェックします。
- Oracle MTS Service の実行に使用するアカウントを認証します。
- 依存性の有効性を検査します。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合。
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合。

- Oracle MTS Service の可能所有者ノードであるノードから、Oracle MTS Service を誤って削除してしまった場合。この場合は、「グループの検証」操作を実行して Oracle MTS Service 用の Windows サービスを再作成します。
- クラスタに新しいノードが追加された場合。

「グループの検証」で問題が検出された場合、Oracle Fail Safe は問題を修正するための応答を求めるか、または問題を詳しく説明したエラー・メッセージを返します。

## 14.5.1 Oracle MTS Service のグループへの追加の問題

グループにスタンドアロン Oracle MTS Service を追加する際に発生した問題のトラブルシューティングを行うには、次のことを確認します。

- Oracle MTS Service に関連付けられているデータベースが、Oracle MTS Service の可能所有者ノードであるすべてのクラスタ・ノードからアクセスできること。
- 必要な Oracle MTS Service の実行に使用するアカウントが、システム・アカウントまたはドメイン・ユーザー・アカウントであること。
- Windows レジストリ・エントリの LAST\_HOME が、ローカル・システム上で最も大きい Oracle ホーム番号に設定されていること。

Oracle MTS Service のグループへの追加を試行する際に、HKEY\_LOCAL\_MACHINE/SOFTWARE/ORACLE/ALL\_HOMES の LAST\_HOME エントリがシステム上の最も大きい Oracle ホーム番号と等しくない場合は、Oracle MTS Service のグループへの追加操作が失敗し、次のメッセージを受け取ります。

FS-10778: Oracle MTS Service リソース・プロバイダで、クラスタ・リソース<リソース名>の構成に失敗しました。

FS-10618: NT サービス<サービス名>の構成の変更に失敗しました。

0x424: 指定されたサービスはインストールされたサービスとして存在しません。

たとえば、システムに HOME0、HOME1 および HOME2 がある場合、Oracle MTS Service のグループへの追加操作が機能するためには、LAST\_HOME の値が 2 である必要があります。問題を修正するには、LAST\_HOME の値をローカル・システムで最も大きい Oracle ホーム番号と一致させ、「リソースをグループに追加」ウィザードを再実行します。

## 14.5.2 グループのオンライン化の問題

Oracle MTS Service が含まれるグループをオンラインにする際に問題が発生した場合は、グループを検証します。(Oracle Fail Safe Manager の「トラブルシューティング」メニューから)「グループの検証」コマンドを使用すると、Oracle Fail Safe ではグループの構成をチェックし、問題が見つかったら修正を行います。「グループの検証」コマンドで問題を修正できない場合は、その問題を手動で解決するヒントとなるエラー・メッセージが返されます。「グループの検証」コマンドで問題が見つかった場合には、その後の手順についてユーザーの応答が求められます。

### 14.5.3 削除された Oracle MTS Service の Windows サービス

Oracle MTS Service の可能所有者ノードであるクラスタ・ノードから、Oracle MTS Service を誤って削除してしまった場合は、「グループの検証」操作を実行して Oracle MTS Service を再作成します。





## 可用性を高めるための汎用サービスの構成

汎用サービスは、Microsoft Cluster Server (MSCS) で提供される汎用サービス・リソース DLL によってサポートされる Windows サービスです。Oracle Fail Safe では、可用性を高めるための汎用サービスの構成がサポートされており、次のことが可能です。

- 手持ちのアプリケーションの可用性を高めます。
- Oracle Fail Safe では現在特にサポートされていない他のアプリケーションの可用性を高めます。

たとえば、現在 Oracle Fail Safe では、Oracle データベース、Oracle WebDB リスナー、Oracle Forms Load Balancer Server、Oracle Reports Server、Oracle Intelligent Agent、Oracle Application Server、Oracle HTTP Server および Oracle Applications Concurrent Manager に高い可用性を持たせるための固有のサポートを提供しています。Oracle Fail Safe が提供する汎用サービスを使用して、他の Windows サービスも高い可用性を持つように構成できます。(15.5 項でこの方法について説明します。)

この章では、次の項目について説明します。

項目	参照
概要	15.1 項
スタンドアロン汎用サービスの検出	15.2 項
汎用サービスのグループへの追加	15.3 項
汎用サービスのセキュリティ要件	15.4 項
サンプル汎用サービスの構成	15.5 項
汎用サービスに関する問題のトラブルシューティング	15.6 項

## 15.1 概要

Oracle Fail Safe でサポートされている可用性を高めるためのリソースの構成と、可用性を高めるための汎用サービスの構成では、(Oracle Fail Safe Manager の)「リソースをグループに追加」ウィザードにより提供される支援レベルが異なります。サポートされているリソースの場合、「リソースをグループに追加」ウィザードでは、その特定のリソース用の構成情報が要求されます。汎用リソースの場合、「リソースをグループに追加」ウィザードに構成情報が備わっていないため、詳細な定義データは要求されません。このため、汎用サービスが依存するリソース、必須の Windows レジストリ・エントリなどに注意する必要があります。

### 15.1.1 Oracle Fail Safe を使用する利点

汎用サービスを構成するにあたり、MSCS ではなく Oracle Fail Safe を使用した場合の利点について考えます。次のような、様々な利点があります。

- Oracle Fail Safe を使用すると、既存のサービスを可用性が高まるように構成したり、「リソースをグループに追加」操作の一部として汎用サービスを作成し構成できます。一方、MSCS で可能なのは、既存の汎用サービスを可用性が高まるように構成することのみです。
- Oracle Fail Safe の「リソースをグループに追加」ウィザードには、可用性の高い汎用リソースを構成するために役立つより多くの質問があります。たとえば、このウィザードを使用すると、汎用リソースに必要なディスク、クラスタ全体に複製される必要のある Windows レジストリ・エントリなどを指定できます。MSCS のウィザードには、Oracle Fail Safe のウィザードほど多くの役立つ質問はありません。
- Oracle Fail Safe により、「リソースをグループに追加」操作の一部として、(可用性が高まるように構成された全タイプのリソースに行うのと同じように) 構成がテストされます。また、ネットワーク、フェイルオーバーおよびフェイルバックがテストされ、リソースがリソースの可能所有者となる全クラスタ・ノードで起動できることが確認されます。
- クラスタ環境では手動設定が必要になるため、Oracle Fail Safe により、各クラスタ・ノード上のリソースの起動タイプが手動に設定されます。MSCS で設定されるのは、リソースを所有しているノードの起動タイプのためのため、必ずその他の可能所有者となるクラスタ・ノードで起動タイプを手動に設定する必要があります。

Oracle Fail Safe インストールの「リソースをグループに追加」ウィザードで、特定のリソースが Oracle Fail Safe インストールに含めることのできるコンポーネントとして表示されない場合、「リソースをグループに追加」ウィザード内の Generic Service リソース・タイプを使用して、可用性が高まるようにサービスを構成します。

## 15.1.2 可用性を高める構成にしない汎用リソース

可用性が高まるように汎用サービスを構成すると、ある時点で1つのクラスタ・ノードでしか実行されなくなるため、構成する際にはこの点に注意してください。クラスタ・ノードで実行する必要があるサービスは、同時に可用性を高める構成にしないでください。たとえば、Windows イベント ログの場合を考えてみます。Windows イベント ログは、特定のシステム上のすべてのサービスにより通知メッセージやエラー・メッセージなどが書き込まれるファイルです。これは、サービスから管理者に状況を通知するための手段です。

Windows イベント ログのサービスの可用性を高めると、サービスはある時点で1つのクラスタ・ノードでしか動作しなくなります。この場合、他のクラスタ・ノード上のサービスにより返されたメッセージは、イベント ログを実行しているクラスタ・ノード上のイベント ログにアクセスできなくなります。したがって、Windows イベント ログをクラスタ・リソースとして構成することは賢明ではありません。

## 15.2 スタンドアロン汎用サービスの検出

Oracle Fail Safe Server は、汎用リソースを検出する際、Windows サービス マネージャで汎用リソースを検索します。検出プロセスの間に、Oracle Fail Safe はクラスタ内の各ノードの Windows サービス マネージャでサービスを見つけ、Oracle Fail Safe Manager のツリー・ビューに新しく検出したサービスを表示します。

## 15.3 汎用サービスのグループへの追加

可用性が高まるように汎用サービスを構成するには、Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用してサービスをグループに追加します。既存の汎用サービスをグループに追加したり、Oracle Fail Safe により自動的に汎用サービスを作成するように指定できます。次の各項では、構成手順と、汎用サービスについて「リソースをグループに追加」ウィザードを完了するために必要なデータについて説明します。

### 15.3.1 構成手順

表 15-1 に、可用性が高まるように汎用サービスを構成する際に必要な作業の一覧を示します。各作業の手順ごとの指示は、Oracle Fail Safe のオンライン・ヘルプとチュートリアルを参照してください。Oracle Fail Safe Manager のメニュー・バーで、「ヘルプ」→「キーワードで検索」を選択するか、または「ヘルプ」→「チュートリアル」を選択します。

表 15-1 汎用サービスの構成手順

手順	処置	説明
1	汎用サービスの可能所有者となる各クラスタ・ノードのプライベート・ディスク上に、汎用サービス実行可能ファイルがインストールされていることを確認	必須ではないが、強く推奨。通常は、複数のサービス・インスタンスが同じ実行可能ファイルを使用します。実行可能ファイルが共有クラスタ・ディスクにインストールされている場合、その実行可能ファイルを使用するすべてのサービスが、現在そのディスクのホストになっているクラスタ・ノードで稼働する必要があります。
2	汎用サービスに必要なファイルをクラスタ・ディスクへコピー	汎用サービスに必要なデータ・ファイルは、共有記憶装置インターコネクト上のクラスタ・ディスクに配置する必要があります。
3	Oracle Fail Safe Manager の起動	Windows の「スタート」メニューから、次の順に選択します。 「<Oracle_Home>」 → 「Oracle Fail Safe Manager」
4	クラスタの検証	「トラブルシューティング」 → 「クラスタの検証」を選択して、クラスタのハードウェアおよびソフトウェア構成を検証するプロシージャを実行します。
5	グループの作成	「グループ」 → 「作成」を選択して、「グループの作成」ウィザードを実行します。このウィザードを使用して、フェイルオーバーおよびフェイルバック・ポリシーを設定します。「リソースをグループに追加」ウィザードが自動的に開くため、ここで仮想アドレスをグループに追加します。Oracle Fail Safe では、汎用サービスを追加する前に仮想アドレスをグループに追加するように要求されません。ただし、汎用サービスが依存するリソースに対して仮想アドレスが必要になる場合があります。詳細は、 <a href="#">15.3.2.5.2 項</a> を参照してください。
6	必要な場合、グループに 1 つ以上の仮想アドレスを追加	「リソース」 → 「グループに追加」を選択して、「リソースをグループに追加」ウィザードを実行します。このウィザードは、可用性の高い仮想サーバー・アドレスを作成および構成する際に役立ちます。
7	汎用サービスが依存するリソースの追加	「リソース」 → 「グループに追加」を選択して、「リソースをグループに追加」ウィザードを開きます。
8	汎用サービスをグループに追加	「リソース」 → 「グループに追加」を選択して、「リソースをグループに追加」ウィザードを開きます。このウィザードは、汎用サービスをグループ内に構成する際に役立ちます。汎用サービスを新規に作成するか、または既存の汎用サービスを指定できます。
9	グループの検証	「トラブルシューティング」 → 「グループの検証」を選択して、グループ、仮想アドレス、リソースまたはフェイルオーバー構成に問題がないかどうかを確認し、問題があれば修正します。

## 15.3.2 汎用サービス用構成データ

可用性が高まるように汎用サービスを構成するには、汎用サービスをグループに追加します。Oracle Fail Safe により新規の汎用サービスを作成してグループに追加したり、既存の汎用サービスをグループに追加できます。いずれの場合も、Oracle Fail Safe Manager の「リソースをグループに追加」ウィザードを使用する際に次のデータが必要になります。

- 汎用サービスの識別情報（ノード名、表示名、サービス名、イメージ・パス）
- 汎用サービスの実行に使用されるアカウントおよびその起動パラメータ
- 汎用サービスにより使用されるディスク（ある場合）
- 汎用リソースが依存するその他のリソース
- 汎用サービスにより使用される Windows レジストリのキー値

他のリソースを可用性が高まるように構成する場合と異なり、汎用サービスを追加する前には仮想アドレスをグループに追加するように要求されません。汎用サービスの使用法に応じて、仮想アドレスが必要かどうかを自分で判断する必要があります。次の各項では、仮想アドレスをグループに追加する必要性を判断する際に考慮すべき問題と、汎用リソースをグループに追加するために必要な構成情報について検討します。

### 15.3.2.1 ノードの選択

汎用サービスをグループに追加する際に、クラスタが3つ以上のノードから構成されている場合は、[図 15-1](#) に示すように、選択済ノードのリストを指定して、汎用サービスの可能所有者となるノードを指定するように求められます。特定のノードを汎用サービスの可能所有者として指定しない場合は、そのノードを「選択済ノード」リストから選択して、左矢印をクリックします。

[2.6.6 項](#)では、「可能所有者ノード」リストの概念について詳細に説明します。

図 15-1 全ノードが使用可能な場合の「ノードの選択」ウィザード・ページ



汎用サービスをグループに追加する際に、クラスタが2つ以上のノードで構成されており、そのうちの1つ以上のノードが使用できない場合、どのノードを汎用サービスの可能所有者とするのかを指定するように求められます。このような場合、図 15-2 に示すように、ウィザード・ページには使用できないノードとその理由が表示されます。

図 15-2 使用できないノードがある場合の「ノードの選択」ウィザード・ページ



### 15.3.2.2 汎用サービスの識別情報

可用性が高まるように汎用サービスを構成する場合、Oracle Fail Safe がその汎用サービスの実行可能ファイルを一意に識別して見つける際に使用する基本情報を提供する必要があります。特に、「リソースをグループに追加」ウィザードでは、汎用サービスを識別するための次の情報が要求されます。

- ノード名

既存の汎用サービスの場合、汎用サービスが現在置かれているクラスタ・ノードを Oracle Fail Safe に認識させる必要があります。複数のノードに存在する場合は、そのうちの1つを「リソースをグループに追加」ウィザードで指定できます。サービスが存在しない場合は、汎用サービスの可能所有者である任意のノードを選択します。

- 表示名

表示名は、サービス名よりも詳しくサービスを説明するために使用します。表示名には、スペース文字、および最高 256 文字を指定できます。表示名は、Windows の「サービス」ダイアログ・ボックスに表示されます。

表示名は、Oracle Fail Safe が Oracle Fail Safe Manager のツリー・ビュー内のサービスを参照するために使用する名前でもあります。

- サービス名

サービス名は、短縮名と呼ばれることもあり、サービスの構成情報を含む Windows レジストリのサブキーにラベル付けするものです。サービス名にはスペース文字を含めることはできず、通常は表示名よりも短くなります。

- イメージ名

これは、汎用サービス実行可能ファイルのパスおよびファイル名です。汎用サービスの実行可能ファイルは、汎用サービスの可能所有者である全クラスタ・ノードのディレクトリおよびプライベート・ディスクにインストールする必要がありますので注意してください。これにより、汎用サービスがフェイルオーバーした場合も、サービスが依存する実行可能ファイルは、もう一方のクラスタ・ノードで使用可能になります。

汎用サービスの実行可能ファイルは、共有クラスタ・ディスクにはインストールしないでください。通常は、複数のサービス・インスタンスが同じ実行可能ファイルを使用します。実行可能ファイルが共有クラスタ・ディスクにインストールされている場合、その実行可能ファイルを使用するすべてのサービスが、現在そのディスクのホストになっているクラスタ・ノードで稼働する必要があります。

各クラスタ・ノードの同じ場所に実行可能ファイルをインストールすると、どちらのクラスタ・ノードも、同じ実行可能ファイルにアクセスする様々なサービス・インスタンスのホストになります。たとえば、Service\_A および Service\_B という 2 つのサービスがあり、どちらも同じ実行可能ファイルを使用するとします。実行可能ファイルが各クラスタ・ディスク上の同じ場所にインストールされている場合、Service\_A は、プライマリ・ノードが Node\_1 である Group\_A に属し、Service\_B は、プライマリ・ノードが Node\_2 である Group\_B に属することができます。Group\_C に属する共有クラスタ・ディスクに実行可能ファイルをインストールした場合、サービスは、その時点で Group\_C のホストになっているクラスタ・ノード上でしか稼働できません。

図 15-3 は、「リソースをグループに追加」ウィザードの、汎用サービスの識別情報を指定するページです。



図 15-3 「汎用サービスの識別情報」ウィザード・ページ

リソースをグループに追加 - 汎用サービスの識別情報 ステップ2/6

このグループに追加したい汎用サービスはどれですか？ 既存のサービスの名前から選択できます。または、一意な名前を入力することによりOracle Fail Safeで新しいサービスを作成できます。

ノード名

表示名

サービス名

状態

このサービスに対応付けられたイメージのフルパスは何ですか？

イメージ名

< 戻る(B)    次へ(N) >    キャンセル    ヘルプ

### 15.3.2.3 汎用サービスの起動パラメータ

「リソースをグループに追加」ウィザードでは、汎用サービスの起動方法について次の詳細情報が求められます。

- 起動パラメータ

Oracle Fail Safe から Windows サービス コントロール マネージャに渡される起動パラメータを指定します。これらのパラメータは、Windows の「サービス」ダイアログ・ボックスを使用した場合に指定するパラメータ（-t など）と同じです。Oracle Fail Safe により、パラメータがそのままサービス コントロール マネージャに渡されます。

- システム・アカウントまたはユーザー・アカウントとしてのログオン

サービスの実行に使用するアカウントを、システム・アカウントまたはユーザー・アカウントに指定します。デフォルトで、システム・アカウントを使用するログオンが選択されます。ユーザー・アカウントでログオンするには、「ログオン」で「このアカウント」を選択します。サービスの実行に使用されるアカウントによって、汎用サービスのセキュリティ・コンテキストが決まります。サービスがシステム・アカウント

(LocalSystem) としてログオンする場合、サービスは、ローカル・システム上のすべてのファイルにアクセスできますが、ネットワーク全体のファイルにはアクセスできません。サービスがユーザー・アカウントとしてログオンする場合、そのサービスが持つ権限に応じて、ローカル・システム上のファイルとネットワークにわたるファイルの両方にアクセスできます。たとえば、Oracle Fail Safe 自体は、両方のクラスタ・ノードのファイルにアクセスできる必要があるため、(Oracle Fail Safe をインストールしたときに指定した) ユーザー・アカウントで実行されます。

「リソースをグループに追加」ウィザードでは、既存のサービスの実行に使用されているアカウントを変更することはできません。既存の汎用サービスの実行に使用されるアカウントを変更する場合は、サービスをグループに追加する前に、Windows の「サービス」ダイアログ・ボックスを使用して変更します。(Oracle Fail Safe Server の実行に使用されるアカウント変更の詳細は、[4.3.1 項](#)を参照してください。)

Oracle Fail Safe では、汎用サービスの起動タイプの情報(「自動」、「手動」または「無効」)は要求されません。Oracle Fail Safe を使用して可用性が高まるよう構成されたリソースの起動タイプはすべて、「手動」に設定されます。クラスタ環境では、サービスは特定の時点で 1 つのノードでしか実行されません。起動タイプを「手動」に設定すると、Oracle Fail Safe では必ず、リソースは特定の時点で 1 つのノードのみで実行され、MSCS によってのみ起動されます。

[図 15-4](#) は、「リソースをグループに追加」ウィザードの、汎用サービスの起動パラメータとアカウントを指定するページです。

図 15-4 「汎用サービスのアカウント」ウィザード・ページ



リソースをグループに追加 - 汎用サービスのアカウント: ステップ 3/6

サービスを実行するときの起動パラメータはどれですか?

起動パラメータ

ログイン

☒ システム・アカウント

☐ このアカウント:

パスワード:

パスワードの確認

ドメイン

< 戻る(B)    次へ(N) >    キャンセル    ヘルプ

#### 15.3.2.4 汎用サービスにより使用されるディスク

Oracle Fail Safe では、その時点でサービスが実行されているクラスタ・ノード上で、可用性の高い汎用サービスに必要なデータ・ファイルが使用可能である必要があります。これには、次の 2 つの方法のいずれかを使用します。

- サービスに必要なデータ・ファイルを、リソースと同じグループに含まれている共有クラスタ・ディスク上に配置します。  
フェイルオーバー時には、サービスが引き続きファイルを使用できるように、ディスクはサービスとともにフェイルオーバーします。
- 可能所有者である全クラスタ・ノードの同じプライベート・ディスクおよびディレクトリに、同じファイルを配置します。

フェイルオーバー時には、サービスはプライベート・ディスク上で同じパスを使用してファイルを見つけます。ファイルのパスは各クラスタ・ノードで同じなので、どちらのクラスタ・ノードがリソースのホストであるかにかかわらず、リソースはファイルを見つけることができます。

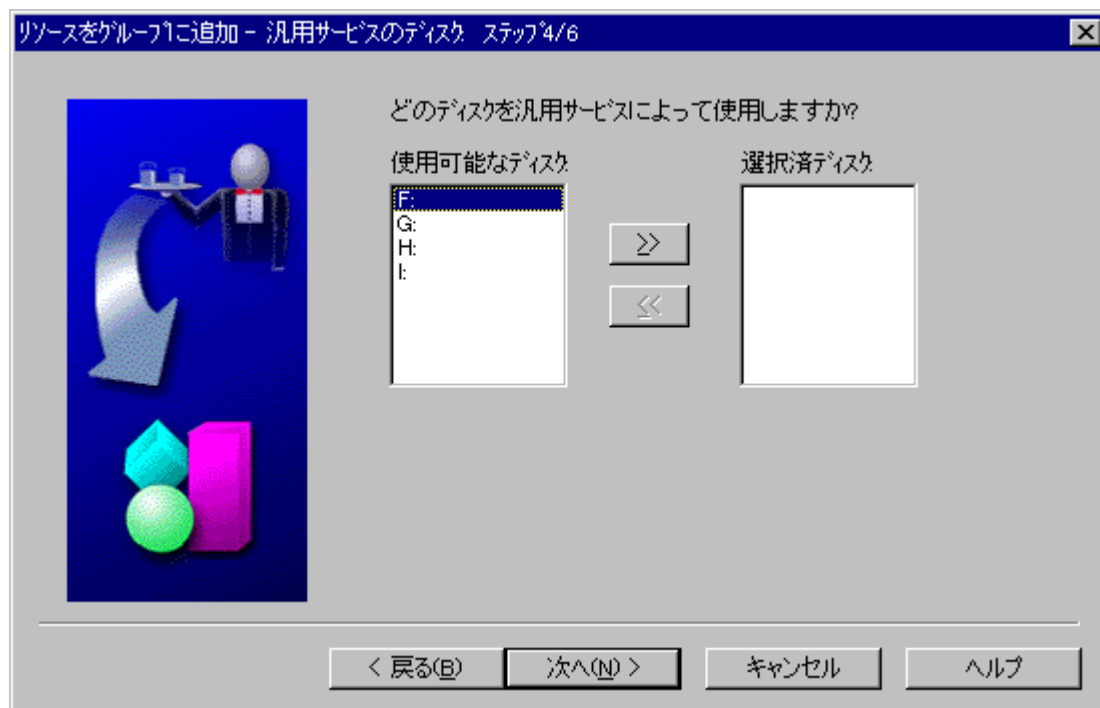
通常、サービス実行可能ファイルは各クラスタ・ノードのプライベート・ディスク上にインストールし、データ・ファイルは共有クラスタ・ディスク上に配置します。実行可能ファイルの配置場所の詳細は、[15.3.2.2 項](#)を参照してください。

汎用サービスが稼働しているクラスタ・ノードによってファイルの内容を変える場合は、各クラスタ・ノードの同じプライベート・ディスクおよびディレクトリに同じデータ・ファイルを配置します。たとえば、Node\_1 の CPU およびメモリーが Node\_2 の 2 倍であるとし、サービスを同時にアクセスできる最大ユーザー数を指定するファイルが汎用サービスに使用されている場合、その数値を Node\_1 では 100、Node\_2 では 50 に設定します。

ただし、可能な場合は、データ・ファイルを共有クラスタ・ディスク上に置いてください。このようにする場合は、「リソースをグループに追加」ウィザードを実行する前に、汎用リソースにより使用されるデータ・ファイルを共有クラスタ・ディスクに移動する必要があります。

[図 15-5](#) は、「リソースをグループに追加」ウィザードの、ディスク依存性を指定するページです。

図 15-5 「汎用サービスのディスク」ウィザード・ページ

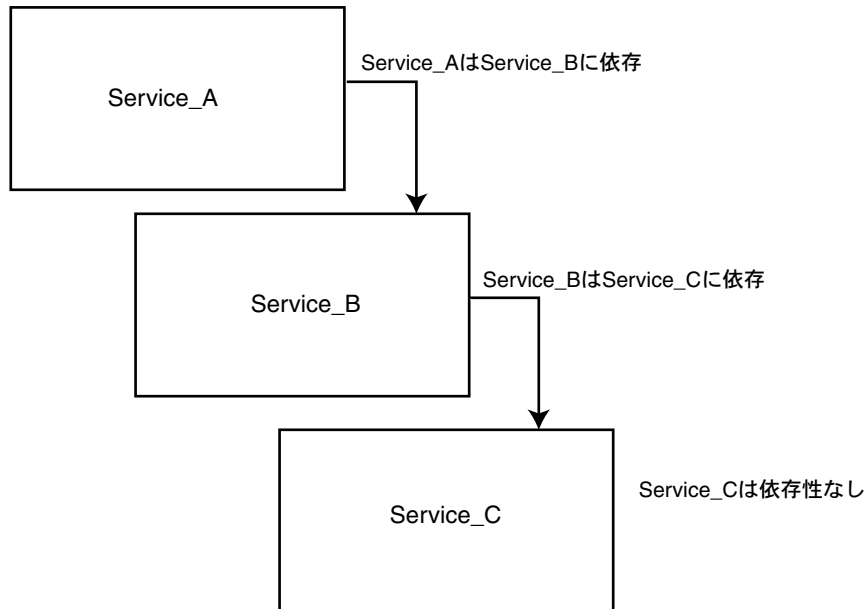


### 15.3.2.5 汎用サービスの依存性

構成するリソースについては Oracle Fail Safe Manager にも MSCS にも詳細情報が用意されていないため、Oracle Fail Safe が詳細情報を持っているリソース・タイプ（Oracle データベースなど）に比べ、可用性を高めるための汎用サービスの構成プロセスで自動的に行われる部分は少なくなります。たとえば、Oracle Fail Safe を使用して可用性の高い Oracle データベースを構成する場合は、Oracle Fail Safe Manager により、データベースが依存するリソースが追加され、それらのリソースをオンライン化する順序が決定されます。これに対し、汎用リソースを可用性が高まるように構成する場合は、この依存性情報を指定する必要があります。

**15.3.2.5.1 汎用サービスの依存性の指定** 汎用リソースの依存性情報の指定は、リソースをグループに追加する順序で行います。たとえば、Service\_A の可用性を高め、Service\_B と Service\_C がオンラインになっていないと Service\_A をオンライン化できないという構成にします。つまり、Service\_A は、Service\_B および Service\_C に依存しています。さらに、Service\_B を正常にオンライン化するには、Service\_C がすでにオンラインである必要があります。図 15-6 に示すように、この依存性の連鎖はツリー構造で図示できます。

図 15-6 依存性ツリー



このシナリオでは、最初に Service\_C をグループに追加します。次に Service\_B をグループに追加し、依存先として Service\_C を指定します。最後に Service\_A をグループに追加し、依存先として Service\_B を指定します。つまり、リソースごとに依存性ツリーを作成することになります。リソースをグループに追加するときに指定できる依存性レベルは1つのみです。このため、依存性ツリーの深さが2レベル（またはそれ以上）の場合、リソースをグループに追加する順序が重要になります。

図 15-7 は、「リソースをグループに追加」ウィザードの、リソース依存性を指定するページです。

図 15-7 「汎用サービスの依存性」ウィザード・ページ



**15.3.2.5.2 汎用サービスと仮想アドレスの依存性** Oracle Fail Safe では、汎用サービスをグループに追加する前に仮想アドレスをグループに追加するように要求されません。仮想アドレスは、クライアントまたは他のサービスがリソースを見つけるときに使用するネットワーク・アドレスを示します。クライアントも他のサービスも汎用サービスにアクセスしない場合は、汎用サービスを追加する前に仮想アドレスをグループに追加する必要はありません。

ただし、汎用サービスがクライアントまたは他のサービスによりアクセスされる場合は、グループに仮想アドレスを追加する必要性が生じる可能性があります。

### 15.3.2.6 汎用サービスのレジストリ・キー

汎用サービスで情報を格納するのに Windows レジストリのエントリを使用する場合は、「リソースをグループに追加」ウィザードでこれらのエントリを指定できます。ウィザードでエントリを指定すると、汎用サービスの可能所有者であるクラスタ・ノード間で、サービスの Windows レジストリ・エントリの一貫性が常に保たれます。これは、フェイルオーバー時に、汎用サービスの可能所有者である全クラスタ・ノードで、サービスが正常に稼働するために重要です。

たとえば、Oracle Forms Server を汎用サービスとして手動で構成する場合は、FORMS60\_PATH レジストリ変数を指定します。

指定するレジストリ・キーのルートは HKEY\_LOCAL\_MACHINE であることが前提とされます。この詳細は、「リソースをグループに追加」ウィザードのオンライン・ヘルプで説明されています。

図 15-8 は、「リソースをグループに追加」ウィザードの、Windows レジストリ・キーを指定するページです。

図 15-8 「汎用サービスのレジストリ」ウィザード・ページ



15.4 汎用サービスのセキュリティ要件

デフォルトでは、汎用サービスはローカル・システム・アカウントで実行されます。汎用サービスがユーザー・アカウントで実行されるように指定する場合、そのアカウントには「サービスとしてログオン」する権限が必要です。汎用サービスをグループに追加すると、Oracle Fail Safe により、汎用サービスの実行に使用されているアカウントにこの権限があるかどうか確認されます。権限がない場合、Oracle Fail Safe によって、指定されたユーザー・アカウントに権限が与えられます。

さらに、Oracle Fail Safe により、「リソースをグループに追加」ウィザードで指定したユーザー・アカウントとパスワードが有効かどうか確認されます。有効でない場合、エラーが返されます。

15.5 サンプル汎用サービスの構成

Oracle Fail Safe には、FsSampleService というサンプル汎用サービスが同梱されています。これを使用して、可用性の高い汎用サービスを構成するための手順およびその影響について理解を深めることができます。

表 15-2 では、Oracle Fail Safe を使用して可用性の高い汎用サービスを構成する方法の例として、FsSampleService を使用しています。

表 15-2 サンプル汎用サービスの構成手順

手順	処置	説明
1	(Oracle Fail Safe のインストール時に) 可用性の高い汎用サービスの可能所有者となる全クラスタ・ノードに対し Generic Service コンポーネントをインストール	Oracle Fail Safe の Generic Service コンポーネントのインストール時には、サンプル汎用サービスのイメージである FsSamplesvc.exe が、ディレクトリ <Oracle_Home>%fs%fs\$svr%\bin\FsSamplesvc.exe に置かれる。
2	Oracle Fail Safe Manager を使用してサンプル汎用サービスの可用性を向上	Oracle Fail Safe Manager で「リソースをグループに追加」ウィザードを使用して、汎用サービスをグループに追加する。この簡単なサンプルでは、ウィザードの最初のページに次のデータを入力するだけで済みます。  ノード名: 任意のクラスタ・ノードを選択します。 表示名: Fail Safe Sample Service サービス名: FsSampleService イメージ名: <Oracle_Home>%fs%fs\$svr%\bin\FsSamplesvc.exe  「リソースをグループに追加」ウィザードでは、他ににも入力する必要ありません。ウィザードの残りのページで「次へ」をクリックして、最後のページで「完了」をクリックします。



可用性が高まるようにサンプル汎用サービスを構成した後で、クラスタ環境でどのように動作するかを次の手順でテストできます。

1. 汎用サービスの可能所有者である各ノードのコマンド・ウィンドウを開きます。
2. 各コマンド・ウィンドウで、「fssvcclient」と入力します。「FailSafe Sample Service is not running」または「FailSafe Sample Service is running」というメッセージが返されます。
3. サンプル汎用サービスを含むグループを新しいノードに移動します。
4. 各ノードのコマンド・ウィンドウで、fssvcclient コマンドを再発行します。現在のノードのサービスが停止され、サービスを含んだグループの移動先のノードが稼働を開始します。

## 15.6 汎用サービスに関する問題のトラブルシューティング

次の項では、可用性が高まるように構成した汎用サービスを使用する際に発生する具体的な問題のトラブルシューティングについて説明します。Oracle Fail Safe のトラブルシューティング・ツール（「クラスタの検証」および「グループの検証」）に関する一般情報は、[第 6 章](#)に記載しています。

「グループの検証」操作は、いつでも実行できます。ただし、次のような場合には必ず実行します。

- グループまたはグループ内のリソースがオンライン化されない場合
- フェイルオーバーまたはフェイルバックが予定どおりに実行されない場合
- クラスタに新しいノードが追加された場合
- クラスタ・ノードから誤って汎用サービスを削除した場合

「グループの検証」操作を実行すると、汎用サービスの可能所有者である全クラスタ・ノード上に同じ汎用サービスが再作成されます。



# 第 IV 部

---

## 付録、用語集および索引

第 IV 部には、いくつかの付録、用語集および索引があります。

第 IV 部には、次の付録が含まれています。

- [付録 A「ODBC および OCI のデモ」](#)
- [付録 B「ネットワーク構成の要件」](#)
- [付録 C「オラクル社カスタマ・サポート・センターへの連絡」](#)



---

## ODBC および OCI のデモ

Oracle Fail Safe 環境では、クライアント・アプリケーションはフェイルオーバー後も引き続き機能しますが、透過的なフェイルオーバーを使用するようにアプリケーションを設計およびコーディングしていない限り、アプリケーションは再接続する必要があります。フェイルオーバーの発生後に、フェイルオーバー・ノードに自動的に再接続するようアプリケーションを設計できます。Oracle8 以降の OCI ライブラリを使用するクライアント・アプリケーションでは、透過的なアプリケーション・フェイルオーバー手法を使用して、クラスタのフェイルオーバー後に OCI または ODBC データベースに自動的に再接続し、処理を再開することができます。

ODBC および OCI クライアント・アプリケーションで透過的なアプリケーション・フェイルオーバーを実装する OCI コールバック関数の使用方法は、[7.10 項](#)に説明されています。

Oracle Fail Safe ソフトウェアに含まれている FSFAILOVERGUI.EXE 実行可能プログラムは、OCI または ODBC プログラムの自動フェイルオーバーを示したデモ・プログラムです。このプログラムには、OCI および ODBC デモを実行する手順と、TNSNAMES.ORA ファイルを更新する手順を示す readme.txt ファイルも含まれています。

このデモでは、Oracle（OCI または ODBC）データベースに対する問合せの実行から開始し、次に（Oracle Fail Safe Manager を使用して）問合せの実行中にデータベースを含むグループを移動します。アプリケーションが新規ノードに再接続し、SELECT 文を再実行している間、一時的に問合せ処理が中断することがわかります。

デモ・プログラムと readme.txt 手順ファイルは、Oracle Fail Safe をインストールした次の Oracle ホームにあります。

```
<Oracle_Home>%fs%fsmgr%sample
```



## ネットワーク構成の要件

この付録では、Microsoft Windows ソフトウェアのネットワーク要件について説明します。  
この付録では、次の項目について説明します。

項目	参照
<a href="#">ホスト名および IP アドレスの登録</a>	<a href="#">B.1 項</a>
<a href="#">クラスタ内の正しい名前解決の有効性検査</a>	<a href="#">B.2 項</a>
<a href="#">不適切な名前解決に関する問題のトラブルシューティング</a>	<a href="#">B.3 項</a>

### B.1 ホスト名および IP アドレスの登録

各 IP アドレスとそれに対応するホスト名は、WINS を使用するかどうかに関係なく、各クラスタ・ノード上の（システム・ディレクトリの system32¥drivers¥etc¥ ディレクトリにある）Hosts ファイルまたは DNS に登録する必要があります。次のアドレスをすべて登録する必要があります。

- クラスタ・ノードのアドレス
- クラスタ別名のアドレス
- 仮想サーバー・アドレス
- Oracle Fail Safe Manager のクライアント・システム

名前レジストリ（DNS サーバーまたは Hosts ファイル）のいずれかに変更を加えた場合は、次のコマンドを使用して、ローカル・キャッシュを削除し、変更を有効にします。

```
nbtstat -R
```

B.2 クラスタ内の正しい名前解決の有効性検査

この項では、全システムを ping して、IP アドレスとホスト名が正しく登録されていることを確認する方法を説明します。この項で説明する手順を実行することによって、ホスト自体からホスト名を ping した場合も、他の任意のシステムから ping した場合も、同一の IP アドレスが返されます。

2 つのシステムから構成されるクラスタと 1 つのクライアント・システムがあるとします。これらのシステムとクラスタ別名のホスト名を次の表に示します。

システム	ホスト名
クラスタ・システム 1	ClusterHost1
クラスタ・システム 2	ClusterHost2
クラスタ別名	ClusterAlias
クライアント・システム	ClientHost

ClusterHost1、ClusterHost2 および ClientHost の各システム上で、4 つのホスト名をすべて ping します。たとえば、ClusterHost1 で次を実行します。

```
ping ClusterHost1
ping ClusterHost2
ping ClusterAlias
ping ClientHost
```

この ping テストを、ClusterHost2 と ClientHost 上で繰り返します。

ping する各ホスト名に対して、ping テストでは同一 IP アドレスが返されます。たとえば、ClusterHost1 が IP アドレス 192.1.99.202 にマップするように登録されている場合、ClusterHost1、ClusterHost2 および ClientHost 上での ClusterHost1 の ping により、192.1.99.202 が返されます。

ping テストがすべて正常に終了した場合、ネットワークは正しく構成されています。

ping テストのいずれかで正しい IP アドレスが返されなかった場合は、[B.3 項](#)を参照してください。



## B.3 不適切な名前解決に関する問題のトラブルシューティング

B.2 項での ping テストで正しいアドレスが返されない場合は、次のいずれかが原因です。

- IP アドレスまたはホスト名（あるいはその両方）を Hosts ファイルまたは DNS サーバーに誤って登録。

これが問題であると思われる場合は、B.1 項を再度調べて、アドレスが正しく登録されていることを確認するか、またはネットワーク管理者に連絡して IP アドレスとホスト名を調べます。

- 複数のネットワーク・インタフェース・カードが存在する。

複数のネットワーク・インタフェース・カードの存在するシステムでは、複数の IP アドレスがホスト名に割り当てられます。たとえば、システム ClusterHost1 にはネットワーク・インタフェース・カードが 2 つ存在したとします。それぞれのカードには、次のように、異なるネットワークまたはサブネット上の IP アドレスが割り当てられます。

ネットワーク	IP アドレス
Net1	192.1.22.101
Net2	192.1.99.202

システム ClusterHost1 では、ping プログラムにより IP アドレスが 1 つ返されますが、ホスト名 ClusterHost1 は両方の IP アドレスにマップされています。ping プログラムにより、他のシステム上で返される IP アドレスと同じ IP アドレスが返されない場合、ホスト名の IP アドレスの順序に問題がある可能性があります。

順序の問題を訂正するには、使用している Windows オペレーティング・システムに応じて、次のいずれかの項の手順を実行してください。

### B.3.1 Windows NT Service Pack 3 に推奨されるソリューション

次の手順を実行して、（ホスト名ではなく）IP アドレスをリスニングするように、Oracle リスナーと Oracle Intelligent Agent を構成します。

1. システム上の Net8（または SQL\*Net）構成ディレクトリの場所を検出します。
  - Oracle7 または Oracle8i コンポーネントが含まれたシステムで、`<Oracle_Home>%network%admin` 内の構成ファイルを変更します。
  - Oracle8 コンポーネントとともに稼働しているシステムで、両方の `<Oracle_Home>%net80%admin` ディレクトリ内の構成ファイルを変更します。

2. LISTENER.ORA ファイルを変更します。

上記の各ディレクトリで、ファイル LISTENER.ORA を編集します。ファイル内のシステムのノード名をすべてネットワーク別名またはネットワーク IP アドレスに変更します。

たとえば、システムのホスト名が `server_name`、既知の IP アドレスが 198.2.38.1 であるとします。このファイルの元の内容は、次のとおりです。

```
(ADDRESS=
 (PROTOCOL=TCP)
 (Host=server_name)
 (Port=1521)
)

(ADDRESS=
 (PROTOCOL=TCP)
 (Host=198.2.38.1)
 (Port=1521)
)
```

3. TNSNAMES.ORA ファイルを変更します。

手順 1 の各ディレクトリで、TNSNAMES.ORA ファイルを編集します。  
LISTENER.ORA で行った変更を、同じように TNSNAMES.ORA ファイルで行います。

4. SNMP\_RW.ORA ファイルを変更します。

事前に決められたポート番号で Oracle Intelligent Agent がシステムのホスト名をリスニングする場合、SNMP\_RW.ORA ファイルにはリスニング・パラメータがまったく指定されません。この例で、リスニング・パラメータを指定するために、次の 2 行を SNMP\_RW.ORA ファイルに追加します。

```
dbsnmp.address= (ADDRESS= (PROTOCOL=TCP) (HOST=198.2.38.1) (PORT=1748))
dbsnmp.spawnaddress= (ADDRESS= (PROTOCOL=TCP) (HOST=198.2.38.1) (PORT=1754))
```

ポート番号は、Oracle Intelligent Agent で使用する事前定義のポート番号です。その他のポート番号を使用する必要はありません。

---

---

**注意：** システムに Oracle データベースがないと、Oracle Intelligent Agent は起動しません。

---

---

### B.3.2 Windows NT Service Pack 4 以降に推奨されるソリューション

Windows NT Service Pack 4 またはそれ以降を実行しているシステムでは、ホスト名に既知のアドレスが返されるように、IP アドレスをネットワーク名にマップする順序を変更できます。マッピング順序を変更するには、次のようにします。

1. Windows の「コントロール パネル」を開きます。
2. 「ネットワーク」をダブルクリックします。
3. 「バインド」タブを選択します。
4. 「バインドの表示 - すべてのプロトコル」を選択します。
5. 「TCP/IP プロトコル」フォルダを拡張します。
6. 移動するアダプタを選択し、「上へ」または「下へ」をクリックして順序を変更します。
7. 「OK」をクリックします。

IP アドレスがホスト名の既知のアドレスにマップするような順序を選択することが目的です。変更が完了した後、[B.2 項](#)に説明されたプロセスを使用して、新しい構成が正しいかどうかを確認します。

### B.3.3 Windows 2000 に推奨されるソリューション

Windows 2000 Advanced Server または Datacenter Server を実行しているシステムでは、ホスト名に既知のアドレスが返されるように、IP アドレスをネットワーク名にマップする順序を変更できます。マッピング順序を変更するには、次のようにします。

1. Windows の「コントロール パネル」を開きます。
2. 「ネットワークとダイヤルアップ接続」をダブルクリックします。
3. 「バインド」タブを選択します。
4. 「詳細設定」メニューで「詳細設定」を選択します。
5. 「アダプタとバインド」タブを選択します。
6. 「接続」ボックスで、移動するアダプタを選択し、上矢印または下矢印をクリックして順序を変更します。
7. 「OK」をクリックします。

変更が完了した後、[B.2 項](#)に説明されたプロセスを使用して、新しい構成が正しいかどうかを確認します。



---

# オラクル社カスタマ・サポート・センターへの連絡

---

この付録では、次の項目について説明します。

項目	参照
<a href="#">問題の報告</a>	C.1 項
<a href="#">バージョン情報の検出</a>	C.2 項
<a href="#">Oracle Fail Safe の問題のトレース</a>	C.3 項
<a href="#">トレース・ファイルとアラート・ファイルの検索</a>	C.4 項

## C.1 問題の報告

メッセージの中には、オラクル社に連絡して問題を報告するようにお薦めしているものがあります。オラクル社カスタマ・サポート・センターにご連絡いただく場合は、次のような情報を用意しておいてください。

- Oracle ソフトウェアが動作しているハードウェア、ならびにオペレーティング・システムとそのリリース番号。
- Oracle を含む他のソフトウェア製品の完全なリリース番号（たとえば、Oracle Fail Safe 3.1.1 など）。
- エラーが発生したときに使用していたすべての Oracle プログラム（およびそのリリース番号）（たとえば、SQL\*Net リリース 2.3 など）。
- エラー・コードやエラー・メッセージが 1 つまたは複数表示された場合は、それらの正確なコード番号とメッセージ・テキスト（表示順に）。

- 問題の重大度。次のコードに従って判断してください。
  - 1 = プログラムは使用不能。操作に致命的な影響を与えます。
  - 2 = プログラムは使用可能。操作は非常に制限されています。
  - 3 = 機能は限られているが、プログラムは使用可能。操作全体にそれほど重大な影響はありません。
  - 4 = ユーザーにより、プログラムで問題を回避。操作への影響はきわめて微小です。
- ユーザーの個人の情報と会社の情報
  - 名前
  - 会社名
  - 会社の Oracle Support ID 番号
  - 電話番号
- 場合によっては、オラクル社カスタマ・サポート・センターよりトレース・ファイルの提供をお願いすることがあります。

エラー出力をファイルに記録するトレース機能の使用方法は、[C.3 項](#)を参照してください。

## C.2 バージョン情報の検出

実行中のソフトウェアのバージョンは、Oracle Fail Safe Manager のヘルプ・メニューで確認できます。次のように選択します。「ヘルプ」→「バージョン情報」

## C.3 Oracle Fail Safe の問題のトレース

トレース機能を使用すると、エラー情報がログ・ファイルにダンプされます。この機能は Oracle Fail Safe で受け取るエラーの追跡、報告、検査に役立ちます。

トレース機能はノード単位で有効にします。次の手順に従い、クラスタのサーバー・ノード上でトレース機能を有効にし、トレース・フラグを設定してください。

1. Windows レジストリ・エディタを起動します。
2. レジストリのツリー表示から、次の順序で選択します。

「HKEY\_LOCAL\_MACHINE」→「SOFTWARE」→「ORACLE」→「FailSafe」→「Tracing」
3. レジストリ・エディタのメニュー・バーから「編集」を選択し、「値の追加」を選択して「文字列の追加」ダイアログ・ボックスを開きます。
4. 「値の名前」フィールドに、[表 C-1](#) に示された Oracle Fail Safe Server 値を入力します。
5. 「データタイプ」フィールドに REG\_SZ と入力します。

6. 「OK」をクリックして「文字列エディタ」ダイアログ・ボックスを開きます。
7. 「文字列」フィールドで、表 C-1 に示されている Oracle Fail Server 文字列を 1 つ以上入力します。複数の項目がある場合は、カンマで区切ってください。
8. 手順 3 ～ 7 を繰り返し実行して、Oracle Fail Safe Server トレース・フラグをさらに設定します。
9. クラスタ アドミニストレータを使用して Oracle Fail Safe Server をシャットダウンします。
10. Windows の「コントロール パネル」で「サービス」を選択し、すべてのクラスタ・ノード上で Oracle Fail Safe Server を停止します。
11. クラスタ アドミニストレータを使用して Oracle Fail Safe Server を再起動し、トレースを開始します。

すべてのクラスタ・ノード上でこれらの手順を繰り返し、クラスタ全体でトレース機能を必ず有効にしてください。

**表 C-1 クラスタのサーバー・ノードが使用するトレース・フラグ**

値	文字列	説明
FSR_TRACE_OUTPUT	パスおよびファイル名	Oracle Fail Safe リソース DLL に関するトレース情報を書き込むファイルのパスおよびファイル名を指定します。たとえば、次のようになります。C:\FSR_TRACE.LOG
FSS_TRACE_OUTPUT	パスおよびファイル名	Oracle Fail Safe Server に関するトレース情報を書き込むファイルのパスおよびファイル名を指定します。たとえば、次のようになります。C:\FSS_TRACE.LOG

表 C-1 クラスタのサーバー・ノードが使用するトレース・フラグ（続き）

値	文字列	説明
FSS_TRACE_FLAGS	CLUSTER_MGR	クラスタ管理ソフトウェアに対して実行されたコールに関する情報を記録します。
	CR_DBRES	「リソースをグループに追加」ウィザードの実行時にデータベース・リソースを作成するときに、ログ情報をとります。
	CR_SAMPLE	「サンプル・データベースの作成」操作に関するトレース情報を獲得します。この情報には、手順ごとの開始と停止の項目（タイムスタンプあり）が含まれています。
	DB_RES	Oracle Fail Safe リソース DLL がデータベースにアクセスするときに、ログ情報をとります。
	DEL_SAMPLE	「サンプル・データベースの削除」操作に関するログ情報をとります。CR_SAMPLE 操作と似ています。
	GR_VERIFY	「グループの検証」操作に関するログ情報をとります。
	SQLNET	Oracle Fail Safe Server が実行した Net8（または SQL*Net）構成に関連する詳しい内部情報を生成します。Net8（または SQL*Net）構成の変更が必要な操作が実行されるたびに、ログ情報がとられます。これには、サンプル・データベースの作成や削除、あるいはグループに対するデータベースの追加や削除が含まれます。
	VERIFY_DB	「スタンドアロン・データベースの検証」操作に関するログ情報をとります。

C.4 トレース・ファイルとアラート・ファイルの検索

トレース・ファイルとアラート・ファイルは、クラスタ・ディスクかプライベート・ディスクにあります。

- クラスタ・ディスクを使用する場合は、トレース・ファイルおよびアラート・ファイルには、操作に関する完全な情報が含まれます。ただし、データベースをホストするノードに関する情報は記録されません。これらのファイルで使用されるクラスタ・ディスクは、アーカイブ・ログ・ファイルまたはデータベース・データ・ファイルで 사용되는ディスク（たとえば、「サンプル・データベースの作成」操作によりファイルが配置されるディスク）である必要があります。
- プライベート・ディスクを使用する場合は、トレース・ファイルおよびアラート・ファイルには、操作に関するノード固有の情報がそれぞれ含まれます。ただし、データベースがフェイルオーバーまたは削除された場合に完全な履歴情報を取得するには、すべてのノードからファイルを表示する必要があることもあります。データがこれらのファイルに正しく書き込まれるように、それぞれのノードで有効なパス名を使用します。プライベート・ディスク上のファイルは、グループに追加されることはありません。



---

# 用語集

## 24 × 365

1 日 24 時間、1 年 365 日のビジネス体制。

## 2 次ノード (secondary node)

アクティブ / パッシブ構成において、フェイルオーバーの際にプライマリ・ノードの作業を受け継ぐためにスタンバイしているノード。「[アクティブ / パッシブ構成](#)」および「[プライマリ・ノード](#)」も参照。

## Bequeath プロトコル (bequeath protocol)

クライアントが、ネットワーク・リスナーを使用せずに Oracle データベースから情報を取り出すことのできるプロトコル。Bequeath プロトコルによって、各クライアント・アプリケーション用のサーバー・スレッドが内部的に生成される。ある意味では、データベース接続に対してリモート・ネットワーク・リスナーが行っているのと同じ操作を、ローカルに実行するものである。

## Concurrent Manager

Oracle Applications のプロセス・マネージャ。データ量の多い様々なプログラムの実行をユーザーが要求することによって生成されるプロセスを調整する。Oracle Applications 製品グループでは、複数の Concurrent Manager を持つことができる。「[内部 Concurrent Manager](#)」も参照。

## IP アドレス (IP address)

インターネット・プロトコル (IP) ・アドレス。IP アドレスは、n.n.n.n という形式になる。たとえば、138.2.134.113。

## Microsoft Cluster Server (MSCS)

Windows NT Enterprise Edition、Windows 2000 Advanced Server または Windows 2000 Datacenter が稼働している個々のノードをクラスタ化する Microsoft 社のソフトウェア。

### **アクティブ/アクティブ構成 (active/active configuration)**

すべてのクラスタ・ノードが処理を実行するクラスタ構成。片方のノードが使用できなくなった場合、使用できないノードの作業負荷を、1つ以上の他のノードが引き継ぐ。

### **アクティブ/パッシブ構成 (active/passive configuration)**

1つのノードが、他のノードからのフェイルオーバーに備えて、通常はアイドル状態で待機するクラスタ構成。

### **仮想アドレス (virtual address)**

グループ内のリソースのホストになっているクラスタ・ノードとは無関係に、そのリソースにアクセスできるネットワーク・アドレス。MSCS クラスタ上の仮想アドレスは、ネットワーク名とそれに関連付けられた IP アドレスで構成される。

### **仮想サーバー (virtual server)**

1つ以上の仮想アドレスを持つグループ。

### **仮想ディレクトリ (virtual directory)**

物理ディレクトリ指定にマップされる名前。仮想ディレクトリは、ファイル構造をユーザーから隠すために指定する。物理ディレクトリが変更された場合でも、ユーザーが指定する URL は変更されない。たとえば、仮想アドレスが「Company」で、仮想ディレクトリ「Sales」を U:\SalesInfo\Webfiles にマップしてある場合、ユーザーは `http://Company/Sales` という URL を入力して販売情報にアクセスする。

### **可能所有者ノード (possible owner ode)**

次の性質に基づいて、指定されたリソースを実行できるノード。

- 指定されたリソースのリソース .dll がノードにインストールされていること
- そのリソースがノード上で実行するよう構成されていること
- そのリソースまたはリソースを含むグループの「可能所有者ノード」リストから、手動でノードを削除していないこと

2つのノードから成るクラスタで、グループがフェイルオーバーできるようにする場合には、そのグループ内のすべてのリソースに対して両方のノードが可能所有者ノードである必要がある。

### **「可能所有者ノード」リスト (possible owner nodes list)**

指定されたリソースの .dll がインストールされ、実行するよう構成されているノードのうち、明示的にセットから削除したノードを除いたすべてのセット。

### 可用性 (availability)

システムまたはリソースが、必要なときに必要なサービスを提供する能力の程度。可用性は、装置が必要とされる合計時間のうち、アクセス可能である時間の割合として測定される。連続したコンピューティング・サービスを必要とするビジネスでは、可用性の目標は 24 × 365。

### 共有記憶装置インターコネクト (shared storage interconnect)

クラスタ・ディスクがクラスタ内のすべてのノードからアクセス可能な I/O 接続。

### クオラム (quorum)

リカバリに必要な特定のデータが全クラスタ・メンバー間で一貫してメンテナンスできることを保証するために使用される選択メカニズム。このメカニズムには、[クオラム・リソース](#)と呼ばれる特殊な記憶装置リソースが関係する。クオラムは、クラスタを確立する際にも使用される。「[クオラム・リソース](#)」も参照。

### クオラム・リソース (quorum resource)

クラスタのリカバリに必要な構成データをメンテナンスするために選択されたクオラム可能な記憶装置リソース。構成データに対する最新の変更をどのクラスタ・ノードからでもアクセスできるように、クオラム・リソースは、通常、他のクラスタ・リソースからアクセス可能になる。「[クオラム](#)」も参照。

### クライアント・アプリケーション (client application)

すべてのユーザー指向のアクティビティを提供するアプリケーション。アクティビティの中には、文字またはグラフィカル・ユーザー表示、画面制御、データの提示、アプリケーションの流れ、およびその他のアプリケーション固有のタスクが含まれる。

### クラスタ (cluster)

単一の仮想システムとして動作する、2 ～ 4 基の独立したコンピューティング・システムのグループ。

### クラスタ対応 (cluster-aware)

フェイルオーバーの発生時に、正常に機能しているクラスタ・ノードに自動的に再接続するよう設計されているアプリケーション。このクラスタ対応のアプリケーションは、正常なノードとの再接続や、障害発生時に損失の可能性があるトランザクションの再実行に関する詳細をすべて処理する。

### クラスタ・ノード (cluster node)

クラスタのメンバーである Windows システム。

### クラスタ別名 (cluster alias)

クラスタを識別し、クラスタ関連のシステム管理に使用される、ノードから独立したネットワーク名。

### クラスタ・リソース (cluster resource)

クラスタ・ノード上で構成および管理されるリソース。「リソース」および「スタンドアロン・リソース」も参照。

### グループ (group)

フェイルオーバーの最小単位を形成するクラスタ・リソースの論理的な集まり。フェイルオーバーの際、リソースのグループはまとめてフェイルオーバー・ノードに移される。グループがある特定の時点で常駐するクラスタ・ノードは1つである。

### グループ・フェイルオーバー・ポリシー (group failover policy)

次の2つのパラメータを決定するユーザー指定の計画。クラスタ・ソフトウェアがリソースを片方のノードからもう一方のノードに移動し続ける期間（フェイルオーバー期間）と、フェイルオーバー期間中に発生するフェイルオーバー回数の最大値（フェイルオーバーしきい値）。「フェイルオーバー期間」および「フェイルオーバーしきい値」も参照。

### 計画外ダウン時間 (unplanned downtime)

予定外のシステムの中断。突然の停電、自然災害、業務に影響のある予測不可能なハードウェアまたはソフトウェア障害など。「ダウン時間」および「計画的ダウン時間」も参照。

### 計画外フェイルオーバー (unplanned failover)

ソフトウェアまたはハードウェア障害に対応して自動的にトリガーされる、ソフトウェアから起動されるフェイルオーバー・プロセス。「計画的フェイルオーバー」および「計画外ダウン時間」も参照。

### 計画的ダウン時間 (planned downtime)

日常的なメンテナンス作業を実行するための時間間隔。このような作業にはハードウェアやソフトウェアのアップグレード、付加価値サービスなどがある。こうした予定されたメンテナンス作業中は、コンピューティング・システムを生産的に運用することができない。「計画外ダウン時間」および「ダウン時間」も参照。

### 計画的フェイルオーバー (planned failover)

クライアント・アプリケーションおよびクラスタ・リソースを片方のノードで意図的にオフラインにし、もう一方のノードでオンラインにする処理。たとえば、『Oracle Fail Safe Installation Guide』には、計画的フェイルオーバーを実行してローリング・アップグレード（片方のノード上でハードウェアまたはソフトウェアを順番にアップグレードする際に、全リソースを別のクラスタ・ノードにフェイルオーバーすること）を実行する方法が説明されている。「計画外フェイルオーバー」も参照。

### サービス名エントリ (service name entry)

「ネット・サービス名」を参照。

**災害トレラント (disaster tolerance)**

システム・コンポーネント内の障害の影響を受けず（フォールト・トレラント）、さらに地震、火災、洪水、暴風、停電、破壊行為、テロ行為などの災害にも持ちこたえられるコンピューティング・システムの能力。

**サイレント・モード (silent mode)**

ソフトウェアをインストールする際に、レスポンス・ファイルを使用して Oracle Universal Installer への入力を指定できるインストール方法。

**作業負荷のパーティション化 (partitioned workload)**

クラスタの各ノードが、別々の作業を実行する構成。たとえば、2つのノードから成るクラスタで、一方のノードがデータベース・サーバーとして機能し、もう一方のノードがアプリケーション・サーバーとして機能する場合など。

**サブネット・マスク (subnet mask)**

ネットワーク ID 用としてアドレス内のビットがいくつ使用されるかを示す 32 ビット値。

**サンプル・データベース (sample database)**

Oracle Fail Safe の機能を本番データベースで使用する前に試せるように、Oracle Fail Safe で提供されている事前に構成されたオプションの初期データベース。

**シェアード・ナッシング (shared nothing)**

すべてのクラスタ・ノードが物理的に同一ディスクにケーブル接続されているが、ある時点で特定のディスクにアクセスできるのは、読み込みおよび書き込みのいずれの場合も 1 つのノードのみであるクラスタ構成。

**障害 (failure)**

コンピューティング・コンポーネントがその機能を正しく実行できなくなること。

**障害発生箇所 (single point of failure)**

コンピューティング・システムの単一のコンポーネントで、ここに障害が生じるとシステム全体がサービスの提供を停止するようなコンポーネント。

**冗長 (redundant)**

コンピューティング・システムの整合性を保護する、重複した、または余分のコンピューティング・コンポーネント。

**信頼性 (reliability)**

コンピューティング・システムが障害を起こさずに動作する能力。

**スケーラビリティ (scalability)**

システムへの負荷の増大に対応して拡張できるシステムの能力。

### スタンドアロン・リソース (standalone resource)

グループに含まれていないリソース。スタンドアロン・リソースは、特定のクラスタ・ノードによりホストされる。「[クラスタ・リソース](#)」および「[グループ](#)」も参照。

### スタンバイ・ノード (standby node)

優先ノードに障害が発生した場合にアプリケーション処理をいつでも継続できるように準備された、アクティブ / パッシブ・アーキテクチャ内のノード。「[アクティブ / パッシブ構成](#)」および「[優先ノード](#)」も参照。

### ダウン時間 (downtime)

システムまたはリソースで、必要時に必要なサービスの提供が不可能な割合。ダウン時間は、装置が必要とされる合計時間のうち、アクセス不能である時間の割合として測定される。「[計画的ダウン時間](#)」および「[計画外ダウン時間](#)」も参照。

### 透過的アプリケーション・フェイルオーバー (transparent application failover)

フェイルオーバーの発生後、クライアント・アプリケーションが自動的にデータベースに再接続し操作を再開する能力。

### 内部 Concurrent Manager (internal concurrent manager)

プログラム実施要求の監視と、他の Concurrent Manager の制御を行い、要求をいつ処理するか、およびどの Concurrent Manager でそれを実行するかを決定するプロセス・マネージャ。Oracle Fail Safe を使用して可用性が高まるように構成するのは、この内部 Concurrent Manager である。「[Concurrent Manager](#)」も参照。

### ネット・サービス名 (net service name)

ネットワークと Oracle データベースの接続データを記述するネットワーク情報。1つの Oracle データベースに対して複数のネット・サービス名を定義できる。Oracle8i より前では、ネット・サービス名はサービス名エントリと称した。

### ネットワーク名 (network name)

Microsoft Cluster Server (MSCS) で使用される用語で NetBIOS 名のこと。ネットワーク上の特定の IP アドレスに変換される名前。「[ホスト名](#)」も参照。

### ノード (node)

クラスタのメンバーであるコンピューティング・システム。

### ノード間ネットワーク接続 (internode network connection)

「[プライベート・インターコネクト](#)」を参照。

### ハートビート接続 (heartbeat connection)

「[プライベート・インターコネクト](#)」を参照。

### **パブリック・インターコネクト (public interconnect)**

クライアントをクラスタに接続するネットワーク接続 (LAN や WAN など)。「[プライベート・インターコネクト](#)」も参照。

### **汎用サービス (generic service)**

Microsoft Cluster Server (MSCS) で提供される汎用サービス・リソース DLL によりサポートされる Windows サービス。汎用サービス・リソース DLL は、標準の Windows サービス (IP アドレス、物理ディスクおよび一部のアプリケーションなど) をクラスタ内のリソースとして構成するために使用される。

### **フェイルオーバー (failover)**

クラスタ・リソースを片方のノードでオフラインにし、そのリソースをもう一方のノードでオンラインにするプロセス。このプロセスは、計画的 (アップグレードおよびメンテナンスの際など) または計画外 (システム障害やリソース障害の場合など) のいずれかにできる。

### **フェイルオーバー期間 (failover period)**

クラスタ・ソフトウェアがフェイルオーバー・プロセスを停止してクラスタ・リソースをオフラインにする前に、クラスタ・リソースを片方のノードからもう一方のノードに移動し続けるユーザー指定の期間。「[グループ・フェイルオーバー・ポリシー](#)」も参照。

### **フェイルオーバーしきい値 (failover threshold)**

クラスタ・ソフトウェアが、指定された期間 (フェイルオーバー期間) 内にリソースを片方のノードからもう一方のノードに移動しようとする回数の最大値。クラスタ・ソフトウェアは、指定されたフェイルオーバーしきい値に達すると、フェイルオーバー・プロセスを停止し、リソースをオフラインにする。「[グループ・フェイルオーバー・ポリシー](#)」も参照。

### **フェイルオーバー・ノード (failover node)**

使用不能ノードの作業負荷を引き継ぐサーバー・ノード。

### **フェイルオーバー・ポリシー (failover policy)**

「[グループ・フェイルオーバー・ポリシー](#)」または「[リソース・フェイルオーバー・ポリシー](#)」を参照。

### **フェイルセーフ・リソース (fail-safe resource)**

可用性が高まるように構成されたリソース。

### **フェイルバック (failback)**

優先ノードが動作可能状態にリカバリした後、クラスタ・リソースのグループをフェイルオーバー・ノードから優先ノードへ意図的に戻す処理。

### **フェイルバック・ポリシー (failback policy)**

クラスタ・リソースをフェイルオーバー・ノードから優先ノードへフェイルバックする必要があるか、およびいつフェイルバックするかを決定するユーザー指定の計画。

### **フォールト・トレラント (fault tolerance)**

コンピューティング・システムがコンポーネント内の障害やエラー（ハードウェア障害、ファイルの破壊またはオペレーティング・システムの障害）を持ちこたえて、サービスを提供し続ける能力。

### **プライベート・インターコネクト (private interconnect)**

クラスタ内通信専用のネットワーク接続。プライベート・インターコネクトは、片方のノードが使用可能か使用不能かをもう一方のノードで検出できるため、ハートビート接続とも呼ばれる。プライベート・インターコネクトは、パブリック・インターコネクトとは異なるものである。「[パブリック・インターコネクト](#)」も参照。

### **プライマリ・ノード (primary node)**

アクティブ / パッシブ構成において処理を実行するノード。「[アクティブ / パッシブ構成](#)」も参照。

### **ホスト名 (host name)**

ネットワーク上の特定の IP アドレスを表す名前。Microsoft Cluster Server (MSCS) では、ホスト名はネットワーク名リソースにマップされる。「[ネットワーク名](#)」も参照。

### **ミッション・クリティカル (mission critical)**

企業にとってきわめて重要であり、高い可用性を必要とする種類の業務機能。

### **優先ノード (preferred node)**

可能所有者であるすべてのクラスタ・ノードが稼働している場合に、グループを常駐させるノード。「[フェイルオーバー・ノード](#)」も参照。

### **リスナー (listener)**

クライアントからの要求を受け取り、その要求を適切なサーバーにリダイレクトするサービス。

### **リソース (resource)**

コンピューティング・システムが使用できる物理的または論理的コンポーネント。たとえば、ディスク、ネットワーク IP アドレス、Oracle データベース・サーバー、リスナー、Oracle Forms Server、Oracle Reports Server、Web サーバーなどはリソースである。「[クラスタ・リソース](#)」および「[スタンドアロン・リソース](#)」も参照。



### **リソース再起動ポリシー (resource restart policy)**

リソース障害があった場合に、クラスタ・ソフトウェアが現在のノード上でそのリソースの再起動を試行するかどうかを指定し、試行する場合には一定時間内に試行する再起動の回数を指定するポリシー。

### **リソースの依存性 (resource dependencies)**

1 グループ内のリソース間の関係。クラスタ・ソフトウェアがそれらのリソースをオンラインおよびオフラインにする順番を定義する。

### **リソース・フェイルオーバー・ポリシー (resource failover policy)**

リソース障害によってグループのフェイルオーバーを発生させるかどうかを指定するポリシー。

### **ローリング・アップグレード (rolling upgrade)**

ソフトウェアを次のリリースにアップグレードしている間に、クラスタ・システムがサービスを提供し続けることを可能にしたソフトウェア・インストール・テクニック。クラスタ・システムおよびクライアント・ノードのアップグレードが完了するまで、各ノードが順々にアップグレードされてリブートされるため、このプロセスをローリング・アップグレードと呼ぶ。片方のノードが一時的にオフラインになっている間、アップグレード中のノードの作業負荷は他のノードに引き継がれる。



# 索引

## 数字

### 2 次ノード

- データベース・インスタンスの作成, 4-3
- データベース停止のテスト, 4-3

## A

### abort モード

- データベースのオフライン化, 5-4

### Active Server Page (ASP)

- Web アプリケーション, 12-4

### ALTER SESSION 文

- アプリケーション・フェイルオーバー後の再実行, 7-43

### Apache

- Oracle HTTP Server の起動

「Oracle HTTP Server」を参照

### Apache による Oracle HTTP Server の起動

- 「Oracle HTTP Server」を参照

## B

### Bequeath プロトコル・アダプタ

- Oracle Fail Safe のデータベースへの接続, 7-11

### BYPASS 値, 7-22

## C

### CGI (共通ゲートウェイ・インタフェース)

- Oracle Application Server, 12-13
- Oracle Forms Load Balancer Server, 9-2
- Oracle HTTP Server, 12-26
- RWCGI60.EXE 実行可能ファイルの検索, 11-26
- Web アプリケーション, 12-4

### CLUSTER\_MGR 文字列, C-4

### /CLUSTER 修飾子

- FSCMD コマンド, 5-4

### Concurrent Manager

- 「Oracle Applications Concurrent Manager」を参照

### CONNECT\_DATA パラメータ

- 値, 7-13

- 使用される SID パラメータ, 7-13

### CONNECT\_DATA 文字列, 7-38, 7-40

- DELAY キーワード, 7-40

- FAILOVER\_MODE パラメータ, 7-39

- METHOD キーワード, 7-40

- RETRIES キーワード, 7-40

- TYPE キーワード, 7-39

### CR\_DBRES 文字列, C-4

### CR\_SAMPLE 文字列, C-4

### ctxsrv サーバー

- 可用性を高めるための構成, 7-26

- 汎用アプリケーションとしての構成, 7-26

- ctxsrv サーバーのグループへの追加, 7-26

## D

### Database Configuration Assistant (DBCA), 7-14

- Net8 構成の作成, 7-14

- ネット・サービス名エントリ, 7-14

### DB\_NAME パラメータ, 7-8

### DBA\_AUTHORIZATION パラメータ, 7-22

### DCOM, 14-1

### DEL\_SAMPLE 文字列, C-4

### DELAY キーワード

- CONNECT\_DATA 文字列, 7-40

### DNS サーバー

- IP アドレスおよびホスト名の登録, B-1

### /DOMAIN 修飾子

FSCMD コマンド, 5-5  
DUMPCLUSTER パラメータ, 5-3

## F

---

FAILOVER\_MODE パラメータ, 7-39  
CONNECT\_DATA 文字列, 7-39  
METHOD キーワード, 7-40  
TYPE キーワード, 7-39  
FORMS60\_HTTP\_NEGOTIATE\_DOWN 環境変数,  
10-11  
FORMS60\_PATH レジストリ変数, 10-18, 10-19  
Oracle Forms Server, 10-12  
formsweb.cfg ファイル, 9-12, 9-14  
MetricServerPort パラメータ, 9-12  
Forms アプリケーション  
マルチフォーム・ページにアクセスできない,  
10-19  
Forms の .fmx ファイル, 10-19  
FSCMD コマンド, 1-4  
DUMPCLUSTER パラメータ, 5-3  
MOVEGROUP パラメータ, 5-3  
OFFLINEGROUP パラメータ, 5-3  
OFFLINERESOURCE パラメータ, 5-3  
ONLINEGROUP パラメータ, 5-3  
ONLINERESOURCE パラメータ, 5-3  
VERIFYALLGROUPS パラメータ, 5-3  
VERIFYCLUSTER パラメータ, 5-3  
VERIFYGROUP パラメータ, 5-3  
アクティブ / アクティブ構成での使用, 3-5  
構文, 5-2  
使用方法, 5-6  
スクリプトでの使用例, 5-7  
パラメータと修飾子, 5-3  
例, 5-7  
FsDbError.bat, 7-45  
FSFAILOVERGUI.EXE 実行可能プログラム  
デモ, A-1  
FSR\_TRACE\_OUTPUT 値, C-3  
FsResOds.DLL ファイル  
機能, 2-8  
FsResOdsEx.DLL ファイル  
機能, 2-8  
FSS\_TRACE\_FLAGS 値, C-4  
FSS\_TRACE\_OUTPUT 値, C-3

## G

---

GR\_VERIFY 文字列, C-4

## H

---

Hosts ファイル  
IP アドレスおよびホスト名の登録, B-1  
HOST パラメータ  
TNSNAMES.ORA ファイル, 7-38  
HTML ファイル  
JInitiator タグ, 10-18  
module= 構文, 10-18  
Oracle Forms アプリケーション, 10-17  
Oracle WebDB リスナー, 8-9  
HTTPS 接続モード, 10-11  
HTTP 接続モード, 10-11

## I

---

immediate モード  
データベースのオフライン化, 5-5  
INTERNAL アカウント, 7-49, 7-54  
パスワードの変更, 7-24  
IP アドレス  
LISTENER.ORA ファイルのネットワークの変更,  
B-4  
クラスタへの割当て, 2-13  
正しい名前解決の有効性検査, 12-31, B-2  
登録, B-1  
必要数の決定, 2-13  
複数のアドレスのトラブルシューティング, B-3  
複数リスナーによる同一アドレスのリスニング,  
8-6, 12-9  
不適切な名前解決のトラブルシューティング, B-3  
リスナー間での重複, 7-52  
リソース・タイプ, 2-8  
Is Alive ポーリング  
DLL ファイルの機能, 2-8  
バックアップ操作中の影響, 7-31  
変更された INTERNAL アカウントへの影響, 7-24

## J

---

Java  
アプレット・キャッシュをオフにする, 10-18  
アプレットの再ロード, 10-18

アプレットの実行時の問題, 10-17  
アプレットのダウンロード時の問題, 10-17  
仮想マシンを使用した一貫性の保持, 12-4

## JavaScript

Web アプリケーション, 12-4

## JInitiator

Oracle Forms に関するソフトウェア互換性, 10-17  
使用の推奨, 12-4

# L

---

## LISTENER.ORA ファイル, 4-4

Database Configuration Assistant による更新, 7-14  
EXTPROC 接頭辞, 7-17  
Net8 構成データのチェック, 7-48  
アーカイブ, 7-53  
仮想サーバーの構成で発生する問題, 7-51  
サンプル, B-4  
サンプル定義, 7-52  
複数の構成, 7-16  
変更, 7-51, B-4  
例, 7-52  
ロールバック・ファイル, 7-52

## Listener 制御ユーティリティ (LSNRCTL)

リスナーの作成, 7-53

## Load Balancer Server

「Oracle Forms Load Balancer Server」を参照

## LOCAL\_LISTENER パラメータ

グループの情報の更新, 7-21  
データベース・パラメータ・ファイルへの書き込み,  
7-22  
データベース・パラメータ・ファイルへの追加,  
7-20  
データベースをグループから削除した後の削除,  
7-22

## LocalSystem アカウント

汎用サービス, 15-10

## /LOGFILE 修飾子

FSCMD コマンド, 5-4

## LSNRCTL ユーティリティ

リスナーの問題のトラブルシューティング, 7-53

# M

---

## METHOD キーワード

CONNECT\_DATA 文字列, 7-40

## MetricServerPort パラメータ

formsweb.cfg ファイル, 9-12

## Microsoft Transaction Server

「Oracle MTS Service」を参照  
説明, 14-1

## Microsoft Windows NT Enterprise Edition 4.0

Service Pack 3 での Oracle Intelligent Agent の構成,  
B-5

登録に関する考慮事項, B-1

## Microsoft ハードウェア互換性リスト, 1-7

Microsoft 分散トランザクション・コーディネータ  
「MS DTC」を参照

## MOVEGROUP パラメータ, 5-3

## MS DTC

可用性を高めるための構成, 14-4

## MS DTC 構成の検証, 14-4

## MSCS

Is Alive ポーリング, 2-8

クオラム・リソース, 2-5

クラスタ アドミニストレータによるデータベース・  
プロパティの表示, 2-8

クラスタ・ソフトウェア, 1-1

保留タイムアウト値, 7-49, 13-16

リソース・タイプ, 2-8

## MTS\_DISPATCHER パラメータ

完全なリスナー情報の指定, 7-22  
更新, 7-21

## MTS\_LISTENER\_ADDRESS パラメータ

アドレスの変更, 7-22

# N

---

## Net8 Easy Config, 7-4, 7-13

## Net8 ネットワーク

TNSNAMES.ORA ファイルの変更, 7-13

アーカイブされた構成ファイル, 7-53

グループ内のリソース, 2-6

構成情報のトレース, C-4

構成のトラブルシューティング, 7-48

データベース構成, 7-12

ロールバック・ファイル, 7-53

## /NODE 修飾子

FSCMD コマンド, 5-4

## normal モード

データベースのオフライン化, 5-5

## O

### OCI

- アプリケーション操作, 7-42
- アプリケーション・フェイルオーバー・コールバック関数の起動, 7-43
- クライアント

- 自動再接続を使用可能にする, 7-38, 7-41

- 透過的アプリケーション・フェイルオーバーを示すデモ, A-1

- フェイルオーバー・プロセス, 7-36

OCI\_ATTR\_FOBCK 属性, 7-44

### ODBC クライアント

- 自動再接続を使用可能にする, 7-38, 7-40

- 透過的アプリケーション・フェイルオーバーを示すデモ, A-1

OFFLINEGROUP パラメータ, 5-3

OFFLINERESOURCE パラメータ, 5-3

/OFFLINE 修飾子

- FSCMD コマンド, 5-4

ONLINEGROUP パラメータ, 5-3

ONLINERESOURCE パラメータ, 5-3

ORA-01012, 7-33

ORA-01031, 7-54

ORA-01033, 7-33

ORA-01034, 7-33

ORA-01089, 7-33

ORA-03113, 7-33

ORA-03114, 7-33

ORA-12203, 7-33

ORA-12500, 7-33

ORA-12537, 7-34

ORA-12571, 7-34

Oracle Application Server Manager

- CGI 実行コードの変更, 12-13

Oracle Application Server (OAS), 12-4

- DLL ファイル, 2-8

- インストールのヒント, 12-17

- 構成情報, 12-7

- 単一ノードとしてのインストール, 12-17

- トラブルシューティング, 12-19

- マルチノード・インストールのプライマリ・ノードとしてのインストール, 12-17

- リソース・タイプ, 2-8

Oracle Application Server Web リスナー

- CGI 実行許可, 12-13

- IP アドレス, 12-5

- 仮想ディレクトリ, 12-16

- グループからの削除, 12-16

- グループへの構成, 12-6

- スタンドアロンの検出, 12-5

- リスナーの削除, 12-18

- 「リソース」も参照

Oracle Application Server のグループへの追加, 12-5

- Oracle ホーム, 12-11

- Web リスナー名, 12-12

- イニシャル・ファイル, 12-13

- 仮想アドレス, 12-14

- 仮想ディレクトリ, 12-13

- 可能所有者ノード, 12-9

- 前提条件, 12-6

- 手順, 12-7

- デフォルト・ファイル, 12-12

- ノード名, 12-11

- ルート・ディレクトリ, 12-13

Oracle Applications Concurrent Manager

- オフライン化

- 保留タイムアウト, 13-17

- 仮想アドレス, 13-9

- 可能所有者ノード, 13-7

- 構成 ID, 13-5, 13-10

- 削除された Windows サービスの再作成, 13-15

- スタンドアロンの検出, 13-4

- データベース, 13-13

- 名前, 13-10

- 不安定な状態の解決, 13-16

- 「リソース」も参照

Oracle Applications Concurrent Manager のグループへ

- の追加, 13-4

- アカウント, 13-12

- 仮想アドレス, 13-9

- 可能所有者ノード, 13-7

- 構成 ID, 13-10

- 前提条件, 13-4

- データベース, 13-13

- 手順, 13-5

- マネージャ名, 13-10

Oracle Data Migration Assistant

- データベースの移行, 7-25

Oracle Enterprise Manager

- アクティブ / アクティブ構成での使用, 3-5

- クライアント接続の問題のトラブルシューティング, 7-54

- 統合, 7-27

- 認証の問題, 7-54
- 認証の問題のトラブルシューティング, 7-54
- ユーザー設定項目の設定, 7-54
- Oracle Fail Safe
  - インストール, 1-4
  - 有効なクラスタ, 1-7
- Oracle Fail Safe Manager, 1-2
  - ウィザード, 1-4
  - 概要, 1-4
  - クラスタ別名, 2-13
  - 検証ツール, 1-4
  - ツリー・ビュー, 1-4
- Oracle Fail Safe Server
  - アカウント, 4-6
- Oracle Forms
  - CGI-bin 実行可能ファイル, 9-3
  - Forms ランタイム・エンジンの起動, 10-19
  - JInitiator の使用, 10-17
  - Web サーバーとの構成, 12-4
  - アプリケーションの JInitiator タグ, 10-18
- Oracle Forms Load Balancer Client
  - localhost, 9-14
  - Oracle Forms Load Balancer Server との関連, 9-2
  - 仮想アドレス, 9-14
  - 起動パラメータ, 9-14
  - データ・ポート, 9-14
  - データ・ホストのデフォルト値, 9-14
- Oracle Forms Load Balancer Server
  - CGI 実行可能ファイル, 9-2
  - formsweb.cfg ファイル, 9-12
  - Oracle Forms Load Balancer Client との関連, 9-2
  - Oracle Forms Server との関連, 9-2
  - 概要, 9-2
  - 仮想アドレス, 9-10
  - 可能所有者ノード, 9-8
  - 起動パラメータ, 9-14
  - 起動パラメータの変更, 9-14, 9-15
  - クライアントの最大数, 9-12
  - クラスタの 3 層構造, 9-3
  - グループへの追加, 9-5
  - グループへの追加手順, 9-6
  - 検出, 9-5
  - 構成手順, 9-6
  - スタンドアロンの検出, 9-5
  - セキュリティ要件, 9-13
  - データ・ポート, 9-11
  - トラブルシューティング, 9-15
  - トレース・レベル, 9-12
  - 名前, 9-11
  - リクエスト・ポート, 9-12
  - 「リソース」も参照
- Oracle Forms Load Balancer Server のグループへの追加, 9-6
  - 仮想アドレス, 9-10, 10-8
  - 可能所有者ノード, 9-8
  - サーバー名, 9-11
  - 前提条件, 9-6
  - データ・ポート, 9-11
  - 手順, 9-6
  - リクエスト・ポート, 9-12
- Oracle Forms Server, 10-16
  - FORMS60\_PATH レジストリ変数, 10-12
  - Oracle Fail Safe Manager ツリー・ビューでの外観, 10-14
  - Oracle Forms Load Balancer Server との関連, 9-2
  - アクセスされる Oracle データベースの指定, 10-13
  - アプリケーション・ファイルの場所, 10-12
  - 仮想アドレス, 10-4, 10-8
  - 可能所有者ノード, 10-7
  - 可用性の向上に必要な情報, 10-6
  - 可用性を高めるための構成, 10-4
  - クライアント接続, 10-16
  - クラスタ内に設定されるアプリケーション, 10-4
  - グループへの追加手順, 10-5
  - グループへの追加に対する前提条件, 10-4
  - 検出, 10-3
  - 構成手順, 10-5
  - セキュリティ要件, 10-15
  - データ・ファイル, 10-12
  - データベースへのアクセス, 10-13
  - トラブルシューティング, 10-16
    - アプリケーションが見つからない, 10-18
    - サーバーにアクセスできない, 10-17
  - 名前, 10-9
  - パス, 10-12
  - プール・パラメータ, 10-11
  - ポート, 10-9
  - 「リソース」も参照
- Oracle Forms Server のグループへの追加, 10-5
  - FORMS60\_PATH, 10-12
  - TNSNAMES.ORA ファイル, 10-13
  - アプリケーション・ファイルの場所, 10-12
  - 可能所有者ノード, 10-7
  - サーバー名, 10-9

- 接続モード・パラメータ, 10-10
- 前提条件, 10-4
- データベース, 10-13
- 手順, 10-5
- ポート番号, 10-9
- Oracle HTTP Server
  - HTTP アクセス, 12-27
  - Oracle ホーム, 12-20
  - SSL アクセス, 12-27
  - Web サイトに接続できない, 12-32
  - 仮想アドレス, 12-27
  - 仮想ディレクトリ, 12-26
  - 「可能所有者ノード」リスト, 12-22
  - 可用性を高めるための構成, 12-20
  - クライアント接続, 12-29
  - クラスタ・ディスク, 12-25
  - グループからの削除, 12-29
  - グループへの追加, 12-22
  - グループへの追加に対する前提条件, 12-20
  - 構成ファイル, 12-25, 12-29
  - スタンドアロンの検出, 12-20
  - セキュリティ要件, 12-30
  - ディレクトリ索引ファイル, 12-26
  - ドキュメント・ルート・ディレクトリ, 12-25
  - トラブルシューティング, 12-30
  - 名前, 12-24
  - ユーザーがアクセスできない, 12-31
  - 「リソース」も参照
- Oracle HTTP Server のグループへの追加, 12-20
  - 仮想アドレス, 12-27
  - 仮想ディレクトリ, 12-26
  - 可能所有者ノード, 12-22
  - サーバー名, 12-24
  - サーバー・ルート, 12-25
  - 前提条件, 12-20
  - ディレクトリ索引ファイル, 12-26
  - 手順, 12-21
- Oracle Intelligent Agent
  - IP アドレス構成に関する考慮事項, B-3
  - Service Pack 3 での構成, B-5
  - 仮想アドレス, 7-29
  - グループへの追加, 7-28
  - 構成ファイルの変更, B-3
  - 通信の管理, 2-6
  - デフォルト, 7-29
  - リスニング・パラメータのための SNMP\_RW.ORA ファイルの変更, B-4
- Oracle Intelligent Agent のグループへの追加, 7-28
  - 前提条件, 7-29
- Oracle MTS Service
  - 仮想アドレス, 14-10
  - 可能所有者ノード, 14-8
  - 検出, 14-4
  - サービス名, 14-11
  - 削除した場合の再作成, 14-16
  - スタンドアロンの検出, 14-4
  - データベース, 14-12
  - 名前, 14-11
- Oracle MTS Service のグループへの追加, 14-4
  - アカウント認証, 14-13
  - 仮想アドレス, 14-10
  - 可能所有者ノード, 14-8
  - サービス名, 14-11
  - 前提条件, 14-4
  - データベース, 14-12
  - 手順, 14-5
- Oracle Net8 Assistant
  - ネットワーク・ファイルの再作成のための使用, 7-51
- Oracle Net8 リスナー
  - 依存性, 7-11
  - 構成, 7-10
  - 作成, 7-10
  - 動作, 7-11
- Oracle Reports Server
  - persistfile パラメータ, 11-13
  - .rdf ファイル, 11-13
  - .rdf ファイル拡張子, 11-27
  - TNSNAMES.ORA ファイル, 11-18
  - TNSNAMES.ORA ファイルの例, 11-18
  - Web サーバー仮想ディレクトリのレジストリ変数, 11-28
  - Web サーバーとの構成, 12-4
  - Web ベースの構成, 11-2
  - アーキテクチャ, 11-2
  - アカウント, 11-15
  - アクセスされるデータベース, 11-14
  - 仮想アドレス, 11-10
  - 仮想アドレスの判断, 11-18
  - 可能所有者ノード, 11-8
  - 可用性を高めるための構成, 11-6
  - 既存のマスター / スレープ実装へのスレープ・サーバーの追加, 11-22
  - キャッシュ・ディレクトリ, 11-13



- 仮想ディレクトリによる指定, 11-28
- キャッシュ・ディレクトリのレジストリ変数, 11-28
- クライアント・アクセスのための
  - TNSNAMES.ORA ファイルの更新, 11-18
- クライアント / サーバー実装, 11-2, 11-3
- クライアント接続, 11-16
- クラスタ構成, 11-5
- 検出, 11-5
- 識別情報, 11-12
- 実行可能ファイルの場所, 11-6
- 出力ファイルの場所, 11-6
- ジョブ・ディレクトリ, 11-13
- スレーブ・サーバーに対するマスター・サーバーの指定, 11-21
- 層, 11-2
- ソース・ディレクトリ, 11-13
  - クラスタ・ディスク, 11-27
- 通常ファイル, 11-25
- データ・ファイルの場所, 11-6
- トラブルシューティング, 11-26
- 名前, 11-12
- 複数, 3-7
- プリンタ, 11-15
- プリンタにアクセスするためのアカウント, 11-15
- ポート, 11-12
- マスター / スレーブ実装, 11-2, 11-5
- マスター / スレーブ実装の構成, 11-19
- マスターに対するスレーブの指定, 11-20
- マスターに対する高い可用性の提供, 11-1
- ランタイム・エンジン, 11-27
- レポートの実行, 11-17
- ロード・バランス, 11-5
- 「リソース」も参照
- Oracle Reports Server のグループへの追加, 11-7
  - persistfile パラメータ, 11-13
  - TNSNAMES.ORA ファイル, 11-14
  - 仮想アドレス, 11-10
  - 可能所有者ノード, 11-8
  - キャッシュ・ディレクトリ, 11-13
  - サーバー名, 11-12
  - ジョブ・ディレクトリ, 11-13
  - 前提条件, 11-6
  - ソース・ディレクトリ, 11-13
  - データベース, 11-14
  - 手順, 11-6
  - プリンタ・アクセス用のアカウント, 11-15
  - ポート, 11-12
- Oracle Services for Microsoft Transaction Server
  - 「Oracle MTS Service」を参照
- Oracle TNS Listener
  - リソース・タイプ, 2-9
- Oracle WebDB
  - 概要, 8-1
- Oracle WebDB リスナー
  - Oracle ホーム, 8-8
  - PL/SQL 設定の変更, 8-13
  - TNSNAMES.ORA ファイル, 8-11
  - wdbsvr.app 構成ファイル, 8-9
  - wdbsvr.cfg 構成ファイル, 8-9
  - アクセスされるデータベース, 8-10
  - インストールのヒント, 8-13
  - 仮想アドレス, 8-7
  - 可用性を高めるための構成, 8-1
  - 関連するデータベースへのクライアント接続, 8-13
  - 共有クラスタ・ディスク, 8-9
  - クラスタの 2 層構造, 8-2
  - クラスタの 3 層構造, 8-2
  - グループへの追加, 8-4
  - グループへの追加に対する前提条件, 8-4
  - 構成ファイル, 8-9
  - スタンドアロンの検出, 8-4
  - セキュリティ要件, 8-13
  - データベースへのアクセスに使用するアカウント, 8-12
  - トラブルシューティング, 8-15
  - 名前, 8-8
  - 複数データベースの指定, 8-11
  - ポート, 8-8
  - リスナー設定の変更, 8-13
  - 「リソース」も参照
- Oracle WebDB リスナー構成ファイル
  - wdbsvr.app, 8-9
  - wdbsvr.cfg, 8-9
- Oracle WebDB リスナーのグループへの追加, 8-4
  - HTML ファイル, 8-9
  - Oracle Fail Safe で作成されるファイル, 8-9
  - Oracle ホーム, 8-8
  - TNSNAMES.ORA ファイル, 8-11
  - wdbsvr.app, 8-9
  - wdbsvr.cfg, 8-9
  - アカウント, 8-12
  - 仮想アドレス, 8-7
  - クラスタ・ディスク, 8-9

- 作成される構成ファイル, 8-9
- 接続データベース, 8-10
- 前提条件, 8-4
- 手順, 8-4
- ポート, 8-8
- リスナー名, 8-8
- Oracle7 データベース
  - 「データベース」を参照
- Oracle8i データベース
  - 「データベース」を参照
- Oracle8 データベース
  - 「データベース」を参照
- Oracle データベース
  - Oracle8i へのアップグレード, 7-15
- Oracle データベースのグループへの追加, 7-3
  - SQL\*Net リスナー, 7-12
  - TNSNAMES.ORA ファイル, 7-4
  - インスタンス名, 7-8
  - 仮想アドレス, 7-6
  - 可能所有者ノード, 7-5
  - サービス名, 7-8
  - 前提条件, 7-3
  - データベースの INTERNAL アカウント, 7-9
  - データベース名, 7-8
  - 手順, 7-4
- Oracle データベース・リソース・タイプ, 2-8
- Oracle ホーム
  - Oracle Application Server の構成, 12-6
  - Oracle Forms Load Balancer Server の構成, 9-6
  - Oracle Forms の構成, 10-4
  - Oracle HTTP Server のパス, 12-20
  - グループ, 4-9
  - 複数, 4-8
- Oracle リスナー
  - 汎用サービス, 2-9

## P

---

- Perl
  - Web アプリケーション, 12-4
- persistfile パラメータ, 11-13
  - Oracle Reports Server, 11-13
- ping
  - クラスタ・ノード間のネットワーク通信の検証, 12-31, B-2
- PL/SQL ゲートウェイ
  - 設定の変更, 8-13

- /PWD 修飾子
  - FSCMD コマンド, 5-5

## R

---

- RAID
  - テクノロジー, 2-3
  - ハードウェア, 1-7
- readme.txt 手順ファイル
  - 透過的アプリケーション・フェイルオーバー・デモ, A-1
- REMOTE\_LOGIN\_PASSWORDFILE 初期化パラメータ, 7-49
  - 設定, 7-49
- REPORTS60\_CGIMAP レジストリ変数, 11-17
- REPORTS60\_PHYSICAL\_MAP レジストリ変数, 11-17, 11-28
- REPORTS60\_VIRTUAL\_MAP レジストリ変数, 11-17, 11-28
- RETRIES キーワード
  - CONNECT\_DATA 文字列, 7-40
- RWCIG60.EXE ファイル
  - 検索時の問題, 11-26
  - ダウンロード, 11-27

## S

---

- SCSI
  - 共有記憶装置インターコネクト, 1-7
- Security Setup ツール
  - Windows NT セキュリティ情報の更新, 4-6
- SELECT 文
  - 透過的アプリケーション・フェイルオーバー中の再実行, A-1
  - フェイルオーバー後の再実行, 7-39
- Server Manager
  - 管理作業の実行, 7-31
- Service Pack 4
  - デフォルトの Oracle Intelligent Agent の構成, B-5
- SERVICE\_NAME パラメータ
  - CONNECT\_DATA パラメータでの使用, 7-13
  - Database Configuration Assistant による追加, 7-14
- SID
  - CONNECT\_DATA で使用されるパラメータ, 7-13
  - CONNECT\_DATA 文字列内のキーワード, 7-39
  - Database Configuration Assitant により追加される名前, 7-14

- TNSNAMES.ORA ファイルの項目, 4-4
  - 正しいデータベースの入力, 7-51
  - 複数のリスナー, 7-16
  - リスナー間で重複した名前, 7-52
- SID\_DESC パラメータ, 7-14
  - Oracle8i へのアップグレード, 7-15
  - 新規データベース用の追加, 7-14
  - データベースをグループに追加した後の更新, 7-14
- SNMP\_RW.ORA ファイル
  - サンプル, B-4
  - 複数 Oracle ホーム環境での変更, B-4
- Socket 接続モード, 10-10
- SQLNET.ORA ファイル
  - デフォルト・ドメイン値の更新, 7-14
- SQLNET 文字列, C-4
- SYSDBA 権限, 7-23
- SYSOPER 権限, 7-23

## T

---

- TNSNAMES.ORA ファイル, 4-4
  - CONNECT\_DATA 文字列, 7-38, 7-39
  - Database Configuration Assistant による更新, 7-14
  - EXTPROC\_CONNECTION\_DATA ネット・サービス名エントリ, 7-18
  - HOST パラメータ, 7-38
  - Net8 構成データのチェック, 7-48
  - Oracle Forms Server, 10-13
  - Oracle Reports Server, 11-14, 11-18
  - Oracle Reports Server のマスター / スレーブ実装, 11-20
  - Oracle Reports Server の例, 11-18
  - Oracle WebDB リスナー, 8-11
  - Oracle データベース・サーバー, 7-4
  - アーカイブ, 7-53
  - 仮想サーバーの構成で発生する問題, 7-51
  - 可用性の高いデータベース用の変更, 7-12
  - 更新, 7-12
  - データベース検出のための分析, 7-2
  - デフォルト・ドメイン名の値での更新, 7-14
  - 透過的アプリケーション・フェイルオーバー・デモのための更新, A-1
  - 複数 Oracle ホーム環境での変更, B-4
  - ロールバック・ファイル, 7-52
- TNS リスナー
  - ネット・サービス名の更新, 7-16
- transactional モード

- データベースのオフライン化, 5-5
- TYPE キーワード
  - CONNECT\_DATA 文字列, 7-39

## U

---

- UNIX システム
  - 階層化構成, 3-8
- URL
  - Oracle HTTP Server Web サイト用の指定, 12-29
  - Oracle Reports Server の通信のトラブルシューティング, 11-27
  - 仮想ディレクトリの指定, 12-13
- /USER 修飾子
  - FSCMD コマンド, 5-5

## V

---

- VERIFY\_DB 文字列, C-4
- VERIFYALLGROUPS パラメータ, 5-3
- VERIFYCLUSTER パラメータ, 5-3
- VERIFYGROUP パラメータ, 5-3
- Visual Basic
  - Web アプリケーション, 12-4

## W

---

- wdbsvr.app 構成ファイル, 8-9
  - データベースの指定, 8-10
- wdbsvr.cfg 構成ファイル, 8-9
- WebDB
  - 「Oracle WebDB」を参照
- Web アプリケーション
  - Oracle Fail Safe によるサポート, 12-4
  - 標準, 12-4
- Web サーバー
  - 「Oracle Application Server (OAS)」を参照
  - 「Oracle HTTP Server」を参照
- Web サイト
  - Oracle HTTP Server に接続できない, 12-32
  - トラブルシューティング, 8-15, 12-19
- Web ブラウザ
  - Java アプレットをダウンロードできない, 10-17
  - JInitiator を使用した一貫性の保持, 12-4
  - Oracle Reports の .rdf ファイルへのアクセス, 11-27
  - Oracle Reports の再ロード, 11-16

- フェイルオーバー後の Oracle Forms の再ロード,  
10-4
- Web ページ
  - 仮想ディレクトリにより表される, 12-13
  - 作業負荷のパーティション化構成, 3-6
- Windows 2000 Advanced Server, 1-7
  - デフォルトの Oracle Intelligent Agent の構成, B-5
- Windows NT Enterprise Edition, 1-7
- Windows NT ユーザー・アカウント
  - セキュリティ設定, 4-6
- Windows クラスタ
  - 「クラスタ」を参照
- Windows レジストリ・エディタ
  - 「レジストリ・エディタ」を参照
- Windows レジストリキー
  - 「レジストリ・キー」を参照
- WINS
  - IP アドレスおよびホスト名の登録, B-1

## あ

---

- アーカイブされたファイル, 7-53
- アカウント
  - INTERNAL
    - パスワードの変更, 7-24
  - Oracle Application Server のインストールに使用,  
12-16
  - Oracle Applications Concurrent Manager を実行す  
る, 13-12
  - Oracle Fail Safe Server の稼働する, 4-6
  - Oracle MTS Service に必要な権限, 14-15
  - Oracle MTS Service に必要なデータベース・ロー  
ル, 14-15
  - Oracle MTS Service の実行に使用, 14-13
  - Oracle Reports Server の実行に使用, 11-15
  - Oracle WebDB でデータベースへのアクセスに使  
用, 8-12
  - Oracle WebDB リスナーのデータベースへのアクセ  
ス, 8-12
  - Oracle データベースの管理, 7-22
    - 可用性の高い Oracle データベースを監視するため  
の使用, 7-9
    - 権限および許可, 4-5
    - データベースの INTERNAL ログイン・アカウント,  
7-9
    - データベース・パスワード・ファイルへの追加,  
7-23

- 汎用サービスの実行に使用される, 15-9, 15-10
- アクセス
  - リソース, 2-6
- アクティブ / アクティブ構成, 3-3
- アクティブ / パッシブ構成, 3-2
- 値
  - BYPASS, 7-22
- アプリケーション
  - クラスタ対応, 7-32
  - クラスタ対応ではない, 7-33
  - クラスタ対応の ODBC および OCI, 7-33
  - フェイルオーバー, 1-6, 2-33, 7-32
    - 概念, 7-36
    - フロー・チャート, 7-36
  - フェイルオーバー後の再接続, 7-32, 7-38, 7-39
  - フェイルオーバー後の自動再接続, 1-6
  - フェイルオーバー時に返される一般的なエラー,  
7-33, 7-34
  - ミッション・クリティカル, 3-5
- アプリケーション・ソフトウェア
  - ローリング・アップグレード, 1-4
- アプリケーション・フェイルオーバー・コールバック,  
7-43
  - 登録, 7-44
- アプリケーション・ログ
  - 起動問題のトラブルシューティング, 6-12
- アラート・ファイル, C-4

## い

---

- 依存性
  - Net8 リスナー, 7-11
  - 定義, 15-13
  - 汎用サービス, 15-13
  - リソース, 2-7
- 一時表
  - フェイルオーバー, 7-3
- 「一般」タブ・ページ
  - リソース, 2-23
- イニシャル・ファイル
  - Oracle Application Server Web リスナー, 12-13
- イベント
  - 情報を格納, 7-29
- イベント ビューア
  - Oracle Application Server メッセージの検査, 12-9
- イメージ名
  - 汎用サービス, 15-8

## インスタンス名

- 高い可用性を備えたデータベース, 7-8
- 「SID」を参照

## インストール

- Oracle Application Server 用のヒント, 9-14, 12-17
- Oracle Fail Safe, 1-4
- Oracle WebDB リスナーのヒント, 8-13
- 応答ファイルを使用, 1-4
- サイレント・モード, 1-4
- セキュリティ設定情報の提供, 4-6
- 対話式, 1-4

## インターコネクト

- 共有, 7-47
- 共有記憶装置, 2-2
- プライベート・ノード間接続, 2-2

## う

---

### ウィザード, 1-4

- 入力項目, 4-2

## え

---

### エラー

- FS-10066, 7-52
- FS-10070, 7-51
- FS-10101, 7-54
- FS-10270, 7-50
- Net8 リスナーによる記録, 7-48
- トレース・ファイル内の情報の追跡, C-2
- 不十分な権限に対して返される, 7-54
- ユーザー認証, 7-54
- レポート, C-1
- 「ORA」も参照

### エラー処理

- スクリプト, 7-45

## お

---

### オフライン・オプション・モード, 5-4

- オラクル社カスタマ・サポート・センター
- 連絡, C-1

## か

---

### 階層化構成, 3-6

- UNIX の使用, 3-8

## 外部プロシージャ

- 構成, 7-17

## カスタマ・サポート・センター

- 連絡, C-1

## カスタム・リソース・タイプ, 2-8

## 仮想アドレス, 1-2

- Oracle Applications Concurrent Manager, 13-9
- Oracle Forms Load Balancer Client, 9-14
- Oracle Forms Load Balancer Server, 9-10
- Oracle Forms Server, 10-4, 10-8
- Oracle HTTP Server, 12-27
- Oracle Intelligent Agent, 7-29
- Oracle Intelligent Agent での使用, 7-29
- Oracle MTS Service, 14-10
- Oracle Reports Server, 11-10
- Oracle Reports Server のマスター / スレーブ実装, 11-20
- Oracle WebDB リスナー, 8-7
- Oracle データベース, 7-6
- クラスタ別名, 2-13
- グループ, 7-4
- グループで複数の仮想アドレスを使用する理由, 4-10

- 構成の問題のトラブルシューティング, 7-51

- 識別, 2-10

- 定義, 2-10

- 登録, B-1

- 内部クラスタ通信, 7-13

- 汎用サービス, 15-14

- 複数, 2-11, 4-9

## 仮想サーバー, 1-2

- TNSNAMES.ORA での指定, 7-38

- アドレスの登録, B-1

- 検出, 7-27

- 構成, 2-12

- 構成時の TNSNAMES.ORA ファイルに対する更新, 7-12

- 構成の問題, 7-51

- 定義, 2-11

- ネットワーク構成, 7-12

- 複数, 4-9

- 複数仮想アドレス, 2-11

- リスナー障害, 7-53

## 仮想ディレクトリ

- CGI 実行可能ファイルのアクセス, 11-26

- Oracle Application Server Web リスナー, 12-13

- Oracle HTTP Server, 12-26

- キャッシュ・ディレクトリ, 11-28
- 指定, 10-17
- 仮想ディレクトリ設定
  - Java アプレットのダウンロード, 10-17
- 可能所有者ノード
  - フェイルオーバーへの影響, 2-28
- 「可能所有者ノード」リスト
  - Oracle Application Server Web リスナー, 12-9
  - Oracle Applications Concurrent Manager, 13-7
  - Oracle Forms Load Balancer Server, 9-8
  - Oracle Forms Server, 10-7
  - Oracle HTTP Server, 12-22
  - Oracle MTS Service, 14-8, 14-11
  - Oracle Reports Server, 11-8
  - Oracle データベース, 7-5
  - グループ, 2-23, 2-24
  - 「グループの検証」コマンド, 2-24, 4-11
  - 「グループの検証」操作, 2-23, 2-24
  - 指定, 2-23
  - 調整, 2-23
  - 汎用サービス用, 15-5
  - フェイルオーバー・ノードの決定, 2-28
  - 変更, 2-23
  - リソース, 2-22, 2-23
  - リソースへの影響, 2-22
- 可用性
  - RAID テクノロジー, 2-3
- 環境変数
  - FORMS60\_HTTP\_NEGOTIATE\_DOWN, 10-11
- 関数
  - アプリケーション・フェイルオーバー・コールバック, 7-43
- 管理
  - Oracle Enterprise Manager との統合, 7-27
  - クライアント・システムの TNSNAMES.ORA, 7-4, 7-13
  - クラスタ・セキュリティ, 4-5
  - クラスタ別名の使用, 2-13
  - 計画的ダウン時間, 2-18
- 管理者権限
  - FSCMD コマンドによる指定, 5-5
  - クラスタへのログイン, 4-7
  - セキュリティの設定, 4-6
- 管理操作
  - データベース, 7-31

## き

---

- 起動タイプ
  - 汎用サービス, 15-10
- 起動パラメータ
  - Oracle Forms Load Balancer Client, 9-14
  - Oracle Forms Load Balancer Server, 9-14
  - Oracle Forms Load Balancer Server の変更, 9-15
  - 汎用サービス, 15-9
- キャッシュ
  - Java アプレット, 10-18
- キャッシュ・ディレクトリ
  - Oracle Reports Server, 11-13
- 共有記憶装置インターコネクト, 1-3, 2-2
  - SCSI, 1-7
  - ファイバ・チャネル, 1-7
- 許可
  - 不十分なため返されるエラー ORA-01031, 7-54

## く

---

- クオラム・ディスク
  - クラスタ・リカバリ, 2-5
  - 情報のダンプ, 6-10
- クオラム・リソース
  - 場所の指定, 2-5
  - 場所の表示, 2-5
  - メタデータ, 2-5
- クライアント
  - アドレスの登録, B-1
  - クラスタの接続, 2-33
  - フェイルオーバー後の Oracle Forms Server への再接続, 10-4
  - フェイルオーバー後の Oracle Reports Server への再接続, 11-16
  - フェイルオーバー後の Oracle7 データベースへの再接続, 7-33
  - フェイルオーバー後の Oracle8i データベースへの再接続, 7-34
  - フェイルオーバー後の Oracle8 データベースへの再接続, 7-34
  - フェイルオーバー後の WebDB Web サイトへの再接続, 8-1
  - フェイルオーバー後の Web サーバーへの再接続, 12-1
  - フェイルオーバー後の Web サイトの再ロード, 8-1, 12-1

- フェイルオーバー後のデータベースへの再接続, 7-32
- 複数仮想アドレスによるグループのアクセス, 4-10
- 複数仮想アドレスによるリソースへの接続, 4-9
- リソースへのアクセス, 2-10
- クライアント / サーバー構成
  - Oracle Reports, 11-1
- クライアント接続
  - Oracle Forms Server, 10-16
  - Oracle HTTP Server, 12-29
  - Oracle Reports Server, 11-16
  - Oracle WebDB リスナー, 8-13
  - Oracle データベース・サーバー, 7-12
- クラスタ, 1-1
  - Oracle Fail Safe の使用有効, 1-7
  - 階層化構成, 10-2
  - 概要, 2-2
  - 各種の構成, 2-3
  - クラスタ別名, 2-13
  - 構成, 2-2
  - 構成の検証, 5-3, 6-3
  - 新規ノードの追加, 4-10
  - 単一ノード, 2-5
  - 定義, 2-1
  - ディスク
    - 「クラスタ・ディスク」を参照
  - ディスク構成, 2-4
  - 典型的な構成, 1-7
  - トラブルシューティング
    - 「クラスタのダンプ」操作, 6-10
  - ノード, 1-2, 2-2
  - パスワード・ファイルの同期化, 7-23
  - メタデータ, 2-5
  - メンバー, 2-2
- クラスタ アドミニストレータ
  - Oracle Application Server プロパティの表示, 2-8
  - Oracle Reports Server、Forms Server およびリスナー・リソース・タイプの表示, 2-9
  - Oracle データベース・プロパティの表示, 2-8
  - トレース中の使用, C-3
- クラスタ・グループ, 2-13
  - クラスタ別名ネットワーク名, 2-13
  - リソース, 2-13
- クラスタ対応のアプリケーション, 7-32
  - フェイルオーバー時のエラー処理, 7-33, 7-34
- クラスタ・ディスク
  - Oracle Application Server, 12-6
  - Oracle Forms Server, 10-4
  - Oracle HTTP Server, 12-21, 12-25
  - Oracle Reports Server, 11-6
  - Oracle WebDB リスナー, 8-9
  - Oracle データベース, 7-3
  - RAID ハードウェア, 1-7
  - 汎用サービス, 15-11
- クラスタ内通信, 1-7
- クラスタ・ノード, 2-2
  - 既存クラスタへの追加, 4-10
  - 定義, 1-1
- 「クラスタの検証」操作
  - 説明, 6-2
  - 用途, 6-3
- 「クラスタのダンプ」コマンド, 6-10
  - クオラム・リソースの表示, 2-5
  - 発行, 2-5
- クラスタ別名
  - Oracle Fail Safe Manager, 2-13
  - アドレスの登録, B-1
  - 使用, 2-13
  - 定義, 2-13
- クラスタへの接続
  - ドメイン・ユーザー・アカウント, 4-7
- クラスタ・リカバリ
  - クオラム・ディスク, 2-5
- クラスタ・リソース
  - 定義, 2-6
- クラスタ・レジストリ
  - 修正のための「グループの検証」の実行, 7-31
- クラスタワイド操作
  - 「グループの検証」操作, 6-5
  - 「スタンドアロン・データベースの検証」操作, 6-9
  - ロールバック, 7-47
- クラス例外エラー
  - Java アプレットの実行時, 10-17
- グループ
  - DBA パスワードの不一致, 7-50
  - Oracle Application Server の追加, 12-6, 12-7
  - Oracle Applications Concurrent Manager の追加, 13-5
  - Oracle Enterprise Manager による検出, 7-27
  - Oracle Forms Load Balancer Server の追加, 9-6
  - Oracle Forms Server の追加, 10-5
  - Oracle HTTP Server の追加, 12-21
  - Oracle Intelligent Agent の追加, 7-28
  - Oracle MTS Service の追加, 14-5

Oracle Reports Server の追加, 11-7  
Oracle WebDB リスナーの追加, 8-4  
Oracle ホーム, 4-9  
移動, 5-3  
移入, 4-3  
オフライン化, 5-3  
オンライン化, 5-3  
仮想アドレス, 7-4  
クラスタ内のすべてのグループの検証, 5-3  
情報のダンプ, 6-10  
所有権, 2-6  
帯域幅追加, 2-11  
定義, 2-6  
データベースの追加, 7-4  
データベースを追加するための権限, 7-54  
ネットワーク名の修正, 7-51  
「ノード」タブ・ページ, 2-23  
汎用サービスの追加, 15-3  
フェイルオーバー, 2-6  
フェイルオーバー期間, 2-26  
フェイルオーバーしきい値, 2-26  
フェイルオーバー・ポリシー, 2-25  
複数仮想アドレス, 4-9, 4-10  
「優先ノード」リスト, 2-28  
リソースの依存性, 2-7  
リソースの検証, 6-5  
リソースの例, 2-6  
グループの移動操作  
    フェイルバックへの影響, 2-32  
「グループの検証」コマンド  
    「可能所有者ノード」リスト, 2-24, 4-11  
「グループの検証」操作  
    FSCMD コマンド, 5-3  
    Oracle Forms Server の検証, 10-17  
    Oracle HTTP Server の検証, 12-31  
    概要, 6-5  
    「可能所有者ノード」リスト, 2-23, 2-24  
    クラスタ・レジストリ内の情報の更新, 7-31  
    説明, 6-2  
    用途, 6-5  
    リソース依存性の更新, 7-46  
    ログ情報, C-4

## け

---

計画外フェイルオーバー, 2-15  
計画的フェイルオーバー, 2-18

    静的ロード・バランス, 2-18  
計画的メンテナンス, 2-18  
権限  
    各クラスタ・ノードでの付与, 7-23  
    汎用サービス, 15-16  
    必須, 4-5  
検出  
    Oracle Enterprise Manager, 7-27  
    スタンドアロン Oracle Application Server Web リス  
        ナー, 12-5  
    スタンドアロン Oracle Applications Concurrent  
        Manager, 13-4  
    スタンドアロン Oracle Forms Load Balancer  
        Server, 9-5  
    スタンドアロン Oracle Forms Server, 10-3  
    スタンドアロン Oracle HTTP Server, 12-20  
    スタンドアロン Oracle MTS Service, 14-4  
    スタンドアロン Oracle Reports Server, 11-5  
    スタンドアロン Oracle WebDB リスナー, 8-4  
    スタンドアロン・データベース, 7-2  
    スタンドアロンのサンプル・データベース, 7-2  
    スタンドアロン汎用サービス, 15-3  
    リソース, 4-8  
検証ツール, 1-4

## こ

---

構成, 3-1 ~ 3-9  
    アクティブ / アクティブ, 3-3  
    アクティブ / パッシブ, 3-2  
    ウィザード入力項目の使用, 4-2, 4-3  
    階層化, 3-6  
    カスタマイズ, 3-1  
    グループの問題のトラブルシューティング, 13-16,  
        14-16  
    検証, 6-5, 6-7  
    作業負荷のパーティション化, 3-5  
    シェアード・ナッシング, 2-4  
    システム・レベル, 2-3  
    ディスク・レベル, 2-4  
    典型, 1-7  
    複数仮想アドレス, 4-9  
構成 ID  
    Oracle Applications Concurrent Manager, 13-5,  
        13-10  
構成のカスタマイズ, 3-1  
構成ファイル



- Oracle HTTP Server, 12-25
- Oracle WebDB リスナー, 8-9
- コールバック
  - アプリケーション・フェイルオーバー, 7-43, A-1
- コンポーネント
  - 冗長, 12-3

## さ

---

- サーバー・コンテキスト・ハンドル
  - アプリケーション・フェイルオーバー, 7-44
- サーバー・ノード
  - FSS\_TRACE\_FLAGS 値および文字列, C-4
  - アドレスの登録, B-1
  - アプリケーションへの自動再接続, 7-32
  - クライアント接続の喪失, 7-54
  - クラスタ・ポーリングの障害, 7-50
  - 再起動の試行, 2-33
  - トレース・フラグ, C-4
- サーバーの冗長性, 2-3
- 「サービスとしてログオン」する権限
  - 汎用サービス, 15-16
- サービス名
  - Oracle MTS Service, 14-11
  - 汎用サービス, 15-8
  - 「ネット・サービス名」も参照
- 再接続
  - アプリケーション, 7-39
  - クライアント, 7-38
- サイレント・モードのインストール, 1-4
- 作業環境
  - アクセス問題を回避するための設定, 7-54
- 作業負荷
  - 多大な負荷に耐えうる保留タイムアウトの設定, 7-49, 13-16
- 作業負荷のパーティション化構成, 3-5
- 削除
  - グループから Oracle Application Server Web リスナーを, 12-16
  - グループから Oracle HTTP Server を, 12-29
- サンプル・データベース
  - 必要な権限, 7-54
  - 問題のトラブルシューティング, 7-50
- サンプル・データベースの削除
  - エラー, 7-51
  - 情報の獲得, C-4
- サンプル・データベースの作成

- 障害, 7-50
- 情報のトレース, C-4

## し

---

- シェアード・ナッシング構成, 2-4
- システム・アカウント
  - Oracle Reports Server, 11-15
- 実行
  - スタンドアロン Oracle HTTP Server, 12-29
- 指定
  - 「可能所有者ノード」リスト, 2-23
- 出力ファイル
  - リスナー, 7-53
- 障害
  - アプリケーション内での OCI エラー処理, 7-34
  - 階層化構成での除去, 9-3, 10-2, 14-3
  - 計画外, 2-15
  - 計画的, 2-18
  - 防止のためのリソースの検証, 6-5, 6-7
- 冗長性
  - サーバー, 2-3
- 初期化パラメータ・ファイル
  - LOCAL\_LISTENER パラメータの削除, 7-22
  - MTS\_LISTENER\_ADDRESS パラメータのアドレスの変更, 7-22
  - REMOTE\_LOGIN\_PASSWORDFILE 初期化パラメータ, 7-22, 7-49
  - 高い可用性を備えたデータベース, 7-8
  - 配置, 7-8
  - 場所, 7-8
- ジョブ
  - アクセス問題のトラブルシューティング, 7-54
  - 情報を格納, 7-29
  - スケジュールした Oracle Reports, 11-13
- ジョブ・ディレクトリ
  - Oracle Reports Server, 11-13
- 所有権
  - グループ, 2-6
- 所有者ノード
  - 「可能所有者ノード」リスト」を参照

## す

---

- スクリプト
  - FSCMD コマンドの使用, 5-2, 5-7
  - FsDbError.bat, 7-45

アクティブ / アクティブ構成の FSCMD コマンド,  
3-5  
データベースをオンラインにできない場合のエラー  
処理, 7-45  
スケジュールしたジョブ  
Oracle Reports Server, 11-13, 11-25  
スタンドアロン・データベース  
Oracle8i およびマルチスレッド・サーバー, 7-20  
検出, 7-2  
検証, 4-2, 6-7  
スタンドアロン・データベースの検証, 4-2, 6-7  
表示, 7-2  
マルチスレッド・サーバーを伴うグループへの追  
加, 7-21  
リスナーの検索, 7-16  
「スタンドアロン・データベースの検証」操作, 6-7  
「クラスタワイド操作の状態」ウィンドウ, 6-9  
新規 Oracle8i データベースに対する実行, 7-14  
説明, 6-2  
トレース, C-4  
スタンドアロン・リソース  
検出, 4-8  
各リソースの索引項目も参照  
スタンバイ構成, 3-2  
スループット  
最大化, 3-5  
スレーブ・サーバー  
マスター Oracle Reports Server に対する指定,  
11-20

## せ

---

静的ロード・バランス, 2-18  
セキュリティ  
FSCMD コマンドによる指定, 5-5  
INTERNAL アカウントの変更, 7-24  
Oracle Application Server Web リスナーの CGI 実  
行許可, 12-13  
Oracle Fail Safe Security Setup ツール, 4-6  
データベースの INTERNAL パスワードの変更,  
7-24  
ノード上のパスワード・ファイルの同期化, 7-23  
複数仮想アドレスによる向上, 2-11  
要件  
Oracle Application Server, 12-16  
Oracle Applications Concurrent Manager, 13-14  
Oracle Forms Load Balancer Server, 9-13

Oracle Forms Server, 10-15  
Oracle HTTP Server, 12-30  
Oracle MTS Service, 14-15  
Oracle WebDB リスナー, 8-13  
データベース, 7-22  
汎用サービス, 15-10, 15-16  
接続モード  
HTTP, 10-11  
HTTPS, 10-11  
Socket, 10-10  
専用ネットワーク・リンク, 3-6

## そ

---

ソース・ディレクトリ  
Oracle Reports Server のための指定, 11-13  
ソフトウェア構成, 1-7  
ソフトウェア互換性, 10-17

## た

---

帯域幅  
ノード間通信の要件, 3-6  
複数仮想アドレスによる向上, 2-11  
単一ノード・インストール  
Oracle Application Server, 12-17  
単一ノード・クラスタ, 2-5

## ち

---

チェックポイント  
データベース・フェイルオーバーの前, 2-19

## つ

---

通常ファイル  
Oracle Reports Server, 11-25  
フェイルオーバー, 11-25  
通信  
Oracle Intellignet Agent による管理, 2-6  
クラスタ・ノード間で失われた通信, 2-5  
システムの障害発生時に喪失, 7-54  
ツリー・ビュー, 1-4

## て

### 停止

- データベースのモード, 5-4

### ディスク

- 構成, 2-4
- 冗長性, 2-3
- ジョブおよびイベントの情報を格納, 7-29
- リソース・タイプ, 2-8
- 「クラスタ・ディスク」も参照

### ディスク装置

- グループ内のリソース, 2-6
- 「グループの検証」による変更の検出, 7-46
- 追加後の検証, 7-46

### ディスク・リソース

- 「可能所有者ノード」リスト, 2-24

### ディレクトリ索引ファイル

- Oracle HTTP Server, 12-26

### データ・ファイル

- .fmx ファイル, 10-19
- Oracle Forms Server, 10-12, 10-19
- Oracle Reports Server, 11-13
- Oracle Reports をスケジュールする .dat ファイル, 11-13
- 新規の追加, 7-31

### データベース

- 2 次ノード上の停止テスト, 4-3
- DBA パスワードの変更, 7-50
- DLL ファイル, 2-9
- EXCLUSIVE アクセス, 7-22
- INTERNAL アカウント, 7-9
- INTERNAL アカウントの変更, 7-24
- Is Alive ポーリング, 7-50
- MTS\_LISTENER\_ADDRESS パラメータの更新, 7-22
- Net8 リスナーへの依存性の削除, 7-11
- Oracle Data Migration Assistant での移行, 7-25
- Oracle Forms Server によりアクセスされる, 10-13
- Oracle Reports Server によりアクセスされる, 11-14
- Oracle WebDB リスナーに対する複数の指定, 8-11
- インスタンス名, 7-8
- ウィザード入力項目の使用による構成, 4-3
- オフライン化, 5-4, 7-20, 7-24, 7-31
  - FSCMD コマンドの使用, 7-30
  - グループを停止してから再起動, 7-20
  - コールド・バックアップ操作, 7-31
  - パスワードの変更に起因, 7-50

- 保留タイムアウト, 7-49

- 問題, 7-49

- オンライン化の障害のエラー処理, 7-45

- 階層化構成, 10-2

- 仮想アドレス, 7-6

- 「可能所有者ノード」リスト, 7-5

- クライアントによるデータベース接続の損失, 7-54

- クライアント・フェイルオーバー後のアプリケーションの自動再接続, 7-33

- グループへの追加に対する前提条件, 7-3

- グループ・リソース, 2-6

- 検証, 6-5, 6-7

- 構成データ, 7-4

- 構成の手順, 7-4

- コールド・バックアップ操作のための Server Manager の使用, 7-31

- サービス名, 7-8

- 識別情報, 7-8

- 初期化パラメータ・ファイル, 7-8

- 初期化ファイル内の LOCAL\_LISTENER パラメータ, 7-22

- 新規データファイルの追加, 7-31

- スタンドアロン用のリスナー, 7-16

- セキュリティ, 7-22

- 専用ネットワーク・リンクを通じたアクセス, 3-6

- チェックポイント取得, 2-19

- 透過的アプリケーション・フェイルオーバー・デモ, A-1

- 動的リンク・ライブラリ, 2-8

- 名前, 7-8

- 認可, 7-49

- ネット・サービス名エントリ, 7-13, 7-14

- パスワード・ファイルの同期化, 7-23

- バックアップ操作, 7-31

- FSCMD コマンドの使用, 5-7

- 不安定な状態の解決, 7-49

- フェイルオーバー後のデータベースへの再接続, 7-38

- マルチスレッド・サーバーの使用, 7-18

- マルチスレッド・サーバーを使用する Oracle8, 7-18

- マルチスレッド・サーバーを使用する Oracle8i, 7-20

- マルチスレッド・データベースを使用するためのデータベース・リソースの構成, 7-18

- リモート・クライアント・アクセスのための TNSNAMES.ORA ファイルの更新, 7-13

- 「リソース」も参照
- データベース管理者
  - DBA\_AUTHORIZATION パラメータ, 7-22
  - Internal としての接続, 7-49
  - パスワードの変更, 7-50
- データベースの INTERNAL アカウント, 7-9
- データベースの移行, 7-25
- データベース・パスワード・ファイル
  - アカウントの追加, 7-23
- データベース・リカバリ
  - 最適化, 7-30
- データ・ポート
  - Oracle Forms Load Balancer Client, 9-14
  - Oracle Forms Load Balancer Server, 9-11
- データ・ホスト
  - Oracle Forms Load Balancer Client のデフォルト値, 9-14
- デフォルト・ファイル
  - Oracle Application Server Web リスナー, 12-12
- デモ
  - 透過的アプリケーション・フェイルオーバー, A-1

## と

- 透過的アプリケーション・フェイルオーバー, 7-35
  - 概要, 7-34
  - デモ・アプリケーション, 7-44, A-1
- 動的検出
  - Oracle Enterprise Manager, 7-28
- 動的リンク・ライブラリ (DLL)
  - FsResOas.DLL, 2-8
  - FsResOasEx.DLL, 2-8
  - FsResOdbbs.DLL, 2-8, 2-9
  - FsResOdbbsEx.DLL, 2-8, 2-9
  - IP アドレス, 2-8
  - Is Alive ポーリング間隔でのデータベース・アクセス, 7-48
  - Oracle Application Server, 2-8
  - Oracle データベース, 2-8
  - Oracle リソース, 2-8
  - カスタム, 2-8
  - 汎用サービス, 2-8
  - 物理ディスク, 2-8
  - リソースの管理, 2-8
- 登録
  - FORMS60\_PATH 変数, 10-18, 10-19
  - Oracle Reports Server の変数, 11-28

- アプリケーション・フェイルオーバー・コールバック関数, 7-44
- ジョブおよびイベント, 7-27
- 正しい名前解決を確認する ping の実行, 12-31, B-2
- 不適切な名前解決のトラブルシューティング, B-3
- ホスト名および IP アドレス, B-1
- リソースの情報のダンプ, 6-10
- ドキュメント・ルート・ディレクトリ
  - Oracle HTTP Server, 12-25
- ドメイン・アカウント
  - Oracle Fail Safe Manager, 4-5
  - Oracle Fail Safe Server, 4-5
  - Oracle Reports Server, 4-5
  - デフォルト値の更新, 7-14
- トラブルシューティング, 10-16
  - IP アドレスおよびホスト名の登録, 6-3, B-3
  - Oracle Application Server, 12-19
  - Oracle Applications Concurrent Manager, 13-15
  - Oracle Enterprise Manager を使用したセキュリティ・アクセス, 7-54
  - Oracle Forms Load Balancer Server, 9-15
  - Oracle Forms Server, 10-16
  - Oracle HTTP Server, 12-30
  - Oracle MTS Service, 14-15
  - Oracle Reports Server, 11-26
  - Oracle WebDB リスナー, 8-15
  - 仮想アドレスの構成, 7-51
  - 起動問題, 6-12
  - 「クラスタのダンプ」操作, 6-10
  - クラスタへのアクセス, 7-54
  - グループ, 7-48, 13-16, 14-16
  - グループをオンライン化できない, 7-48, 13-16, 14-16
  - 検証ツール, 6-1
  - データベース, 7-44 ~ 7-51
  - データベースに接続できないユーザー, 7-11
  - ネットワーク・アダプタの順序, 6-3, B-3
  - 汎用サービス, 15-17
  - 複数の IP アドレス, B-3
- トランザクション
  - フェイルオーバー, 1-6
  - フェイルオーバー後の未コミット・トランザクションの再実行, 7-32
  - フェイルオーバー後のロールバック, 7-39
- トレース
  - Oracle Forms Load Balancer Server, 9-12

- 使用可能にする, C-2
- フラグ, C-3
- トレース・ファイル, C-4
- トレース・レベル
  - Oracle Forms Load Balancer Server, 9-12

## な

---

- 内部クラスタ通信
  - 仮想アドレスの構成, 7-13

## に

---

- 認証
  - DBA 認証のトラブルシューティング, 7-49
  - Oracle Enterprise Manager の作業環境のチェック, 7-54
  - REMOTE\_LOGIN\_PASSWORDFILE 初期化パラメータ, 7-22
  - SYSDBA ロール, 7-23
  - SYSOPER 権限, 7-23
  - データベースでの INTERNAL の使用, 7-23, 7-49
  - 「バッチ ジョブでログオン」アクセス権限, 7-54
  - 必要な許可および権限, 4-5

## ね

---

- ネーム・サーバー
  - CONNECT\_DATA 文字列, 7-38, 7-39
  - HOST パラメータ, 7-38
- ネット・サービス名, 7-8
  - ofsdomain の追加, 7-13
  - 仮想サーバーの構成時, 7-51
  - 新規作成, 7-13
  - 新規データベース用のエントリ, 7-14
  - 高い可用性を備えたデータベース, 7-8
  - 定義
    - アーカイブされた構成ファイルでの参照, 7-53
  - データベース用の追加, 7-14
  - データベースをグループに追加した後のエントリ, 7-12
  - デフォルト・ドメイン名による追加, 7-13
  - デフォルトのドメイン名, 7-13
- ネットワーク
  - TNSNAMES.ORA ファイルのプロトコル情報, 4-4
  - アクティブ / アクティブ構成でのプライベートの使用, 3-6

- 仮想サーバーの構成時に検出される問題, 7-51
- 構成, B-1
  - Database Configuration Assistant (DBCA) の使用, 7-14
  - 「グループの検証」による検証, 7-46
  - 高い可用性を備えたデータベース, 7-12
  - 複数のリスナーを伴うノードでの SQL\*Net および Net8, 7-16
- 構成情報のトレース, C-4
- 構成ファイルの場所, B-3
- パブリック情報およびプライベート情報のダンプ, 6-10
- ネットワーク構成
  - SQLNET.ORA ファイル内のドメイン名の値の更新, 7-14
- ネットワーク名
  - クラスタ別名, 2-13
  - グループ用の修正, 7-51

## の

---

- ノード
  - 可能所有者, 2-22
  - 既存クラスタへの追加, 4-10
  - クラスタ, 1-2
  - グループのフェイルオーバー先, 2-28
  - グループを所有する優先順位, 2-28
  - 定義, 2-2
  - 優先, 2-30
- ノード障害
  - 影響, 2-33
  - フェイルオーバー, 2-17
- 「ノード」タブ・ページ
  - グループ, 2-23
- ノード名
  - 汎用サービス, 15-7

## は

---

- ハードウェア
  - RAID, 1-7
  - 構成, 1-7
- ハートビート接続, 2-2
  - 定義, 1-7
- パスワード
  - DBA アカountのための変更, 7-50
  - INTERNAL アカountの変更, 7-24

- REMOTE\_LOGIN\_PASSWORDFILE パラメータで  
の問題, 7-49
- 変更によるグループの問題, 7-49
- パスワード・ファイル
  - 同期化, 7-23
- 破損
  - TNSNAMES.ORA および LISTENER.ORA ファイ  
ル, 7-51
- バックアップ操作
  - スクリプトでの FSCMD コマンドの例, 5-7
- パフォーマンス
  - アクティブ / アクティブ構成, 3-4
  - データベースの最適化, 3-6
- パブリック・インターコネクト, 1-7
- パラメータ・ファイル
  - 高い可用性を備えたデータベース, 7-8
- 汎用サービス
  - LocalSystem アカウント, 15-10
  - Oracle Fail Safe によりサポートされる他のサービス  
と比較, 15-2
  - Oracle Fail Safe を使用して構成する利点, 15-2
  - Windows NT サービス, 2-8
  - Windows サービス コントロール マネージャへパラ  
メータを渡す, 15-9
  - Windows レジストリ・キー, 15-15
  - アカウントの指定, 15-9
  - 依存性, 15-13
  - イメージ名, 15-8
  - 仮想アドレス, 15-14
  - 可能所有者ノード, 15-5
  - 可用性を高める構成にしない汎用サービス, 15-3
  - 可用性を高めるための構成, 15-3, 15-5
  - 可用性を高めるための構成手順, 15-3
  - 起動タイプ, 15-10
  - 起動パラメータ, 15-9
  - グループへの追加, 15-3
  - 構成手順, 15-5
  - サービス名, 15-8
  - 作成およびグループへの追加, 15-5
  - 実行可能ファイルの配置場所, 15-8
  - スタンドアロンの検出, 15-3
  - セキュリティ要件, 15-16
  - 定義, 2-8, 15-1
  - データ・ファイルの配置場所, 15-11
  - トラブルシューティング, 15-17
  - ノード名, 15-7
  - 汎用サービスの識別情報, 15-7

- 表示名, 15-8
- リソース・タイプ, 2-8
- 「リソース」も参照
- 汎用サービス DLL
  - 機能, 2-9
- 汎用サービスのグループへの追加, 15-3
  - アカウント, 15-9
  - 依存性, 15-13
  - イメージ名, 15-8
  - 仮想アドレス, 15-14
  - 可能所有者ノード, 15-5
  - 起動パラメータ, 15-9
  - クラスタ・ディスク, 15-11
  - サービス名, 15-8
  - サンプル・サービス, 15-16
  - 手順, 15-3
  - ノード名, 15-7
  - 表示名, 15-8
  - レジストリ・キー, 15-15

---

## ひ

- 表示名
  - 汎用サービス, 15-8

---

## ふ

- ファイバ・チャネル
  - 共有記憶装置インターコネクト, 1-7
- ファイル
  - アーカイブ, 7-53
  - アラート, C-4
  - トレース, C-4
  - ロールバック, 7-53
- ファイルの破損
  - TNSNAMES.ORA および LISTENER.ORA ファイ  
ル, 7-51
- プール・パラメータ
  - Oracle Forms Server, 10-11
- フェイルオーバー, 2-14
  - Oracle Intelligent Agent リソース, 7-30
  - Web サーバー, 12-3
  - アクティブ / アクティブ構成, 3-3
  - アプリケーション, 1-6
  - アプリケーション・コールバック, 7-43
  - アプリケーションにおける OCI エラーの処理,  
7-33, 7-34

- アプリケーションの自動再接続, 1-6, 7-33
- 可能所有者ノードによる影響, 2-28
- 「可能所有者ノード」リストによる影響, 2-28
- クライアント・アプリケーション, 2-33
- グループ, 2-6
- グループの移動先のノード, 2-28
- 「グループの検証」操作による検証, 6-5, 7-46
- 計画外, 2-14
- 計画的, 2-14
- 「検証」コマンドによるチェック, 4-3
- 最高速, 3-3
- 単位, 2-6
- 通常ファイル, 11-25
- 定義, 2-2
- データベースのチェックポイント取得, 2-19
- トランザクションの再実行, 1-6
- ノード障害, 2-17
- バックアップ操作中の使用禁止, 7-31
- 反復を制限する, 2-25
- ポリシーに関する情報のダンプ, 6-10
- ユーザー・セッションの再確立, 7-39
- 「優先ノード」リストによる影響, 2-28
- リソース再起動ポリシーによる影響, 2-27
- リソース障害, 2-15
- ロード・バランス, 2-14
- フェイルオーバー期間, 2-26
- フェイルオーバーしきい値, 2-26
- フェイルオーバー・ポリシー
  - グループ, 2-25
- フェイルバック
  - グループの移動操作による影響, 2-32
  - 「優先ノード」リストによる影響, 2-31
- フェイルバック・ポリシー
  - 「グループの検証」操作による検証, 6-5, 7-46
  - 指定, 2-30
  - 情報のダンプ, 6-10
- 複数 Oracle ホーム, 4-8
  - Database Configuration Assistant を使用した TNSNAMES.ORA の更新, 7-14
  - LISTENER.ORA ファイルのデータベース SID, 7-51
  - アドレスおよびデータベース SID 名の重複のチェック, 7-52
  - 複数のリスナー, 7-16
  - 複数のリスナーの構成, 7-16
  - 変更
    - LISTENER.ORA ファイル, B-4
    - SNMP\_RW.ORA ファイル, B-4

- TNSNAMES.ORA ファイル, B-4
- ネットワーク構成ファイル, B-3
- 複数のネットワーク・カード
  - 不適切な名前解決のトラブルシューティング
  - Windows 2000, B-5
  - Windows NT Service Pack 3, B-3
  - Windows NT Service Pack 4 以降, B-5
- 不十分な権限, 7-54
- プライベート・インターコネクト, 1-7, 2-2
  - 帯域幅, 3-6
- プライベート・ディスク
  - 汎用サービス, 15-11
- フラグ
  - Oracle Fail Safe Server のエラーのトレース, C-4
- プリンタ
  - Oracle Reports Server, 11-15
- プロトコル・アドレス, 7-12

## へ

- 別名
  - LISTENER.ORA ファイルのネットワークの変更, B-4
- 変更
  - Oracle WebDB リスナーの設定, 8-13
  - PL/SQL ゲートウェイの設定, 8-13
  - 「可能所有者ノード」リスト, 2-23

## ほ

- ポート
  - Oracle Forms Server, 10-9
  - Oracle Reports Server, 11-12
- ポート番号
  - Oracle Forms Server 用の検証, 10-17
  - SNMP\_RW.ORA ファイルでの変更, B-4
- ポーリング
  - 障害, 7-49, 7-50
- ホスト名
  - 正しい名前解決の有効性検査, 12-31, B-2
  - 登録, B-1
  - 不適切な名前解決のトラブルシューティング, B-3
- ポリシー
  - フェイルオーバー情報およびフェイルバック情報のダンプ, 6-10
- 保留タイムアウト値
  - 設定, 7-48, 13-16

トラブルシューティング, 7-49, 13-16

## ま

---

マスター Oracle Reports Server

高い可用性の提供, 11-1

マスター / スレーブ実装

Oracle Reports Server のための構成, 11-19

マルチスレッド・サーバー

Oracle Fail Safe で使用するための構成, 7-18

Oracle8i データベースおよび Oracle Fail Safe との使用, 7-20

Oracle8i データベースのための構成, 7-20

Oracle8 データベースおよび Oracle Fail Safe との使用, 7-18

Oracle8 データベースのための構成, 7-18, 7-20

マルチノード・インストールのプライマリ・ノード

Oracle Application Server, 12-17

マルチフォーム・アプリケーション, 10-19

## み

---

ミッション・クリティカル・アプリケーション

アクティブ / アクティブ構成, 3-5

## め

---

メタデータ

クオラム・リソース, 2-5

## も

---

文字列

Oracle Fail Safe Server のトレース機能, C-3

## ゆ

---

ユーザー

Oracle Fail Safe Manager のドメイン・アカウント, 4-7

Windows NT サービスのためのアカウント情報の変更, 4-7

ユーザー・アカウント

Oracle Fail Safe, 4-6

Oracle Reports Server, 11-15

ユーザー・セッション

フェイルオーバー後の再確立, 7-39

ユーザー設定項目

設定, 7-54

優先ノード

データベース停止のテスト, 4-3

フェイルオーバー・ポリシーの作成, 4-3

「優先ノード」リスト

グループの移動操作, 2-32

フェイルオーバーへの影響, 2-28

フェイルバックへの影響, 2-31

## り

---

リクエスト・ポート

Oracle Forms Load Balancer Server, 9-12

リスナー

Listener 制御ユーティリティ (LSNRCTL) を使用した作成, 7-53

LOCAL\_LISTENER パラメータへの指定, 7-21

MTS\_DISPATCHER への情報の指定, 7-21

Oracle Application Server Web, 12-18

Oracle Net8 リスナーで変更された動作, 7-11

アドレスの変更, 7-22

起動, 7-16

グループに追加するデータベース用に作成, 7-12

構成, 4-3

作成時の問題, 7-53

出力ファイル, 7-53

新規作成, 7-53

定義の変更または作成後の問題, 7-52

トラブルシューティング, 7-48

ネットワーク構成に関する考慮事項, B-3

ネットワーク・リソース, 2-6

複数のリスナーでの SQL\*Net または Net8 の構成, 7-16

マルチスレッド・サーバーのための変更, 7-20

ロールバック・ファイル内の定義, 7-52

ログ・ファイル, 7-48

リソース, 1-2

DLL, 2-8

Oracle Intelligent Agent, 7-28

アクセス, 2-6

依存性, 2-7

「一般」タブ・ページ, 2-23

オフライン化, 5-3, 5-4

FSCMD の使用, 5-1

オンライン化, 5-3

FSCMD の使用, 5-1



- 仮想サーバーを介したアクセス, 2-12
- 「可能所有者ノード」リスト, 2-22
- クライアントのアクセス, 2-2
- グループの「ノード」タブ・ページでの変更による影響, 2-24
- 検出, 4-8
- 構成の誤りの修正, 7-46
- 構成の検証, 6-5, 6-7
- 障害, 2-15
- 定義, 2-6
- データベース構成に使用するディスク, 4-3
- 名前の変更, 4-8
- フェイルオーバー・ポリシーによるグループへの影響, 2-25
- 優先ノードに戻ることに, 2-30
- 例, 2-6
- リソース再起動ポリシー, 2-27
- リソース・タイプ
  - IP アドレス, 2-8
  - MSCS, 2-8
  - Oracle Application Server, 2-8
  - Oracle Applications Concurrent Manager, 2-8
  - Oracle Fail Safe Manager での表示, 2-9
  - Oracle Forms, 2-8
  - Oracle Forms Load Balancer Server, 2-8
  - Oracle HTTP Server, 2-8
  - Oracle MTS Service, 2-8
  - Oracle Reports, 2-8
  - Oracle TNS Listener, 2-9
  - Oracle WebDB, 2-8
  - Oracle データベース, 2-8
  - カスタム, 2-8
  - 汎用サービス, 2-8
  - 物理ディスク, 2-8
- リソースの依存性, 2-7
- 「リソースをグループに追加」ウィザード
  - Oracle Application Server, 12-8
  - Oracle Applications Concurrent Manager, 13-6
  - Oracle Forms Load Balancer Server, 9-7
  - Oracle Forms Server, 10-6
  - Oracle HTTP Server, 12-22
  - Oracle Intelligent Agent, 7-28
  - Oracle MTS Service, 14-7
  - Oracle Reports Server, 11-8
  - Oracle WebDB リスナー, 8-6
  - Oracle データベース, 7-4
  - ログ情報, C-4

- 汎用サービス, 15-5
- 「リソースをグループに追加」操作
  - トラブルシューティング, 7-11
- 利点
  - Oracle Fail Safe, 1-2
  - Oracle Fail Safe を使用した汎用サービスの構成, 15-2
  - クラスタ対応のアプリケーション, 7-32

---

## る

- ルート・ディレクトリ
  - Oracle Application Web リスナー, 12-13

---

## れ

- レジストリ・エディタ
  - ホスト名および IP アドレスの登録, B-1
- レジストリ・キー
  - Oracle Applications Concurrent Manager 構成 ID, 13-5
  - Oracle Forms Server 仮想アドレス, 10-8
  - Oracle Forms Server ポート番号, 10-9
  - Oracle Forms 接続モード, 10-11
  - Oracle Forms プール・パラメータ, 10-11
  - Oracle Reports キャッシュ・ディレクトリ, 11-28
  - クラスタ全体に複製, 15-15
  - 汎用サービス, 15-15
- レジストリ変数
  - FORMS60\_PATH, 10-12
  - Oracle Applications Concurrent Manager 構成 ID, 13-10
  - Oracle Reports キャッシュ・ディレクトリ, 11-28
  - REPORTS60\_CGIMAP, 11-17
  - REPORTS60\_PHYSICAL\_MAP, 11-17, 11-28
  - REPORTS60\_VIRTUAL\_MAP, 11-17, 11-28

---

## ろ

- ローカル・エリア・ネットワーク, 1-7
- ロード・バランス, 2-14, 9-2
  - Oracle Forms Server, 9-2
  - Oracle Reports Server, 11-5
  - アクティブ / アクティブ構成, 3-5
  - 静的, 2-18
  - 「Oracle Forms Load Balancer Server」も参照
- ローリング・アップグレード

- アプリケーション・ソフトウェア, 1-4
- ロールバック
  - クラスタワイド操作, 7-47
  - ファイル, 7-52, 7-53

## わ

---

- ワーカー・プロセス
  - ifweb60.exe, 10-11
  - プール, 10-11
- ワイド・エリア・ネットワーク, 1-7
- 割当て
  - クラスタへの IP アドレス割当て, 2-13